Self-organised Sound with Autonomous Instruments: Aesthetics and experiments.

Risto Holopainen

Thesis submitted for the degree of PhD

at the Department of Musicology,

University of Oslo

February 2012

Abstract

Autonomous instruments are computer programmes that generate music algorithmically and without realtime interaction, from the waveform level up to the large scale form. This thesis addresses questions of aesthetics and the role of the composer in music made with more or less autonomous instruments. Furthermore, a particular form of autonomous instruments, called *feature-feedback systems*, are developed. These instruments use feature extractors in a feedback loop, where features of the audio output modulate the synthesis parameters.

Methods adopted mainly from chaos theory are used in experimental investigations of several feature-feedback systems. Design principles are also introduced for controlling limited aspects of these instruments. These experimental methods and design strategies are not widely used in current research on synthesis models, but may be useful to anyone who wishes to build similar instruments.

Whereas Varèse preferred to designate music as "organised sound", autonomous instruments may be said to engender *self-organised sound*, in the sense that the result was not specified in detail by the composer—in fact, the result may not even have been expected. Thus, there is a trade-off between a deliberate sound-shaping by the composer on the one hand, and truly autonomous instruments on the other. The idiomatic way of operating an autonomous instrument is experimentation followed by serendipitous discovery.

Preface

Two broad topics interested me at the time when I conceived of the directions for this thesis, and they still do. One of them is nonlinear feedback systems and chaos, the other is the application of feature extractors in the analysis and synthesis of sounds. I thought they could be combined somehow, so I just had to invent feature-feedback systems.

During the first three or so years of this project (begun in August 2008), I did not know what to call the systems that I wanted to investigate. In the autumn of 2008, it crossed my mind that there must be some people out there who experiment with similar things, so I sent out a request on the cec-conference discussion forum for practitioners of "adaptive synthesis", which was the term used at that time. This is more or less how I then described it:

> Essentially, adaptive synthesis consists of a sound generator (a synthesis technique, either digital or analogue), a signal analysis unit which performs feature extraction of the signal produced by the generator, and finally there is a mapping from the analysed attributes back to the control parameters of the sound generator.

Several people responded to my request with useful suggestions. It is no coincidence that many of them independently pointed to the work of Agostino Di Scipio, whose seminars, music and writings have had a noticeable influence on a current generation of composers and musicians. In particular, I would like to thank Di Scipio himself for his helpful reply, Owen Green for sharing some of his music and other information, and Nick Collins for pointing me in useful directions and for helping me out with installing his Autocousmatic programme.

The common denominator of all the practitioners of "adaptive synthesis" was that they worked with live-electronics. However, as I developed my own feature-feedback systems another aspect foregrounded, namely that of autonomy or self-regulating processes. I have always preferred to work with fixed media rather than live-electronics and realtime processing in my own electroacoustic music making. Therefore, my engagement with feature-feedback systems is restricted to offline processes and computer programming.

Since there does not appear to be any previous studies on autonomous feature-feedback systems, I did not often get the feeling of plodding along in well-trodden paths of research. Nonetheless, many people have contributed to this project in various ways. First of all, I would like to thank my main supervisor Rolf Inge Godøy for his longstanding generous support and encouragement. Sverre Holm entered as my second supervisor half-ways into this project. Eystein Sandvik and Steven Feld read and commented on an early version of Chapter 5. Many others at the Department of Musicology have also con-

tributed with their general encouragement. Asbjørn Flø lent me a recording of *Gendy3*, and has been a great support otherwise by his persistent curiosity.

In an extended e-mail correspondence, Scott Nordlund has introduced me to several fascinating examples of what we think might be autonomous instruments, ranging from analogue neural nets to no-input mixers. In particular, I would like to thank him for sharing his own recordings and for making me reconsider the idea of autonomous instruments once more.

Maury Sasslaff did the copyediting on most of Chapters 1, 2, 4 and 5; then Melinda Hill took over and did the copyediting of Chapter 8. Any remaining stylistic inconsistencies within or across the chapters are my sole responsibility. I would also like to thank the members of the committee, Eduardo Reck Miranda, Stefania Serafin, and Alexander Refsum Jensenius for their advice.

In December 2010 I followed a week-long PhD course at the Aalborg University which was very inspiring. Small traces of ideas from that course (especially due to Dave Meredith and Bob Sturm) seem to have made their way into this thesis. Thanks goes to the tea-drinking contingent of fellow PhD students for a pleasurable time, and in particular to Iballa Burunat for her response on the test version of the Autonomous Instrument Song Contest (the one with the shrill sounds), and of course to all those who answered it.

Risto Holopainen,

Oslo, February 2012

Contents

Chapter 1

Introduction

The motivation behind the present thesis is mainly a curiosity about a feedback system that could be used as a musical instrument. The question was, what would happen if one were to put a synthesis technique into a feedback loop and, so to speak, let it listen to its own output whilst modifying its synthesis parameters in response to the sound it was currently producing? There are already some examples of similar feedback systems, in which a musician may typically interact with the system. In contrast, my research interest soon narrowed down to systems that took no realtime input. Such systems will here be referred to as *autonomous instruments*.

There are not many well-known exemplars of music made with strictly autonomous instruments. Some plausible reasons for this will be discussed in this thesis. Nonetheless, there are several examples of digital music instruments that allow for interaction, although the purpose is not to make the performer directly responsible for every nuance of sound as a violinist would be, but rather to engage the musician in a dialogue. This kind of instruments will be called *semi-autonomous*, because the instrument is able to respond in the musical dialogue with output that the performer did not directly call for.

Topics such as *self-organisation* and *emergence* are recurrent in writings on more or less autonomous instruments. Moreover, there appears to be some shared aesthetic views among the practitioners of music made with autonomous instruments. Related to this aesthetics are dichotomies such as nature versus the artificial and the self-organised as opposed to the deliberately designed. In this thesis, notions of self-organisation will be analysed and related to musical practice. One may think that music resulting from a self-organising process cannot have been deliberately organised by the composer. Whether this is true or a common misunderstanding is another question that will be addressed.

While the discussion of the aesthetics of music made with more or less autonomous instruments is an important part of this thesis, the most original contribution is the introduction of a novel class of autonomous instruments that we call *feature-feedback systems*. These systems consist of three components; a signal generator produces the audio output, a feature extractor analyses it, and a mapping function translates the analysed feature to synthesis parameters (see Figure 1.1). Feature-feedback systems as used here are not interactive in realtime, and most of them are deterministic systems in the sense that they always produce the same output if the initial conditions are the same.

Already some preliminary experimentation with feature-feedback systems revealed

that their behaviour would not be easily understood. Therefore, one of the main purposes with this thesis is to develop a theory as well as practical know-how related to the operation of feature-feedback systems. Now, a closer study of feature-feedback systems leads to several other questions to be inverstigated. First, we need to understand the relationship between synthesis parameters and feature extractors, and in turn, how the sounds are perceived and how this relates to feature extractors. Then, most importantly, a closer look at dynamic systems and chaos theory will be necessary for setting up the proper theoretical framework for these feedback systems.

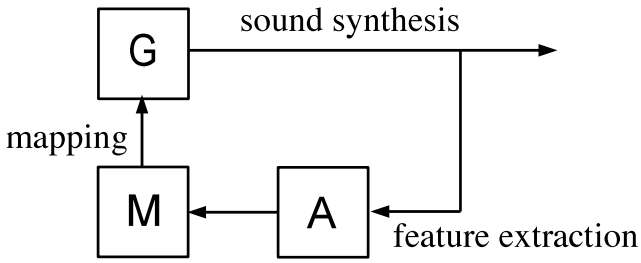

Figure 1.1: Schematic of a basic feature-feedback system.

Experimental investigations of feature-feedback systems (as well as other dynamic systems and synthesis techniques) form a prominent part of this thesis. In effect, one of the major contributions of this thesis is to show how a range of experimental techniques drawn from dynamic systems theory can be applied to any synthesis techniques, and to feature-feedback systems in particular. So far, it has not been common to study novel synthesis techniques as though they were some physical system with unknown properties, but this is exactly the approach taken here.

The new findings about feature-feedback systems, and the broad outlook on the musical scene related to autonomous instruments that are presented in this thesis should be of interest to composers, musicians and musicologists in the field of computer music. Due to the interdisciplinary nature of this thesis, maybe others will find it fascinating as well. Many new feature-feedback systems are presented in full detail, but more general design principles are also provided that can be used as recipes for anyone who wishes to experiment with these techniques for musical purposes or just out of curiosity. The primary motivation behind this research project has actually not been the making of music, and perhaps not even the crafting of useful musical instruments as much as a theoretical understanding of their manner of operation and of efficacious design principles. This point will be clarified below and related to the emerging field of research in the arts.

In the rest of this chapter, some related work will be reviewed, then different approaches to synthesis models are contrasted. The final two sections of this chapter address the musical and aesthetic setting in which autonomous instruments are situated.

1.1 Previous and related work

Machine listening is often a crucial component in semi-autonomous instruments. Feature extraction is then used on the incoming audio signal to extract some pereptually salient descriptors of the sound, but feature extractors have many other uses in techniques such as adaptive effects processing and adaptive synthesis. Many of these techniques have served as an inspiration for the present work on feature-feedback systems and will therefore be briefly reviewed here.

Feedback in various forms is another topic that will be important throughout this thesis. Indeed, feedback loops of various kinds are ubiquitous and, in effect, indispensable for music making as will be exemplified in Section 1.1.3. But first, we shall clarify the aims of this research project by comparing it to artistic research.

1.1.1 Concerning research in the arts

Immediately, it may appear that the goal of developing a new kind of musical instrument would imply an affiliation with so-called artistic research, or *research in the arts*. This is research where the production or performance of works of art is an inseparable part of the research itself, although the emphasis may be more on the working process or the final result. The present thesis does not try to document the traces of a process that led to musical compositions, nor was the composition of music a part of the research. However, the studies of feature-feedback systems and other dynamic systems as well as feature extractors has resulted in a knowledge that can be useful to anyone who might want to make their music with feature-feedback systems. Indeed, myself being a composer, I have made some attempts to compose music using feature-feedback systems, but I do not consider this to be the right place to document my experiences as a composer. Doing so would be more appropriate in the context of research in the arts. Nonetheless, the questions that motivated this research could perhaps not have been posed by anyone else than a composer. In particular, I will use my background knowledge of how some composers think, gained from numerous conversations with composer colleagues and, needless to say, from my own experience; this will perhaps become most evident in the final chapter.

In order to substantiate the claim that this is not reasearch *in* the arts, let us summarise the various kinds of research that are involved with the arts to various degrees, following Henk Borgdorff (2006).

Borgdorff actually draws a distinction between three different approaches to art-related research. First, there is the traditional academic *research on the arts*, including musicology and other disciplines of the humanities. Historically, musicology has mainly been concerned with the analysis or interpretation of existing music from a theoretical distance. There is a separation between the researcher and the object of research; the musicologist is usually not directly involved in producing the music that is the subject of research.

Then there is *research for the arts*, which Borgdorff specifies as "applied research" where art is not the object of investigation, but its objective. As an example, Borgdorff mentions the study of extended techniques of instrumental performance using live elec-

tronics. The point of this type of research is to deliver the tools for artistic practice, including knowledge about novel instruments. This type of research includes instrument making and engineering, and thus comes close to the practical investigations of autonomous instruments in the present thesis.

Finally, *research in the arts*, according to Borgdorff, is the most controversial of the three types of research. It assumes that there cannot be any distance between the researcher and the practice of art making. Then, the artistic practice is essential for the research process as well as for the results of the research. The controversy that surrounds research in the arts has to do with the troubling question, in what sense is this research? It is virtually inconceivable that someone who is not an artist would be able to do research in the arts. But then, as Borgdorff notes, the objectivity of the research becomes an urgent concern, because academic research is supposed to be indifferent to who performs it. Of course there are exceptions, such as participant observation in anthropology.

> In summary, research in the arts is performed by artists as a rule, but their research envisages a broader-ranging impact than the development of their own artistry. Unlike other domains of knowledge, art research employs both experimental and hermeneutic methods in addressing itself to particular and singular products and processes (Borgdorff, 2006, p. 18).

From this description, it may seem that the present thesis has much in common with research *in* the arts. It is written by a composer, indeed it hopes to reach a broad audience and, not least, experimental methods will play an important role. However, the crucial difference is that the immediate goal is not to make music; that may come later, but is not part of the research. Moreover, we will deal also with existing music by other composers and discuss how it relates to concepts such as autonomy and self-organisation, and try to delineate aspects of the aesthetics of this field of music making; this is of course closer to traditional musicology and research *on* the arts.

Clearly, the category of research *for* the arts comes close to the approach taken in parts of this thesis. Autonomous instruments are something that may be built, or written in a computer language to be specific, whence an engineering aspect will be important. However, an engineering point of view differs from the experimental orientation that will also be important in this thesis. One could say that, with the experimental attitude, one tries to find out *how* things work rather than trying to make them work as one would like.

In this context, musical experiments is an interesting category. As Mauceri (1997) has pointed out, the term "experimental" has been used to suggest an analogy between the new experimental music and science, sometimes seeking legitimacy by appealing to the authority of science. The first large scale experiment in algorithmic composition using computers was undertaken by Hiller and Isaacson in 1957. They claimed that the *Illiac Suite*, the string quartet that was composed based on the computer generated output, was not supposed to be regarded as a work of art, but rather as a laboratory notebook. Two of the movements (or "experiments") of the string quartet were written in a contemporary style, which Mauceri finds problematic in view of their being scientific experiments. If these movements set out to display some species of novelty, then it is not clear what criteria to use in the evaluation of the experiment's success or failure, and Mauceri then

boldly concludes that "... the *Illiac Suite* is neither musical art nor science" (Mauceri, 1997, p. 196). This illustrates the kind of scepticism that research in the arts may still face today.

As we introduce novel autonomous instruments, we do so in the understanding that, if they are to be used for making music, then the composer is responsible for evaluating their output. Although my personal predilections have guided the development of autonomous instruments, some examples will be given that, in my opinion, are not yet very musically successful. Those examples can primarily be found in Chapter 6, where the purpose is anyway to introduce analysis methods suitable for studying autonomous instruments. In conclusion, it should now be evident that the present research does not try to investigate autonomous instruments by making music with them, but by carrying out experiments that will result in a better understanding of them, which in turn can be useful for composing music with them. Therefore, it is not research *in* the arts, but *on* and *for* the arts.

1.1.2 Adaptive effects and synthesis

Let us now turn to one of the main components in those autonomous instruments that we are about to construct. *Audio feature extractors* have been developed in the contexts of speech analysis, studies on the perception of musical timbre, music information retrieval and sound classification (Peeters et al., 2011), and they find many uses in music technology, from composition and performance to musicology. Feature extractors, also called signal descriptors, can be used as general analysis tools that produce information on how an audio signal changes over time with respect to perceptual attributes such as loudness, pitch, and various timbral categories. This information can be fed as a stream of control data either to a synthesis model or to a digital audio effect. Using feature extraction in this way as a control source makes the effect or synthesis model adaptive to the input signal. There are many interesting musical applications for such adaptive models, some of which will be discussed below.

Another application of feature extraction is that of feature-feedback systems. Putting feature extractors into closed feedback loops may result in quite unintuitive systems with hard-to-control dynamics, but in some cases feature extractors can be used to increase the controllability of otherwise wayward synthesis models. Pitch control of nonlinear oscillators is a case in point (see Section 4.4.3). Perhaps the most prominent use of feature extractors in music these days is in machine listening and interactive music making. The computer is bestowed with perceptive faculties, as it were, by analysing audio input, and modifying its output behaviour in response to the incoming sound.

An important source of inspiration for feature-feedback systems is the work on adaptive digital audio effects. The idea of adaptive audio effects is relatively recent (Verfaille and Arfib, 2001), although predecessors date back to the analogue era. Dynamic processing, such as compressors and expanders, were the first examples of such effects. Adaptivity enters the picture since the processed signal's gain is controlled by the incoming signal's amplitude level. Another often heard adaptive audio effect is the auto-tuner, which is typically applied to vocals. More generally, the input signal is analysed for certain attributes that dynamically control the audio effect's parameters. Other work

on adaptive audio effects has focused on the extraction of relevant sound attributes and suitable mappings from these to the parameters of the audio effect (Verfaille, 2003).

The notion of adaptivity in audio effects is straightforwardly transferred to synthesis models that normally do not take an audio input. An adaptive synthesis model can be conceptually broken down into the same three components as a feature-feedback system: it has a signal generator, a feature extractor, and a mapping from features to synthesis parameters. The crucial difference, of course, is that there is no feedback path from the output to the feature extractor in adaptive synthesis.

In so-called *feature based synthesis*, an arbitrary synthesis model may be used, but the goal is to match extracted features from a target sound to parameter values of the synthesis model that will minimise the difference between the target sound and the synthesised sound. A similar idea appears in *concatenative synthesis*, where the target sound is resynthesised by splicing together short fragments from a large database of sounds. Nevertheless, the idea of analysing a sound and using analysis data to modulate another sound can be applied more freely if the goal of closely matching resynthesis is given up. All of these signal-adaptive strategies of synthesis and effects processing will be further discussed in Chapter 3 (see Section 3.1).

1.1.3 Closing the feedback loop

As an intuitive metaphor of what feature-feedback systems are about, consider the following situation. A flutist begins playing a tone, only to instantly realise that it was out of tune. Then, the musician has to adjust the playing technique until the tone sounds right. Pitch correction employs the feedback from the produced sound through audition to motor control, including adjustments of embouchure or fingering as needed. The adjusted playing technique immediately changes the produced sound, which is continuously monitored by the listening musician. The parallel to feature extraction as a form of listening, and sound synthesis as playing the instrument, should be obvious. At this point, however, the metaphor of the listening musician has to be left behind, because the aim of the present thesis is not to model existing instruments or the way real musicians play them.

All feature-feedback systems necessarily form closed loops, which is a fact of utmost importance. They also contain nonlinear elements, which make them nonlinear dynamical systems. Chaotic systems such as the Rössler attractor or the logistic map are simple deterministic systems capable of complex behaviour. In comparison, most feature-feedback systems are significantly more complicated than low-dimensional chaotic systems, in the sense that writing out the system equation explicitly would result in a very large system with many variables.

When the rules describing a system are far simpler than the behaviour of the system, one might speak of emergent qualities. A closely related concept is that of a *computationally irreducible* system, which is a system whose future state cannot be predicted from its present state. The only way one may gain access to this future state is to run the system for the corresponding amount of time. Feature-feedback systems are very likely to display computational irreducibility, although they may also be designed so as to be more predictable.

There are many conceivable variations on the basic feedback loop (Figure 1.1) when constructing feature-feedback systems. Apart from this basic feedback loop, more complicated systems may be built by connecting several simpler systems in cross-coupled loops, or even nesting feature-feedback systems inside each other. External input from an audio signal or gestural controller could be provided to any of these models, although that would of course make the system non-autonomous. Although feature-feedback systems are "no input" systems, they have a signal chain into which adaptive effects could be inserted. Indeed, if you take an adaptive audio effect, unplug its input and instead route its output signal back to its input, you then have a feature-feedback system.

Autonomous instruments as we treat them here are usually computer programmes that algorithmically synthesise long stretches of sound without real-time control of any kind. This approach is a bit unusual, especially these days when interactivity is so much in focus. Related efforts are being made though, particularly in certain forms of generative music, but it is safe to say that this is not a mainstream tendency. We will *not* claim that autonomous instruments offer a better or simpler alternative than other forms of music making, only that it is a somewhat neglected possibility that deserves more attention. In fact, the non-interactive mode is precisely the necessary condition for making detailed studies of these instruments as dynamic systems.

If autonomous instruments are said to be non-interactive, this just means that realtime interaction is excluded. Obviously, no music could be made without writing the programme that generates the music, and programming is a highly interactive activity. Furthermore, typical working processes in algorithmic music usually involve an extended cycle of programming, generating output, listening to it and evaluating the result. Then follows revision of the programme and an arbitrary number of repetitions of this cycle. This is yet another feedback loop as illustrated in Figure 1.2; its similarity with the schematic diagram of a feature-feedback system is not an accident.

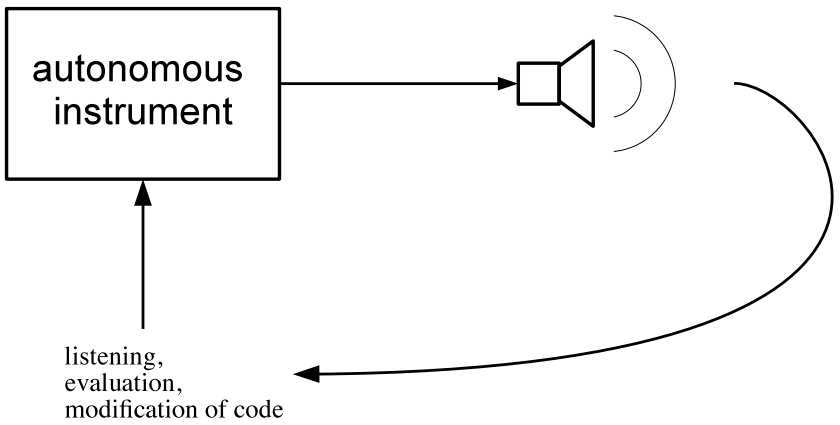

Figure 1.2: The larger feedback loop.

In contrast to strictly autonomous instruments, there are some interesting examples of more interactive systems that share some general properties with autonomous instruments. These include feedback loops and feature extractors, similarly to feature-feedback systems, but they involve acoustic spaces and live electronics as part of the system. It is

appropriate to distinguish between *open* and *closed* systems. Autonomous instruments are closed systems. These systems give little opportunity for user interaction for the reason that they do not offer realtime control; nor are they open to other accidental influences from the environment. At the opposite end are open systems, which are well exemplified by Agostino Di Scipio's *Audible Ecosystemics* series of works (Di Scipio, 2008, 2003). Acoustic feedback is a crucial factor in these works, together with digital signal processing that often has a counterbalancing effect on various sonorous tendencies. Di Scipio's works are open in the sense that the acoustic ambience is not merely the space where sound is projected, but it contributes in a stronger sense, even acting as a sound generator when background noise is taken as the input to the system. So, when sound (and not just a digital or analogue signal) is in fact the interface between system components, it is evident that any perturbations of the room acoustics, such as the presence of an audience, may influence the overall behaviour of the system.

Instrument making is a deliberate design process, more concerned with the end result than we would be when merely letting the instrument generate what it might. It must be understood that design is partly at odds with truly self-governing processes. The same can be said about live interaction: by intervening in the autonomous sound generation of an algorithm, the process becomes partly dependent on active control by the musician and hence loses its autonomy. "Instrument making" is of course just as much a metaphor as "instrument playing" is. The whole concept of instrument—making or playing—may seem a bit strained as a designation of a process of algorithmic sound synthesis. However, later in this chapter we shall make comparisons between different categories of instruments, where autonomous instruments will be juxtaposed with more interactive instruments.

1.2 Modelling

When musical instruments are studied in acoustics, the goal is to understand and model the sound producing mechanism. In contrast, we will be more concerned with understanding synthesis models after having put them together. One might think that the *thing* to be modelled must be well known in advance, but perhaps not its properties. The thing to be modelled is supposed to exist already. This view may be correct for some approaches to synthesis models, but it misses the point in the autonomous instrument approach to sound synthesis. The essence of the latter approach is well captured by Tim Perkis, writing about David Tudor:

> His [Tudor's] practice was to string together cheap electronic components into complex and ill-understood circuits which exhibited complex and ill-understood behavior. These networks can be thought of as simulations of some kind: simulations perhaps of things that never existed, if that makes any sense. The dynamic behavior of these complex systems is very explicitly what this music is about (Perkis, 2003, p. 80).

This experimental approach may be contrasted with an engineering approach that shuns any ill-understood components exhibiting complex and difficult-to-control behaviour. En-

gineers of course do so for good reason, but experimentally minded composers can afford to explore the unknown.

Next, different approaches to sound synthesis will be delineated; then the notions of modelling will be discussed. Control data and parameter spaces are two fundamental aspects of any synthesis model. Their role in autonomous instruments will be clarified below by a comparison with how they are specified and used in other synthesis models.

1.2.1 Perspectives on synthesis models

Various taxonomies of synthesis models have been proposed over the years. J. O. Smith (1991) introduced a taxonomy with four classes: Processed recordings, spectral modelling, physical modelling, and abstract algorithms. He speculated that in the future, physical and spectral modelling would be the dominant trends. Indeed, his prediction has turned out to be quite correct, at least as far as research activity is concerned. The category of processed recordings, including granular synthesis, sampling, and wavetable synthesis has enjoyed the benefits of spectral modelling and to some extent been absorbed into that category. As for abstract algorithms (including FM, waveshaping, phase distortion, and Karplus-Strong), Smith notes that the absence of analysis methods that can be used for sound design is problematic, and that most sounds produced by such means are "simply uninteresting". In a later evaluation of extant synthesis models, Tolonen et al. (1998) clearly favoured spectral and physical models over any others.

Dating back to the first years of electroacoustic music, abstract algorithms include any more or less arbitrary mathematical formula that can be turned into an audio signal. The historical reason for their early popularity was their computational efficiency and simplicity of implementation. Their current decline has to do with the poor performance of these models when it comes to intuitive control, and even more their lack of flexibility and capability of verisimilar sound reproduction. But this description is perhaps a bit one-sided. In fact, spectral matching and genetic algorithms have come to the rescue in the sound design problem for some abstract models, such as FM (Horner, 2003).

Another thing that tends to be forgotten is that abstract models are constantly being developed, although not always flagged as such. But these developments are mainly the work of composers rather than researchers. Autonomous instruments could use any kind of synthesis model, but the examples that will be developed here all seem to most naturally belong to the category of abstract algorithms. In fact, they have much in common with some "nonstandard synthesis methods", under which Roads (1996, ch. 8) lists waveform segment techniques, graphic sound synthesis, noise modulation, and stochastic waveform synthesis.

An alternative to the technically motivated taxonomy of synthesis models is to look at the reasons for their use. For example, one may wish to imitate and vary a given sound, search for the unheard sound or create hybridisations between sounds. Nonstandard synthesis is not concerned with trying to imitate any previously known sound, but to find idiomatic musical expressions of digital computation. Holtzman (1979) introduced a technique that he called *instruction synthesis*, and further qualified it as a "non-standard" synthesis technique, thereby coining the term. According to Holtzman, nonstandard synthesis generates noises that differ from those produced in the traditional instrumental

repertoire as well as much electronic music; furthermore, the sound is specified in terms of basic digital operations rather than in acoustic terms, including concepts such as frequency, pitch or harmonic structure.

In the 1970s, Xenakis experimented with stochastic synthesis of waveforms and Herbert Brün developed his Sawdust programme which took waveform segments as its basic unit. Meanwhile, nonstandard synthesis techniques were also actively developed at the Institute of Sonology by G. M. Koenig, Paul Berg and others. Some aspects of this work include the realtime generation of sound with interactive programmes, which was not common at the time; furthermore, this immediate response made listening to the results an important part of the process, and there was a focus on rule-based generation of material inspired by serialist methods rather than direct specification of the sound in acoustic terms (Berg, 2009). Whereas nonstandard approaches to synthesis have sometimes been reproached for disregarding the perceptual aspect of the result, this critique tends to forget the continual aural evaluation which is necessary when working with synthesis methods that lack predictability (Döbereiner, 2011).

The present study of autonomous instruments and feature-feedback systems in particular has certain similarities with nonstandard synthesis, but there are differences as well. The rule-based approach and the unpredictable results that necessitate continual aural evaluation is certainly a common aspect. However, we will by no means ignore acoustic or psychoacoustic principles in the construction of autonomous instruments; to the contrary, the relationship between a synthesis model and various audio features will be the point of departure. The results may nevertheless be unpredictable, which makes it necessary to take a more empirical approach including evaluation by listening and other forms of experimental study.

1.2.2 Models and simulation

A model of something can be a map, a scale model, a numerical simulation, or an abstract theoretical construct. Common to these varieties of models is that they function as representations of some original object or phenomenon. Models can be used to predict future situations, as in weather forecasts, or to mimic some properties of an object. Unfortunately, the term "model" is ambiguous, as it is used both in the sense of a *model for* (the terrain is the model for the map) and a *model of* (the map is a model of the terrain). In any case, the notion of models *of* something presupposes a reality that is being modelled, as Luc Döbereiner points out in the context of nonstandard synthesis:

> A sound-synthesis method is a formalism, and this formalism can be conceived of as a model. A common [...] understanding of models presupposes a separation between an empirical reality and a formal modeling of that reality. The assumption is that we are on the one hand neutrally observing the facts, and on the other hand, actively producing a model. It is a confrontation between a real thing and an artificial reproduction, [...] and it essentially boilds down to [...] the opposition of "nature" and "culture" (Döbereiner, 2011, p. 33).

If a synthesis model were necessarily a model of something previously existing, then where would the ideas of abstract or nonstandard synthesis come from? Inasmuch as nonstandard synthesis draws upon abstract mathematical formulas, the answer seems to be that the ideas come from where mathematics come from. Mathematicians are often found of thinking that mathematical ideas exist in some platonic sphere, independent of humans. In contrast, Lakoff and Núñez (2000) have argued that mathematics is embodied, in the sense that it is not independent of the brains that construct it and hence not something existing "out there" waiting to be discovered. Incidentally, there are few claims that novel synthesis models are *discovered*; that they exist in a sphere independent of human musicians and programmers. Instead, the opposite occasionally happens: that the creator of some synthesis algorithm claims the rights to it in the form of patents[1].

Sound synthesis by physical modelling is usually a model *of* something. There is a known acoustic or electronic instrument that one tries to model. Although research in physical modelling usually implies studies of existing acoustical instruments, its greatest artistic promise may lie in the possibility to create virtual instruments that would be infeasible or even impossible to construct in the real world, such as making an instrument that would be able to dynamically alter its shape or size. Indeed, Kojs et al. (2007) list compositions made with "cyberinstruments" or physical models that either extend existing instruments, or make hybrids between existing instruments, or take an abstract approach by building novel instruments from basic components such as masses, springs and dampers.

In terms of a research programme, physical modelling typically has the objective to provide useful models of previously known acoustical instruments. In constrast, the strategy of abstract algorithms does not necessarily have the advantage of a predefined sound that its results should be measured against. The process of sound design with abstract synthesis models is more likely to take the form of an iterated cycle where one tries out sounds and modifies synthesis parameters or the entire instrument. The design of abstract synthesis models can of course also be guided by a particular sound that one is trying to imitate, as is amply exemplified in many synthesisers with patches named and modelled after common instruments. However, if there is no previously known phenomenon that the abstract synthesis model tries to imitate, the process of finding a suitable synthesis model is only dictated by the situation it will be used in, as well as the sound designer's taste.

In synthesis by autonomous instruments, there is no known original that one tries to simulate. Yet a premise is that the synthesis model should be able to produce a sufficiently interesting sound. This criterion is of course fully subjective, but we may add a slightly more objective success criterion of an autonomous instrument: The complexity of the model should not overshadow the complexity of the sounds it is capable of producing.

[1]FM synthesis (Chowning, 1973) is probably the most frequently cited example of a simple patented synthesis algorithm. Some people prefer to say that Chowning *discovered* rather than invented FM since the technique was known from radio transmission and had in fact already been used in sound synthesis. Questions about the soundness of granting patents for software are prone to raise heated debates; see the thread at the music-dsp mailing list:
http://music.columbia.edu/pipermail/music-dsp/2011-February/069675.html

If a particular sound can be produced just as well with a much simpler algorithm, there is little reason to use the more complicated algorithm. Hence, we hope to construct synthesis models that are capable of producing complex and preferably evolving sounds without having to specify too much detail of the sound's unfolding. With this goal comes a complementary limitation: we will have to renounce complete control of every detail of the generated sound. Modifying certain aspects of it, say, the character in the beginning of the sound, may not be possible without also influencing the ensuing part.

A synthesis model is not determined from any single sound it may produce, but from the set of all sounds it is capable of. Apart from that, its utility is determined by the way the user can interact with it—how intuitive and easily operated it is. This should be understood as a matter distinct from the particular interface to the synthesis model, and how gestures or other means of control map to control parameters. Being dynamical systems, feature-feedback systems may exhibit a vast range of behaviour ranging from the simplest static character to very complex processes, including variation at several temporal scales, from which a large scale form emerges. A tentative definition of emergence in this context (until we delve deeper into the subject in Chapter 5) could be that it is the appearance of properties that one would not have expected when inspecting the rules that decide the sound's evolution. In a sense, then, emergence is related to the observer's knowledge of the system.

Although the abstract nature of autonomous instruments has been emphasised, it is not prohibited to look for similarities with natural or other known processes. Rather than deliberately constructing accurate and realistic models of existing objects or processes, we will see what remarkable phenomena may arise in feature-feedback systems.

1.2.3 Mappings and control strategies

In sound synthesis, it is common to distinguish an *audio rate* at which the audio samples are generated, and a *control rate* which is used for representing slower changes such as vibrato or dynamic envelopes. Synthesis techniques that use this division into audio and control rates are typically designed so as to yield static sounds unless the control rate synthesis parameters change over time. Synthesised sounds that lack variation over time tends to sound sterile, unorganic, and indeed *synthetic*. In some cases, this artificial effect is intended, as when an auto-tuner is employed to erase any individuality caused by pitch inflections in the singing voice. More often, however, the problem is to control the time-varying synthesis parameters expressively. There are several ways to acheive such an expressive control.

A synthesis model may be controlled in realtime by sensors, keybords or other gestural controllers whose signals are mapped to synthesis parameters. In synthesisers, the control functions may be generated algorithmically, beyond the musician's direct influence. The control data may also be derived from audio signals as in feature-based synthesis, by mapping feature extractors to synthesis parameters. Finally, the control data could be taken from any file or stream of data that is mapped to fit the synthesis parameters, which leads to the approach of sonification.

Mappings from sensors to synthesis parameters and from physical devices to computational algorithms have been much studied. Design questions about the human-computer-

interface have to be taken into account, including human physiology and cognition; these questions become more urgent than in offline algorithmic composition. As an illuminating example, consider an instrument described by Hunt et al. (2002) that has two sliders which control amplitude and frequency. Their original idea was to have each slider control each parameter separately, but by mistake another mapping resulted. Amplitude was now controlled by the speed of change in one slider, as if bowing a string instrument. Users found this to be way more engaging than the "correct" mapping. If the instrument is nondeterministic and functions more like a musical partner, then the concept of mapping becomes less applicable (Chadabe, 2002). Indeed, mapping as the term is used in mathematics means that for one particular input there will be one particular output, which is not the case in nondeterministic instruments.

If there is no sensor input to the instrument, then the control data has to be generated algorithmically or read from memory. Control data typically has a high data rate, albeit slower than the audio sample rate. Entering long lists of control data manually in order to make sounds with lively dynamics implies a prohibitive amount of typing. This problem faced computer music pioneers unless they used analysed input sounds, or generated control data algorithmically. Since it may be hard to find algorithms that produce the intended results, the ease of direct gestural control is sometimes a great advantage. The most immediate form of controller would be to play an instrument or sing into a microphone, track the pitch, amplitude envelope and other features, and map them directly to the corresponding synthesis parameters.

When controlling a synthesis model by another audio signal, the question arises of how input should be mapped to synthesis parameters. Although it appears logical to map pitch to pitch, and preferably in tune, nothing prohibits that the pitch contour is inverted or transposed as it is mapped to the synthesis parameter, or that it is mapped to a completely unrelated synthesis parameter, such as the waveshape. Similar cross-couplings of musical dimensions can also be made in feature-feedback systems.

Adaptive control of synthesis parameters through the analysis of an audio signal has the benefit that any sound can be used as input. Recorded sounds often offer the advantage over simple synthesis models that there is some inherent complexity and micro-variation in the sound that may be hard to model. This is particularly clear in recordings of musicians playing instruments that allow continuous inflexions of pitch, amplitude and timbre. Even if the mapping from audio features to synthesis parameters is altering the timbre and gestural contours so drastically that the original sound becomes unrecognisable, the small fluctuations may nevertheless survive, which can be very useful. Hence, adaptive synthesis has two complementary usages: a musician may use it to alter the timbre of his or her instrument, with as direct tracking as possible in other respects, or it may be used more freely as an input to synthesise novel sounds that may follow the general pattern of variation of the controlling sound.

Autonomous instruments may be modified to receive external control input, although that of course makes them non-autonomous. For example, an audio input signal analysed by feature extractors could be mapped to internal parameters of the (no longer quite) autonomous instrument. This may be both useful and interesting; however, we shall not pursue investigations in that direction except as we review work by others. Restricting our attention to autonomous and mostly deterministic instruments provides more than

enough questions to investigate.

1.2.4 Parameter spaces

As Landy (1991) observes, the parametric thinking that permeates so much twentieth century music is a crucial ingredient in most experimental music. With the 1950s integral serialism, this parametrical thinking reached its peak. Musical dimensions such as pitch, dynamics, duration, timbre, articulation, occasionally spatial position, density, disorder and others were treated separately. Although precursors as regards the independent treatment of musical dimensions can be found much earlier in the history of western music—Guillaume Machaut's isorhythmic motets are often cited examples—the requirement of integral serialism that no value of a parameter be repeated before the entire set of values are used, puts stringent restrictions on the musical possibilities.

In mathematics, the term parameter has a clear meaning: it is frequently used in functions that take on different shapes depending on the parameter. A parameter is also to be distinguished from a variable. In music, its usage is ramified and sometimes confusing. Landy (1991, p. 9) quotes a definition of Josef Häusler: "Musical parameters are all sound or compositional components which can be isolated and ordered".

But to order something does not necessarily imply to impose an ordering relation (such as "greater than") on the elements. There is the classification of scales known from statistics, where the distinction between continuous, discrete and nominal scales is especially relevant. There is some evidence that sometimes even nominal scales, that is, collections of elements that cannot be ordered in some increasing succession with respect to some criterion, are included in what is called musical parameters.

An ordering into categories, such as Pierre Schaeffer's typology of sounds (Schaeffer, 1966), would also qualify as a parameterisation according to Häusler's definition. (Schaeffer had other solutions than to speak of parameters of perceived sound, some of which will be discussed in Chapter 2). We will avoid using the term parameter for perceptual qualities of a sound and limit its use to physical or numerical properties of an algorithm or function.

Electronic music provides eminent opportunities for the control of parameters independently from one another. It is also easy to introduce new musical concepts, such as scales of densities or degree of disorder, that can be treated parametrically. In most of the synthesis models that have been designed to be intuitive to use, the relationship between synthesis parameters and perceived sound is perhaps complicated, but not inscrutable. One may even conjecture that it is precisely this not too distant relationship between control parameters and perceived qualities that leads to their frequent confounding. In autonomous instruments, however, the synthesis model is not always so intuitive; the relationship between parameters and sound may vary from straightforward to incomprehensible. A nonlinear synthesis model may have an interaction effect between parameters, so that the result of varying one parameter depends on the values of others. Additionally, hysteresis is a prominent trait of feature-feedback systems. This means that the path one has taken to arrive at a particular parameter configuration influences the resulting sound. It also means that the system may have very long transient periods before it settles into a relaxed state. And, in case the synthesis model be chaotic, it will by definition

be sensitive to minute changes of its initial condition. In the simplest chaotic systems, such differences will influence the systems trajectory, but on average, in statistical terms and in spectral terms, it may retain the same qualities. More complex systems such as feature-feedback systems may be capable of more differentiated behaviour, especially in terms of perceived sonic qualities. Hence, feature-feedback systems may be susceptible of more unpredictability in their sonic result due to sensitivity to initial conditions.

Another problem that feature-feedback systems share with several other synthesis models is the large number of parameters. Flexible digital instruments that produce anything musically interesting will tend to be complicated and have many parameters. With large parameter spaces, it becomes practically impossible to exhaustively explore the sonic potentialities. For one or two dimensions, it is easy to visualise the parameter dependence of a feature. Such strategies of investigation will be described in more detail in Chapter 6.

1.3 Aesthetics

Decisions about how to design the autonomous instrument do not arise in a void. Presuppositions about what music can and should be, and about what sonic characters would be worth bringing forth with a newly constructed instrument will shape the instrument building process. Here enters questions of aesthetics.

Music made with strictly autonomous instruments appears to be relatively rare, but there are a few examples of semi-autonomous instruments or systems that also shed light on aesthetic problems related to autonomy as well as categories such as "natural" versus "artificial". Finally in this section, experimental music will be discussed.

1.3.1 Practitioners of semi-autonomous instruments

So far, little has been said about how autonomous instruments have been used in musical compositions. The term "autonomous instrument" is not in regular use in the musical community. Awkward as it may sound, it was introduced for the lack of a better term for collectively describing a class of synthesis algorithms, or perhaps rather a certain strategy of music making. At an early stage of the current project, the terminology was different; feature-feedback systems were then called "self-adaptive synthesis" (Holopainen, 2009). Later on, it became necessary to refine the terminology and introduce the concept of feature-feedback systems, which we will develop in the form of autonomous instruments, although they might involve realtime interaction as well. There appears to be no well-known precedents of the use of autonomous feature-feedback systems in musical composition. Thus, we could leave it at that and skip the discussion of existing aesthetic practices related to music made with such means since there is in fact nothing to talk about. Instead, it seems warranted to broaden the view slightly and discuss some related endeavours: First, live interaction with feedback systems where machine listening is involved, and second, closed or non-interactive systems of algorithmic composition at the signal level without feature extractors. Feedback systems in general will also be discussed in Chapter 5. A third extension might be other relatively autonomous systems, either

digital or analogue, which make no use of machine listening or feature extraction in any form, but run almost on their own with little need for tending.

As said, there is no general agreement on terminology. We will use the term *semi-autonomous* to refer to systems that are interactive, but simultaneously doing more than passively and predictably reacting to the musician's input. There is also some precedence for that use of the term (Jordà, 2007). Indeed, there is a small number of names that crop up in several surveys of the more experimental interactive computer music scene (e.g. Collins, 2007b). From the analogue era, there is Gordon Mumma with his work *Hornpipe* (1967) for horn and live electronic sensing and processing, which will be briefly discussed in Chapter 8. Nicolas Collins' *Pea Soup* (1975) is another example of acoustic feedback loops in which the feedback is regulated as it is about to build up. George E. Lewis and David Behrman, among others, started to use microcomputers in the 1970s for simple pitch following and interactive response. The *Voyager* system of Lewis, which grew out of this is a note-oriented interactive system. Later on, when digital realtime processing became more accessible, several other composers and musicians followed. Agostino Di Scipio has already been mentioned; in a way his *Audible Ecosystemics* are a canonical series of works.

It would be wrong to assume that all these people share some common aesthetic values or that they describe their music in similar terms. For example, Mumma called his system cybersonics, Di Scipio refers to his audible ecosystems as *autopoietic* (Di Scipio, 2003), others mention *self-organisation* (Bökesoy, 2007) or simply adaptive live processing or *live algorithms* (Bown, 2011). Live algorithms are described as intended primarily for performance with musicians, but "also on their own or with other live algorithms" (Bown, 2011, p. 73), in which case they seem to qualify as autonomous instruments. Furthermore, Bown mentions the possibility that live algorithms may incorporate precomposed material and audio samples besides the "autonomous responsive behavior that is central to a live algorithm", although he admitts that "these elements may be seen as generally detracting from the autonomy of the live algorithm" (Bown, 2011, p. 74).

If all these trends and activities consolidate, perhaps the term used to describe them will include the word *ecosystem* (McCormack et al., 2009). However, at the moment we should resist lumping together all the mentioned practitioners into a single coherent musical movement.

The metaphor of an acoustic ecology lies readily at hand when describing some work with semi-autonomous instruments. This is particularly true of Di Scipio's audible ecosystems, which are seamlessly integrated into their sounding surroundings. Apart from natural analogies, there are however many disciplines to which semi-autonomous instruments can be related, including artificial life, acoustic ecology, complex adaptive systems and cybernetics. Therefore, the following discussion of nature and the artificial is only one among several possible perspectives that relate autonomous instruments to a broader context.

1.3.2 Aesthetics of nature

Suppose we have developed an autonomous instrument that produces a sonic output of astonishing complexity, comparable to an elaborate electroacoustic composition or

a soundscape recording. Not only can we hear the synthetic sounds as resemblances of natural processes, but we could also listen to environmental sounds as if they were a musical composition, to follow a proposition of Cage as well as the acoustic ecology movement. In other words, the comparison between nature and art can be made in either direction.

Comparing nature with artificial simulations of some aspect of it could be part of a scientific investigation, as in artificial life (Langton, 1995). Observation of phenomena as they happen to occur is not always an option, for example if unusual or hazardous conditions are required. In these cases simulation by simple models is often a viable method for gaining insight into the phenomenon. However, we will focus our discussion of nature and artifice to aesthetics.

Going back to Kant's Critique of Judgement (1790), we find an aesthetic that treats the human apperception of art and nature on equal footing (Kant, 2003). Since Kant, art has been the focus of aesthetic theories at the expense of nature, at least until quite recently. Gernot Böhme is one of those who have taken up the thread from Kant. He introduced a concept of *Atmospheres*, which describes an "in-between" phenomenon: that which stands between subjects and objects. Music is in fact a good example of the atmosphere concept. This is seen particularly in music that explores spatiality and functions as an environment, such as soundscape composition (Böhme, 2000).

The word "nature" has several meanings and usages. To give but a few examples, which is by no means intended as a comprehensive list:

First, it can be understood as the object of human curiosity and scientific experiments. Nature is subatomic particles, the big bang and the laws of physics and biology. Second, it is often conceived of as in opposition to culture and human artefacts, as the world untouched by man but populated by other species. A third meaning arises when we speak of the nature of a person or of a thing. These meanings are not mutually exclusive; rather they are facets of the same concept. Kant has the first meaning in mind when asserting that our faculty of judgement prescribes itself a law for reflection on nature; a law whose purpose it is to facilitate the perception of a comprehensible order in nature (Critique of Judgement; Introduction: Section V). In passages that discuss the dynamically sublime or beauty in nature, the second sense (nature as opposed to culture) seems closer to the point. As for the third meaning, it appears in Kant's discussion of genius as a prerequisite for being an artist and not merely a cunning craftsman: Genius is a talent and an innate property, by which nature provides art with its rule (§ 46; this and all the following paragraphs refer to the Critique of Judgement).

Kant compares art and nature in the following distinction (§ 43): Art differs from nature as making differs from acting. The product of art differs from nature as work differs from effect. Further on he gives the example of a carved piece of wood found in a marsh—we would not think of this as a product of nature, but of culture, he argues. But why is this? It seems necessary to separate humans with their artefacts as not belonging to nature, but standing beside, observing it. Yet, there is such a thing as the nature of a person, and even nature *in* a person. However questionable this divide between nature and culture, or between man and nature may be, it should be noted that if the concept of nature were to be all-encompassing, it would only be rendered useless.

According to Kant, art takes nature as its model, yet distinguishes itself from it.

Kant argues that nature is beautiful because it resembles art, and art can only be called beautiful as long as we are aware of it as art, while it nevertheless resembles nature (§ 45). On the contrary, if we believe that we are listening to a beautiful birdsong in the woods, but on closer inspection we discover it is an artificial bird singing (or a child imitating a bird), we would lose interest in the song or even find it annoying (§ 42). Here, our expectation and ensuing disappointment or a feeling of being deceived has a clear negative influence on our aesthetic judgement. Perhaps this mechanism is at play in situations where imitative sound synthesis is harshly judged and found wanting in its lack of naturalness.

What Kant sketches in the Critique of Judgement can be called an aesthetics of nature, in that it takes nature as the model for art. It is quite remarkable that this high estimation of nature as a model for art has been so prevailing since Kant's days, even to our time. Perhaps it is easier to enumerate those artistic trends and styles that denounce it—futurism springs to mind as one of them—than those that embrace it.

Adorno's diatribe against Sibelius is remarkable in its equating of the tempered scale with control over nature. Sibelius' music is qualified as subversive in the sense of destroying "... all the musical results of control over nature, which humanity bought dearly enough through the use of the tempered scale. If Sibelius is good, then the criteria for musical quality extending from Bach to Schoenberg—wealth of relationships, articulation, unity within diversity, plurality within the singular—are no longer valid"(from Glosse über Sibelius, quoted in Adorno, 2006, p. 236). Not only the tempered tuning system, but notation as well belong to the cultural sphere rather than to nature: "Musical writing is the organon of music's control over nature", Adorno says in a discussion of notation and its power to store and recall "the gestures which music either stimulates or itself imitates" (Adorno, 2006, p. 173). If control over nature is a theme in Adorno's writings, self-organisation as a structuring principle for art appears to be an idea that later entered musical aesthetics. Autonomous instruments are typically not entirely controllable. Inasmuch as they retain an uncontrollable aspect, their resilience can be seen as *natural*.

Maybe it is because the term "nature" carries so many facets of meaning, that aesthetics of nature seem so prominent. Are there in effect several parallel aesthetics of nature? Consider for example when painters turned from more or less naturalistic painting to abstract expressionism—instead of depicting external nature, they turned to their "inner nature" as an artistic source. Thus "nature" is a complex concept, seemingly self-contradictory when understood as a specific entity, but less confusing if taken as referring to the essence of a thing.

As has been pointed out, autonomous instruments do not try to model or simulate any particular aspect of nature. Still, it may be relevant to consider what known phenomena (if any) they resemble. In particular, emergence and self-organisation are hallmarks of living organisms, although not limited to them. If we observe self-organisation in an autonomous instrument, it may be tempting to rise to the level of abstraction where it makes sense to compare it to living organisms, which opens the possibility for an aesthetics of nature. So, even if the aim is not to model nature, the complexity of the result may be the crucial aspect that leads us to think of the autonomous instrument partly in terms of nature.

Indeed, direct imitation or reproduction through recordings of nature are not the only ways to expose an aesthetics of nature. Writing about the GENDYN program of Xenakis, Hélène-Marie Serra (1993, p. 239) notes:

> For Xenakis, the question of the approximation of instrumental sounds and natural sounds is secondary. His primary intention is to (re)create the variety, the richness, the vitality, and the energy that make sounds interesting for music.

A similar view is espoused by Perkis.

> The music is seen not primarily as implementing a vision of the composer, or the will of the composer—something the composer hears in his head. Rather it's about setting up situations that allow the appearance of sonic entities that are more like natural phenomena than traditional music. The practitioners of this type of music build machines, or things akin to machines or simulations, things that have a behavior of some kind that is unanticipated by the composer (Perkis, 2003, p. 76).

One should also keep in mind the sterile appearance of many early attempts of digital sound synthesis. The sounds often lacked nuances and small-scale variation. Edward Carterette posed the question, is electroacoustic sound unnatural, and argued that the reason for "computed music" to sound mechanical is that "the generators used are not made up of [a] complex of interacting components as in the case of a real instrument" (Carterette, 1989, p. 91).

If nature is the complement of man, that is the locus where art-as-nature may be surprising even to its own creator, and not merely function as a vehicle for self-expression. In much experimental music, the process (of performance, of interpreting a score, or as generated by an electroacoustic device) is considered more important than the resulting sounds. In the case of interactive music, this opposition is notable as *"a shift from creating wanted sounds via interactive means, towards creating wanted interactions having audible traces.* In the latter case, one designs, implements and maintains a network of connected components whose emergent behaviour in sound one calls *music*" (Di Scipio, 2003, p. 271, emphasis in original).

Emergence often seems to carry a positive connotation, at least in music. It may be that the concept has more often been associated with processes such as organic growth from a seed to a beautiful plant, rather than say the emergence of war from a series of trivial conflicts. Strictly speaking, emergence should be understood as value-neutral. But there is another side to its appreciation, depending on the observer's knowledge or ignorance of the system.

Given that the observer cannot predict the global behaviour of a system from its rules or from knowledge of its components alone, there will be a surprise effect that we may assume to be a positive experience. However, the opposite situation is equally plausible: A sound produced by a synthesis model may appear simple or dull to a listener who is not aware of the relative simplicity of its generating mechanism. For someone who is aware of what is going on inside the synthesis model and knows that very simple rules guide its behaviour, it may appear the more impressive that its sonic results are not much

simpler. This is of course a problem of aesthetics that composers need to be aware of, even though it can be difficult to cultivate a listening attitude that is unaffected by one's knowledge of the underlying processes that go into the creation of a sound.

1.3.3 The artificial

The category of the artificial may come with positive as well as negative associations. An artifice may fail to appear natural, whence it appears as a deficit, or it may be intended to draw attention to its underlying construction.

Acoustic sounds such as animal vocalisations, musical instrument tones or song appear natural to us, but maybe there was a time when newly invented acoustic instruments were perceived as unnatural. We may still experience some of that sense of wonder when listening to Conlon Nancarrow's humanly unperformable player piano studies. Then came analogue electronic instruments, and the otherworldly sounds they brought with them. Early *electronische Musik* had a peculiarly rough edge to it, as technical limitations did play a significant role in what kinds of sounds were attainable. When the early experiments with digital sound synthesis were begun, sounds usually had an even more sterile appearance than music made with analogue equipment. This is readily explainable in acoustic terms: small instabilities and noise in analogue instruments provide some amount of micro-variation that is not necessarily present in digital synthesis unless deliberately introduced. Today, analogue synthesizers are regarded as venerable collector's items, and there is much talk about the "warmth" of their sound, whatever that is supposed to mean. It appears unlikely that they would be held in such high esteem, were it not for the introduction of (less successful) digital instruments that showed us how harsh and artificial sounds could actually be made.

Along with digital technology came a certain repertoire of malfunctionings such as skipping CDs. That particular sound was soon to be picked up as a signature element in the glitch genre (Cascone, 2000). Failures of all sorts were now exploited as new sources of musical material. But the rhetoric around glitch can be misleading; if a sound can be a resource for a piece of music, then no matter how badly some mechanism had to fail in order to produce that sound, it at least served perfectly well as material for music.

An early piece soundwise reminiscent of other glitch is *Clicks* by Tim Perkis, who was a member of the pioneer computer network ensemble The Hub. Perkis writes about *Clicks*, a short piece that, if any, qualifies as artificial:

> I like to think of this piece as unlike any other; it in fact is not a sound recording at all. [...] But this piece is just a pattern of digital information, generated by a special purpose signal processing computer I built [...] I hope it reveals the immediacy of the physical situation: you have a piece of electrical equipment which is flicking around a cardboard cone. The speaker is making the sound, and I suggest you listen to the speaker, rather than listening "into" it for the spatial illusion all sound recordings create (Bischoff and Perkis, 1989, liner notes).

Doing away with any illusions of naturalness, as in this case of imagined spatial origin, is a common strategy when artificial expressions are embraced. As the title indicates,

Clicks consists of a layer of loud clicks, together with a background layer of dynamically softer sustained tones. Apparently, it qualifies as an exemplar of music made by an autonomous instrument, that is, by algorithmic sound synthesis, though it is hard to tell whether there was some direct interactivity involved in the making.

Further examples of electronic pieces that avoid any gloss in the form of added reverberation and other processing that might bestow some kind of naturalistic spatial illusion can easily be found. Such strategies are used, for example, in Herbert Brün's *Sawdust* pieces, in Xenakis' *S.709*, in G. M. Koenig's *Output*, and in David Dunn's *Nine Attractors* (we will return to several of these works). Of course, the reasons for employing such dry mixing may vary, but at least in the cited examples, it appears likely that a belief in the value of bringing out the raw, unvarnished signal directly from the algorithm is a common motivating factor. If so, then mastering the pieces for release on CD and glossing over the rough edges with a nice reverb would be a betrayal of the composer's intent.

Dunn made a series of site-specific compositions to be performed outdoors that nicely illustrate our discussion of the natural and artificial. Musicians playing acoustic instruments and electronic sounds were recorded along with puffs of wind, passing airplanes and bird calls. The contrast between artificial and natural is highlighted in *Mimus Polyglottos* (1976) for electronically generated sounds and mockingbird. The electronic sounds resemble some kind of bird call, but their timbre, lack of variation and perhaps persistence (as if never breathing) belies their artificial origin. A mockingbird answers the one-way dialogue—of course the recording is fixed, and cannot respond back. "It was a pre-recorded tape because I didn't want to be accused of reacting to the bird. I wanted to see what the bird's reaction would be to the stimulus", Dunn said in an interview (Dunn and van Peer, 1999, p. 64).

With the development of machine listening, more interactive versions could certainly be made, where artificial sound-producing agents enter into conversation with natural environments and biological life. Apart from the machine listening component, these ideas were the basis for Dunn's environmental interactions. Interestingly Dunn denounces the separation of artificial and natural, stating that "... we assume these technologies to be unnatural. In fact, the natural world, other forms of life, find them as fascinating as we do" (Dunn and van Peer, 1999, p. 66).

1.3.4 Experimental musics

It is not always clear what is meant by experimental music, although it often seems to boil down to a certain attitude towards musical activities. In the 1950s, the term was used in a pejorative sense by some music critics, implying unfinished "trial runs of untested materials and methods" and that "the composers have not mastered their methods" (Mauceri, 1997, p. 189). As Mauceri also points out, experimental music today is characterised by the radically new, but is nonetheless a tradition in its own right.

Too often, the term experimental music has been used without any hint of a definition. Not so with Michael Nyman, who firmly opposed it to the avant-garde, represented by composers of the Darmstadt school who developed integral serialism in the 1950s (Nyman, 1999). Before laying out Nyman's views on experimental and avant-garde music,

it is worth noting that his conception of the avant-garde does not quite rhyme with that of some other writers. For Peter Bürger, the avant-garde is actually limited to historical movements such as Dadaism and surrealism, characterised by various attempts at bridging the gaps between life and art and even destroying art as institution and revolutionising society (Bürger, 1984). Later movements, such as pop art, Bürger labels as neo-avant-garde. Bürger criticises these movements for institutionalising the avant-garde as art. On the other hand, what Nyman characterises as avant-garde is usually described as modernism, no more, no less. Incidentally, the appearance of the first editions of Nyman's and Bürger's books roughly coincided; both appeared in 1974.

The avant-garde has a complicated relation to tradition, artistic trends, society and the market. With an oblique reference to the Nobel prize of 1969, Umberto Eco remarks: "All right, now that Samuel Beckett has had the Stockholm treatment, the word 'avant-garde' can hardly keep its meaning" (Eco, 1989, p. 236). Further, "... today, 'to be avant-garde' may well be the only way of belonging to a tradition", but "the artist is a rebel because the market wants him to be one" (Eco, 1989, p. 266). Twentieth century aesthetics with all its short-lived isms has seen several avant-gardes, soon to be overthrown by others that prevent the previous avant-garde from becoming manner, as Eco also points out.

Now, to return to Nyman, the list of composers encountered in his book is a remarkable collection. It seems utterly implausible to include the indeterminate pieces of John Cage, repetitive minimalist works by Philip Glass and Steve Reich, and representatives of Fluxus and Cornelius Cardew's explicitly political expressions in a single genre. And why are not Boulez and the other European serialists who frequented Darmstadt included?

> Experimental composers are by and large not concerned with prescribing a defined *time-object* whose materials, structuring and relationships are calculated and arranged in advance, but are more excited by the prospect of outlining a *situation* in which sounds may occur, a *process* of generating action (sounding or otherwise), a *field* delineated by certain compositional 'rules' (Nyman, 1999, p. 4, emphasis in original).

In this listing of characteristics to look for in music by an "experimental composer", there are only a few that need concern us in the context of autonomous instruments. Processes and certain kinds of compositional rules are of importance here—but not just any type of process or rule.

Chance procedures were developed by Cage in an attempt to reduce the role of ego in his music and replace it with "non-intentionality". At first, chance operations were employed to produce notated scores, but later on Cage left much more choices to the performer's discretion. Chance operations are not the only means to obtain unforeseeable events. Although the output of an hitherto untested autonomous instrument cannot usually be foreseen, it would be wrong to confound it with chance operations unless it expressly includes randomness.

Other types of processes described by Nyman involve performers following prescriptions with some degree of openness, and electronic processes are also mentioned. In fact, a piece could be defined as any sounds made by an electronic circuit, hence the instrument becomes the composition, and not only the means to bring it to life. The identity

of a composition is usually taken for granted—with the proviso that each performer adds something personal to its interpretation, within stylistic limitations. "But identity takes on a very different significance for the more open experimental work, where indeterminacy in performance guarantees that two versions of the same piece will have virtually no perceptible musical 'facts' in common" (Nyman, 1999, p. 9).

Even if the work is not fixed in all details, a set of general procedures to be followed by the performers may result in a coherent and certainly not random musical structure, as Brian Eno has argued in the case of Paragraph 7 of *The Great Learning* by Cardew (Cox and Warner, 2004, pp. 226-233). The crucial instructions are not to sing the same tone on two consecutive lines of the text, and to sing a tone that you can hear (i.e. that someone else is singing). According to Eno, the performers contribute much structure to the work by making mistakes, adjusting to each other, letting aesthetic preferences guide them and adjusting to the acoustics of the room. In accordance with what the score allows, hypothetical performances would be capable of much variation, but as Eno explains, there are several variety reducing mechanisms at work.

Much experimental music has been concerned with the invention of rules for the performers to follow. Verbal instructions may replace the conventional notated score. In some cases, the instruction details strategies to be followed in otherwise open situations, as in Frederic Rzewski's Sound Pool (1969), where "... each player makes only one simple sound and regulates his loudness level so that it is slightly louder than the softest sound he hears around him. He then moves towards the softest sound or conversely away from any loud sounds nearby, continuously adjusting the loudness of his sound" (Nyman, 1999, p. 131). Similar strategies could become useful as design principles for autonomous instruments, particularly so in cross-coupled models, where the entire system can be thought of in terms of agents that listen to each other and react according to a chosen strategy.

Still, there are other views on what experimental music is. Leigh Landy (1991) identified four different usages of the term:

1. Avant-garde music, in being innovative and 'before its time' is seen as synonymous with experimental music. However, as so much has already been tried out, it would seem to become increasingly unlikely to encounter anything genuinely new in music, and hence the avant-garde concept becomes problematic.

2. Lejaren Hiller and Pierre Schaeffer independently defined experimental music as the result of laboratory work, as specifically electronic music made in the studio (more will be said about Schaeffer below).

3. According to Cage, an experimental action is such that the outcome cannot be foreseen. This view of experimental music is the one embraced by Nyman, as described above.

4. *"Experimental music is music in which the innovative component (not in the sense of newness found in any artistic work, but instead substantial innovation as clearly intended by a composer) of any aspect of a given piece takes priority above the more general technical craftsmanship expected of any art work"* (Landy, 1991, p. 7, emphasis in original).

This fourth definition is the one that Landy clings to, but it remains to be defined as to what is innovative in music. The question is complicated by the fact that there is today a tradition of experimentalism in music, as witnessed in the recent publication of a research companion to experimental music (Saunders, 2009) and in frequent references to that tradition by many contemporary composers.

An historical lineage of experimental music is suggested by Tim Perkis (2003) in an insightful essay. Perkis distinguishes two very different attitudes towards composition which must be understood as ideal types rather than likely descriptions of any individual composer. These are the romantic self-expression on the one hand, and an experimental attitude on the other.

> The role of the composer in the experimental view is in a sense more passive than that of a romantic composer: once set in motion, the music has its own life and the composer is a listener like any other. Calling this music experimental is quite precisely correct: like a scientist setting up an experiment, the experimental composer sets up the conditions that define the piece, and is interested in hearing what actually happens with it (Perkis, 2003, p. 76).

These "conditions that define the piece" include anything from devising algorithms to building electronic machines or programming computers for sound generation. The historical heritage of American experimental music that Perkis traces reaches from Ives and Henry Cowell through Cage and on to Alvin Lucier, David Tudor and others, but according to Perkis, the pedigree extends back to French impressionism, and even to Rameau. Looking back through western music history, many great changes have occurred that may have been caused by a curiosity typical of an experimentalist. Nevertheless, it would be unfair to claim that the self-expressing attitude has not had an equal influence on music history.

Many of Lucier's works are at the logical end-point where the composer takes a very passive role after having invented an experimental situation. For a scientist to influence an experiment in some preferred direction would be unacceptable. Inasmuch as the same holds for the experimental musician, this profound respect for the experiment's outcome unites the artist with the scientist. In the end, experimental music is not science but remains an artistic expression, as Perkis admits. Romantic self-expression and radical experimentalism as idealised positions are useful concepts, but most artistic activity takes place somewhere along this continuum rather than at one of the extremes.

Pierre Schaeffer's version of experimental music is different from that espoused by Tim Perkis, but not opposed to it. No dilemma of self-expression or not arises from Schaeffer's methodology.

For Schaeffer, the pair of concepts *faire/entendre* (to make/listen) are closely related (Schaeffer, 1966; Chion, 1983). One has to practise music making—that is, playing, producing and composing it, and then one also has to practise listening critically to it. Behind these concepts there is a critique of a blind reliance upon compositional techniques—i.e. a sophisticated *faire* without an equal engagement of *entendre*. What Schaeffer meant by experimental music should be seen in relation to this dual concept. Here the experimental with its switching back and forth between making and evaluative

listening stands in opposition to a music made "a priori", which Schaeffer blamed the serialists for practising.

In the case of music made with autonomous or semi-autonomous instruments, the picture is complicated by the need to distinguish fixed composition from live electronic and interactive works, as well as treating open form composition separately from a determined or fixed form. It would be easy to adopt the strategy of *faire/entendre* in the case of fixed compositions, which necessarily have a determined form. By contrast, in open form works such as Di Scipio's *Audible Ecosystemics* series of works, the final musical form depends on contingencies of the particular performance and cannot be known in detail in advance. Consequently, it seems that one would have to renounce any ambitions of refining the shape and sonority of the composition by a cycle of making (in this case, programming) and listening. In fact, although a detailed control of the temporal unfolding is no longer feasible, it should still be possible to appreciate what are the probable sonic events and processes, and by experimentation arrive at a setup that is likely to produce a certain kind of musical behaviour.

Finally, how does composition by autonomous instruments relate to experimental music? Aesthetics inevitably have to inform choices of how autonomous synthesis models are constructed. Since the unpredictability of these instruments has been emphasised, one could expect Cage's sense of actions producing unforeseen consequences to be relevant. Indeed it can be, but it is worth pointing out that there are many conceivable usages of autonomous instruments, not all of which are necessarily experimental in a strong sense. An important ambition for this thesis is precisely to tease out the secrets and to make what initially seems unpredictable less so, eventually arriving at a more complete understanding of autonomous instruments in terms of dynamic systems. Perhaps the safest mode of usage is to experiment with these synthesis models, generate a collection of sound files and build up a piece from that, in effect treating the source material as any found sound objects. However, what one must not expect is a close simulation of any hitherto practiced musical style or instrument—at least not without significant efforts to achieve precisely that.

1.4 Instruments and composition

To speak of *instruments* that are scarcely interactive at all may to some seem an abuse of terminology, though the historical precedent comes from the MUSIC N family of languages for sound synthesis constructed by Max Mathews starting in the late 1950s (Roads, 1996; Boulanger, 2000; Manning, 1993). There an instrument is just a specification for how to connect various unit generators into a system that takes input parameters from a score file and outputs an audio signal. Such instruments, like our autonomous instruments, were not played in real time, but only controlled by specifying parameters in the score file. Here we will try to clarify the relations of autonomous instruments to more interactive instruments, as well as the relationship to algorithmic composition.

1.4.1 What is an autonomous instrument?

Musical instruments are typically made for being manipulated by a musician in real time. In some cases the notion of a musical instrument has been extended to situations where some of the actions do not occur in real time. Examples include the preprogramming of a sequencer to accompany a performance, as in automated score following (Orio et al., 2003). The electroacoustic studio has been referred to as the composer's instrument. The turntable is also often considered a musical instrument, in spite of the fact that very much of the performance's content is already engraved in the record's grooves and not provided by the DJ. In that sense, turntables are open-ended and extensible instruments (Tanaka, 2009). Even further removed from the musician's direct influence are mechanical instruments, such as the music box and the pianola. Computers permit a multitude of ways to construct automated instruments.

With these cases in mind, autonomous instruments may be tentatively defined as *an automated instrument which the musician does not control after having started it.* An autonomous instrument can be perfectly deterministic, so as to yield the same output for a particular initial state. Then, different pieces could be generated by specifying different initial conditions. If there is only one allowed initial state, then the instrument reduces to a fixed composition.

In certain kinds of generative music, algorithms shape each particular instantiation of a composition into a unique realisation which depends on a random seed (Collins, 2008a). Given the same random seed, the instrument would produce identical output, but the point is that this number is randomised before running the algorithm, so as to yield new versions each time. However, the instrument may also be sensitive to its environment in such a way that external events influence its sound-producing process. These are two different ways in which non-determinism may enter an automated instrument, and of course they may occur simultaneously. Now, if we again let a musician interact with this instrument, we have come full circle and can no longer speak of an autonomous instrument.

There is, however, a more subtle sense in which one can speak of autonomous instruments. A digital instrument can be designed so as to allow a certain degree of external influence while maintaining an equilibrium state. In this case, there is an ongoing adaptive process that makes the instrument autonomous in this second sense. Alice Eldridge has made some use of a homeostat model that behaves in a similar way (Eldridge, 2008). To avoid any confusion, such systems will here be classified as semi-autonomous.

According to the terminology of dynamic systems, *autonomous systems* are such that terms that are explicitly functions of the time variable do not occur, in contrast to driven (or forced) systems, in which an external signal influences the dynamics (e.g. Strogatz, 1994). As it happens, most of the feature-feedback systems that we will introduce are autonomous systems (or can easily be formulated as such), although this is rather a matter of choice than a necessity. However, if a time dependent signal such as noise is added to the system, it becomes non-autonomous in the dynamic systems sense, regardless of whether or not it allows for real-time interaction.

Words such as "autonomy" or "autonomous" are used in various contexts with different meanings. When we speak of autonomous instruments, this refers to non-interactive in-

struments, but another widespread meaning of "autonomous" is, approximately, a system or process that is capable of maintaining its existence and independence while interacting with an environment.

To quote an example from Atau Tanaka (2009), "autonomous" may also be synonymous with "self-contained" in the same sense as an orchestral instrument such as the violin or the oboe does not need any additional equipment to become functional. The autonomous instrument needs no record collection as the turntable does, it needs no amplifier as the electric guitar does. Thus, it is purely a matter of delimitation before one can say what is a self-contained instrument; for example, the electric guitar with amplified considered as a unit would be a self-contained instrument.

Moreover, Tanaka (2009, p. 238) provides a useful analysis of the components of open-ended systems such as digital musical instruments. The typical components are:

- An input device, usually some kind of sensor that captures the performer's gestures, or a microphone;
- A mapping algorithm that translates the gestural input into data that can be used in the rest of the system, in the sound-producing part of it;
- A sound synthesis engine whose synthesis parameters are controlled or influenced in real-time by the sensor data;
- Compositional structure similar to a score, a structural layer that co-determines the progression of the music;
- An audio output consisting of digital-to-analogue converters and spatialisation to any number of channels.

Whilst it must be remembered that autonomous feature-feedback systems are neither real-time interactive, nor musical instruments in the usual sense, they can easily be obtained by some non-standard reconfigurations of the above listed components. Thus, the audio output has to be routed to the input device, but the mapping from input to synthesis parameters will be kept as is. The sound synthesis engine generates its output as usual, but the compositional structure has to be replaced with an algorithm that is actually situated in the mapping between audio output and audio input.

Although we shall usually have computer programmes in mind when referring to autonomous instruments, there are other electronic systems with a conceptual resemblance such as no-input mixers. This is a relatively recent and not yet much documented trend in the fringe electronic music scene, where the mixer is used as a sound source by feeding its output directly to its input and possibly using effects processing. No-input mixers are not necessarily autonomous instruments however, insofar as they may also be played in real-time. However, analogue synthesisers, mixers, and other non-digital gear can be routed into autonomous instruments. If the actual practice is restricted to very sparingly adjusting sliders or knobs and letting most of the process unfold of its own accord, it becomes a mere hairsplitting matter if one likes to call it an autonomous or a semi-autonomous instrument. Nevertheless, since we will actually develop fully autonomous instruments, it is sometimes useful to be able to make the distinction.

A handful of composers and musicians currently experiment with no-input mixers, home-built electronics and analogue synthesisers; these are idiosyncratic feedback systems that have a behaviour of their own that is often difficult to understand and to control. Almost invariably, there is some form of direct interaction. After all, strictly autonomous instruments offer an austere mode of interaction—that circular sequence of programming, debugging, running the programme and listening to its output in the case of digital instruments, whereas most practical applications have included some form of realtime interaction that makes the system semi-autonomous (see Section 1.3.1 above). That said, it appears that some practitioners such as David Tudor or the network ensemble The Hub developed a musical praxis that included relatively sparse direct interaction with the system or algorithm that engendered the music. In their performances, most of the time very much appears to be going on, often with a human improvisatory quality which might mislead a less well-informed listener into believing that these phrases were performed in real-time by the musicians. Seen in that light, there are a few examples of music making using nearly autonomous instruments, or instruments that are only gently supervised.

The workflow typical of digital autonomous instruments fits into what Curtis Roads (1996, p. 845) calls "batch mode interaction"; furthermore, he notes the problem that after running the programme one has to either accept or reject the complete output, which might be an entire composition. If the algorithm is stochastic and the programme outputs different results each time it is run, then one cannot simply alter some small piece of the code and hope to generate a better version that fixes some problem from a previous run. The same problem often occurs in feature-feedback systems.

1.4.2 Varieties of interaction

Table 1.1 summarises three broad classes of instrument models. The left-hand column contains all the traditional and novel acoustic instruments, in addition to electronic instruments that respond in a deterministic and immediate way to the musician's actions such as electric guitars or simple synthesisers. Semi-autonomous instruments (middle column) are usually digital instruments, but may also be built from entirely analogue components. This category corresponds to what Jeff Pressing (1990) has called intelligent instruments. Semi-autonomous instruments engage the musician in a dialogue as it were, in other words, they offer a *conversational* mode of interaction (Eldridge, 2008; Chadabe, 2002). For this to happen, either some kind of machine listening or other sensors is a vital part of the technical apparatus. Lastly, autonomous instrument are really a variant of algorithmic composition where the composition takes place on the signal level rather than the symbolic note level. The instrument is a piece of programme code (such as an instrument in a Csound orchestra), though in order to be a truly autonomous instrument, we insist that it does not offer real-time interaction.

Substantial activity is today directed at experimentation with novel interactive digital instruments (as witnessed in the NIME conference[2]). Practitioners of these new interfaces for musical expression may rightly feel that there are a few categories missing between the columns with acoustic or electronic instruments and semi-autonomous instruments, but

[2]New Interfaces for Musical Expression, http://www.nime.org/

Instrument	Acoustic / electronic instrument	Semi-autonomous	Autonomous instrument
Medium (Technique)	Acoustic, electronic	Digital (Analogue) + machine listening	Digital instrument (MUSIC N type)
Interaction	Interpretation, improvisation	Improvisation	Algorithmic composition
Interface	Mechanical	Gestural controllers, Sensors, Live coding	Computer programming
Speciality	Direct human control and expression	Conversational mode	Self-organised sound (Emergence)

Table 1.1: Broad instrument classification according to the kind of interaction they offer. Only typical media, modes of interaction and interfaces are listed.

we will not focus on that side of the continuum between direct and indirect control. Many would agree with Magnusson (2009) that digital musical instruments offer something qualitatively different than acoustic instruments, not least by incorporating some amount of music theory into their design. Owen Green (2011), however, made the case for a continuum between acoustic and digital instruments.

Self-organised sound and emergence are listed in the table as the specialities of autonomous instruments, but there is frequently talk about self-organisation in interactive systems as well. For the moment, we take the simplistic view that self-organisation is a form of organisation that arises in a system without any influence from its environment. If there is an external influence, the situation becomes more complicated. Likewise, emergence is a popular qualifier for describing unexpected phenomena in certain semi-autonomous instruments (see Chapter 5).

In contrast to the buzzing research and artistic activity around interactive digital instruments, "autonomous instruments" is not an established term in the musical community. Thus it is fair to say that autonomous instruments in the most uncompromising form are fringe phenomena in the present-day musical environment—even in relation to the already quite marginal electroacoustic music scene. Some of the reasons for its lack of popularity are easy enough to glean: interaction with instruments that provide immediate response may be a more rewarding experience for many musicians.

Maybe the whole instrument metaphor is flawed when applied to autonomous instruments. After all, they are sealed from realtime interaction, and live as if they were in a closed world. On the other hand, autonomous instruments may be thought of as a specialisation of algorithmic composition to operate on the subsymbolic level. Traditionally, algorithmic composition by computers has been carried out in an offline mode. The programme is written, compiled if necessary, and run. Its output may be a list of

numbers representing pitches, durations and so on, or it may be a MIDI file or common practice notation. With faster computers, algorithmic composition can be turned into live coding (Nilson, 2007). Now the human programmer becomes the slow partner in the game. By either live coding, or gestural interfaces, or other sensors, some aspects of autonomous instruments may be brought under realtime control. Therefore, by softening their autonomy, they may participate in music making in the conversational mode. The conversational, semi-autonomous way of music making will be covered as we review some systems that others have developed. Strictly autonomous instruments may not solve any musical problems better than more interactive tools, but they do offer an opportunity to study self-organisation and complexity as it arises from a closed and deterministic system.

The recent practice of live coding provides a middle way that combines some of the immediacy of live performance with the precision of programming. Evidently, live coding may become another approach to performance with autonomous instruments that are made semi-autonomous, although this possibility will not be further investigated here. Generative music provides another perspective on autonomous instruments, but we will save the discussion of that for Chapter 8.

1.4.3 Composed instruments

Whereas real-time synthesis can generate a class of sounds without any signal input, real-time processing requires an input signal. Hence, live processing of sounds produced by a musician necessarily takes place after the event. A consequence for much live-electronic music is that the electronically processed sound has the compositional character of an echo—be it literal or drastically distorted. Delay lines or sample buffers facilitate this mode of operation, while any reversal of the order—the computer taking the lead, perhaps distortedly imitated by the performer—is more difficult to achieve and requires much more planning. Thus, from a birds-eye perspective, much live-electronic music with realtime processing shares the formal property that the listener shall know to expect a repetition with possible modifications in the electronic part of what has just been played live by a musician.

Futile as it is to accuse live-electronic processing of being incapable of reversing the arrow of time, the point is that there are very general limitations to what any particular type of musical technique can be used for. These limitations are frequently taken for granted, and in the case of live processing there is of course nothing to be done about it. Feature-feedback systems are constrained by other limitations. It appears that the use of feedback, with some delay due to the length of the feature extractor window, is an important constraint that defines what is possible in these models. The lack of interaction in autonomous instruments is of course a severe constraint. Even the addition of the slightest control interface that allows kicking the system out of an equilibrium would probably turn many systems that produce dull output when used strictly autonomously into quite versatile musical instruments.

As noted in the discussion on experimental music (Section 1.3.4), an electronic device may be built to function as a musical composition in its own right. There is no need for a score, since the musical unfolding is generated by the way the device functions.

While analogue circuits can be custom built to function as autonomous instruments, it seems fairly uncontroversial to claim that digital instruments, particularly programmable computers, are much more flexible and easier to reconfigure.

Nowadays, laptop computers are frequently used in live musical performances, and generally accepted as new members in the family of musical instruments. Acoustic instruments provide the prototype, easily extensible to electric amplified instruments and analogue electronic instruments, so that finally the sound producing computer fits logically into the line. Of course, the computer is at once both more and less than an instrument. It may have programmes installed that are not used for musical purposes at all—is the laptop performer actually checking his e-mail? But the computer is not very useful as such without the addition of a soundcard, loudspeakers and interactive control devices (taking the computer for a musical instrument is somewhat like speaking of a building as a musical instrument, just because there happens to be a piano in one of its rooms). Nevertheless, "musical instrument" is a metaphor among several others that fits some modes of interaction with the computer. Various metaphors of the musician-computer interaction occur such as: playing an instrument, playing together, conducting, or playing as a one-man band, executing, navigating timbre spaces, mixing (Schnell and Battier, 2002), and even *negotiating* in case the computer is particularly willful (Lewis, 1999). Clearly, this wealth of locutions reflects the allround character of computers, being configurable to satisfy a large range of needs.

Computers used live in musical performance typically combine the functions of a sound producing mechanism and a score (or other rule-following decision mechanism whose function corresponds to that of a score) that determines certain aspects of a musical composition. This dual function motivates the neologism "composed instrument" (Schnell and Battier, 2002). While the term might signify an instrument composed of parts, the meaning of an instrument that incorporates a score-like function is more to the point.

In a composed instrument, the sound producing part and the gestural interface are decoupled. Such a decoupling is generally not seen in acoustic instruments. A consequence of this functional division is that the same sound generating technique can be applied in different instruments, that are controlled through different gestural (or other) interfaces. Both the theremin and the ondes martenot share the same sound generating mechanism, but their control interfaces turn them into two separate instruments with their particular characteristic usages. In digital instruments the decoupling is as evident as it is unavoidable.

Composed instruments exist as a crossing between musical instruments in a narrow sense, and a computational device that is more or less apt to being controlled—from triggering an entire work by pushing a button, to chiseling out and being directly responsible for every minute nuance of the sound, as for instance a violinist must be. However, it is between these extremes that the really interesting possibilities lie: the instrument does respond to the musician's actions, but also contributes with something of its own. Another, albeit similar, way to set up this division is to distinguish between the virtuoso instrument and the amateur instrument. It is a subject of continuing debate whether an instrument can be designed so as to be both expressive and flexible, while still being playable by a novice. According to Jordà (2004), a good instrument is one that should be able to produce terribly bad music rather than imposing its music on the performer.

In light of the fact that composed instruments partly incorporate a score function, or similar specification of constraints that apply to one particular work, it is not so surprising that these instruments are rarely generic, but more often developed in conjunction with a specific composition. Certain objects or functions (subroutines) may be reused in other compositions, but the instrument as such has become over-specific, and might just reproduce a previous work, perhaps in a slightly modified form, if used again. A striking example is the works of Xenakis created with his GENDYN programme (Xenakis, 1992). Although it is virtually impossible to recreate the original pieces, upon experimenting, one soon discovers parameter settings that will produce something very reminiscent of Xenakis' original.

1.4.4 Build your own

The commercial music technology industry has contributed to the development of new instruments, such as analog and digital synthesisers, and effect processors. These instruments have been designed to be user friendly, which would be to say intuitive and easy to operate (Chadabe (1997) gives several examples). But this preoccupation with the alleged needs of the average user often compromises flexibility and versatility, as Risset (1991, p. 37) already observed:

> A few years ago commercial digital synthesizers were limited to the pitches of the tempered scale. Technological progress and commercial logic should not entail musical regression. When musical systems are made too user friendly, too easy to use, they usually restrict the possibilities to a menu of operations that may severely hinder creativity.

The trend towards user friendly design has hardly declined, though it may be mistaken to assume that this tendency comes to the dismay of the typical user. In fact, Chadabe (1997, p. 258) quotes an advisor for a synthesizer producing company who had noticed that when instruments came in for repair, the factory presets were rarely changed by the user; the users apparently had been content with using those preprogrammed sounds that came with the synthesiser. What those users needed the synthesiser for was predominantly its sound emulating capabilities.

The current interest in circuit bending appears to reflect a certain boredom with the too easily attained results provided by commercial synthesisers, combined with their non-unique character of a marketed product available as identical objects. Of course, each individual who experiments with circuit bending may have his or her particular reasons for doing so. The other avenue away from the imposed restrictions of readymade instruments is programming in audio synthesis languages. However, its inherently abstract and mathematical approach cannot appeal to everyone, so it is quite understandable that some musicians prefer the hands-on concreteness of tinkering with electronic circuits.

Regarding autonomous instruments, they might be realised either as computer programmes or as special purpose hacked circuits, although if one insists on pure autonomy, then it appears to be more difficult to isolate an analogue circuit from its surroundings. And as far as machine listening techniques are concerned, there are some interesting fore-

runners who worked with analogue circuits and made pitch and envelope followers. The work of David Behrman and George Lewis in the 1970s are notable examples.

In principle, there is no reason why feature extractors and a modular unit generator design should not appear in commercial synthesisers. This is the basic requirement for doing adaptive synthesis and setting up feature-feedback systems in particular. On the other hand, there is probably no big need for it, as long as it is relatively simple to set up one's favourite synthesis language for this. Csound has witnessed a huge increase in code size, with several unit generators being added for each new release. ChucK is promising since it is directly aimed at live coding, and has processing units for the extraction of spectral flux along with a few other features. Other popular choices include MAX/MSP, Pd and SuperCollider, all of which offer functionality for machine listening.

The programming language used for experimentation with synthesis models in this thesis is C++, a choice made mostly for reasons of familiarity. The programmes have command line interfaces and operate in offline mode for reasons of simplicity of implementation. Nonetheless, it should be stressed that there is nothing in the feature-feedback systems that make them unsuitable for being implemented in a more fashionable and up-to-date version with graphical user interface and all types of knobs, bells, sliders and whistles. That is, provided the instruments one builds are not far too computationally demanding for the computer they run on, they should be able to run in realtime. The instruments that will be introduced in Chapters 6 and 7 seem to be well within the limits of typical present day computer power.

There are not only disadvantages to working offline, as one might think. This approach enables us to closely scrutinise the dynamics of closed systems in ways that would be impossible if they were interactive. Perhaps some of our findings generalise to the interactive setting, but most do not. If the autonomous synthesis models are the least successful, they should have some interest for use as sound generators in computer music.

1.4.5 Algorithmic composition

We will not dwell long on the historical roots of algorithmic composition, which many authors rightly trace back to long before the computer era, indeed as far back as to the introduction of musical notation (e.g. Roads, 1996; Essl, 2007; Nierhaus, 2010). If computers are not necessary for algorithmic composition, then at least some codified system of symbols, a musical notation, seems to be needed. A prevailing conception of algorithmic composition is that it deals with notated music, but there are other domains where algorithms can structure music.

For various reasons, musical composition lends itself well to systematised approaches. What this means is that the composer does not just conjure up sequences and juxtapositions of notes or sounds as seems fit, but rather begins with a system or a procedure, which can be followed more or less automatically once it is set up. A typical example would be the canon, which takes one voice as its input and generates all the rest by imitation (this is of course a crude over-simplification, since it ignores additional constraints such as harmony and voice leading). Integral serialism is another example. A necessary prerequisite in these cases is that the process can be defined in terms of symbol manipulation. The notes with their pitch values, durations, and dynamics can all be expressed

as numbers, and are thus amenable to mathematical operations. All this organisation ignores the question of audibility of the organising principles. Indeed, this was a critique raised by Xenakis against integral serialism in "The crisis of Serial Music" originally published in Gravesaner Blätter in 1955 (Xenakis, 1992, p. 8). What Xenakis had found was that integral serialism was only able to generate perpetual maximal variety, which somewhat paradoxically leads to perceptual stasis, whence he suggested statistical treatments of masses of notes.

Serialism, based on the assumption that each possible value must occur only once before being repeated, is just one amongst an endless number of ways to organise material. There are two important extensions: first, what material is used, or what are the underlying units that are manipulated (e.g. discrete pitches or more complex objects) and second, what principles of organisation are used.

Algorithms can be defined as a process that acts on a system or a set of elements. Given an initial state of the system and a set of rules for how to update the state, a succession of states follows by iteratively applying the rule, possibly to a predefined final state. As Herbert Brün has noted, there is often a close link between algorithms and more or less gradual transformations:

> If we now call an algorithmically controlled change of state a "transformation", then we can say that an algorithm produces an uninterrupted chain of transformations between a given initial and a given final state of a system. Or, the other way around: if two states of a system appear to be connected by an uninterrupted chain of transformations, then we may assume the presence of a controlling algorithm (Brün, 2004, p. 194).

If the output of an algorithm is used as musical material in the same order of appearance as it is generated by the algorithm, then a knowledgeable listener may well realise that this is the result of applying a process. However, there is no guarantee that an algorithm will generate a predictable and smoothly varying output. Stochastic algorithms may generate sequences of elements where each element has a certain probability of occurring, but where the occurrence of one element does not have any bearing upon the next outcome.

Although today, algorithmic composition is most likely carried out by computer, the same sequence of operations could be carried out manually, insofar as the workload does not exceed the composer's patience. As already mentioned, algorithmic composition is often assumed, tacitly or explicitly, to deal with the note level of music and nothing else. This would imply that we cannot find algorithmic composition in musical cultures that have not developed notation. So much may be commonsensical, but there are interesting findings in ethnomusicology that indicate uses of quite complicated patterns in aurally transmitted music. Of particular interest are the harp patterns used in Nzakara harp music in Central Africa (Chemillier, 2002). Such elaborate constructions are perhaps at the limit of what can be done without recourse to a notational representation, and are, so to speak, borderline cases of algorithmically constructed music. If it is hard to decide whether the Nzakara harp patterns really qualify as algorithmically constructed music, at least the musicologists who analyse it have come up with algorithms that regenerate the same patterns.

	Style imitation	**Genuine composition**
Symbolic (note level)	Cope's EMI, Hiller's Illiac suite I–II	Xenakis' and Koenig's instrumental works
Subsymbolic (signal level)	Synthesis of musical expressivity (Sax phrases)	Xenakis' S-709, Brün's SAWDUST, autonomous instruments

Table 1.2: Approaches to algorithmic composition.

But any music might be analysed for a candidate generative algorithm that recreates it—surely that does not imply that the music is conceived through algorithmic composition! That is true, but some music may be easier than others to find generative algorithms for. The work of David Cope on style imitation indicates that at least Western music from the common practice repertoire and more recent styles may be successfully mimicked if the algorithm he has developed is fed with the right input (e.g. Cope, 1992). Cope has produced music with his EMI (Experiments in Musical Intelligence) programme that achieve a high similarity with the styles of the old masters.

In algorithmic composition, the output of the algorithm can be applied on various musical levels. Much use has been made of algorithms to generate material on the note level, though there are also some examples where algorithms are used directly on the sound signal level, and sometimes in such a way that it becomes hard to distinguish what may otherwise be called a note level and a timbral level. A brilliant case in point is—to return to our often cited example—Xenakis' Gendyn pieces such as *S.709*.

Now it is legitimate to ask whether there is really any need for a term such as "autonomous instruments"—apart from being shorter—if it is already covered by a phrase like "algorithmic composition on the signal level". We would like to reserve the autonomous instrument category to works that as much as possible let the algorithm speak for itself. This is an ideal aesthetic position that can only be approached partly in practice. Algorithms have to be invented, and even more importantly, the musical representation used for mapping abstract data to sound plays a crucial role.

Table 1.2 is one possible classification of approaches to algorithmic composition. On the one hand, there is a division between *style imitation* and *genuine composition*. These slightly value-laden terms come from Nierhaus (2010). On the other hand, the table is divided into note level or symbolic composition, and subsymbolic, signal level composition. There is a tendency to overlook the subsymbolic approach to algorithmic composition; Nierhaus, for instance, barely mentions this possibility. A plausible reason to why it is rarely mentioned as such, is that it gets subsumed under other categories such as generative music.

Copes Experiments in Musical Intelligence, as already mentioned, is a canonical example of style imitation. The *Illiac Suite* from 1957 by Lejaren Hiller and Leonard Isaacson (mentioned in the beginning of this chapter) often counts as the first time computers were used for algorithmic composition. Hiller had a pragmatic attitude and his compositions are often eclectic; the two last movements of the *Illiac Suite* involve contemporary musical idioms whereas the two first movements deal with style imitation (Hiller, 1981). Apart from Hiller, Xenakis, G. M. Koenig, Herbert Brün and many others were investigating the

possibilities offered by computers in genuine algorithmic composition, mostly in notated scores, but arguably also directly in sound synthesis. It is quite obvious that autonomous instruments belong to the category of subsymbolic genuine composition. However, the fact that it is implemented on a subsymbolic level does not imply that it will or should stay on that level; on the contrary it is most interesting when higher levels emerge from the low-level specifications.

Examples of subsymbolic style imitation are hard to find. If imitation is restricted to short segments such as single notes, then this is just imitative sound synthesis, and both physical and spectral modelling would be perfect examples. On the other hand, imitation of longer segments are easiest to tackle on the note level, but a possible case of subsymbolic style imitation might be the work on expressive synthesis of instrumental phrases. In that case, a notated phrase is synthesised, and suitable expressive nuance such as pitch inflexions, vibrato, and other articulations are generated algorithmically. One such example is the expressive synthesis of saxophone phrases (Arcos et al., 1997; Kersten et al., 2008).

In a few rare cases of subsymbolic genuine algorithmic composition, such as Xenakis' piece *S.709*, there exists a complete source code for the generation of the piece. Thus it would be possible, at least in principle, to recreate such pieces from their code. In the case of *S.709* this is virtually impossible, because the programme depends heavily on stochastic number generation. But then, if one would rerun the programme that generated the piece, one would not get a cloned version, but rather a style imitation that might come quite close to the original version.

Perhaps it is incorrect to draw a sharp division between style imitation and genuine composition. As the composer invents rules for the algorithm to carry out, some kind of musical ideal must surely linger in his or her mind, an ideal that can hardly escape being influenced by all the music the composer is familiar with. And if the intent is to mimic an existing style, but the result fails to resemble the original, is that not originality, even if unintended? Originality is a difficult concept offering plenty of opportunities for misunderstandings. What is often meant, is not merely that an original work of art differs in a few details from another work, but rather that there is some stylistic novelty. With algorithms that produce complex and unexpected output, questions of style may seem moot. But this is probably a misapprehension, because as the composer acquires knowledge and experience from working with that unwieldy algorithm, some understanding of the causal implications of various ways to structure the programme or setting its parameters will emerge; this understanding will undoubtedly influence the choices that the composer makes.

Finally, it should be added that strict algorithmic composition in the sense that the composer invents an algorithm, runs it, and translates its output to sounding music without any further interventions is rather uncommon. The other extreme is free or intuitive composition, where formalisations are avoided and inspiration supposedly flows freely from the composer's musical imagination to the score. A fertile middle ground is that of computer assisted composition (Anders and Miranda, 2009), where some tasks are delegated to the computer, but the composer retains much of the detailed control. We will return to these ideas in the final chapter.

1.5 Summary and outline

Autonomous instruments can generate sound files of arbitrary duration, limited only by available computer memory. Only a reduced form of interaction takes place, which is a cycle of writing the code, setting parameter values, running the programme, listening to the output, perhaps modifying the programme code, and trying out new parameter values. If an interesting sound should emerge among all test runs, it could be saved and used as material for a fixed composition. This is the mode of operation that we will mostly have in mind when exploring these techniques.

In this chapter, we have discussed some obscure corners of the electroacoustic music scene where performers and composers invent more or less autonomous systems that produce sound. In many cases, these works can be related to a tradition of experimental music. A recurrent aesthetical standpoint is to let the process speak for itself rather than engage in self-expressive music making. This attitude can be related to an aesthetics of nature, if one understands the autonomous system and its sonic output as a natural process. Yet the result may sound extremely artificial and impossible to achieve without electronics or computers.

The feature-feedback systems that will be studied in later chapters apply a feature extractor to the output of the system which is used for controlling its synthesis parameters. A number of useful feature extractors will be introduced in the next chapter. In particular, we emphasise the usefulness of time-domain feature extractors for our purposes. Chapter 3 continues by discussing how feature extractors relate to synthesis parameters, and gives a review of abstract synthesis models that will be used in feature-feedback systems. Feature extractors are crucial ingredients in adaptive audio effects, feature-based synthesis and concatenative synthesis, as well as in all kinds of interactive music making that employs machine listening. By sealing off any direct control and setting it up in a closed loop, autonomous feature-feedback systems are very different from any other of the cited applications of feature extraction. In Chapter 4, the relation to dynamic systems will be developed, but we also take a close look at some uses of chaotic systems in musical composition and sound synthesis. Feature-feedback systems are complicated and potentially chaotic dynamic systems, which is why the theory of dynamic systems is necessary for understanding their behaviour.

Feedback is an important concept, thus Chapter 5 begins with a short survey of feedback systems of various kinds that have been used in music. Many feedback systems arguably show emergent behaviour and self-organisation. The same chapter also brings up various notions of "complexity", both in relation to contemporary music and complex adaptive systems. We will argue that some kind of complexity could be a good criterion for the evaluation of autonomous instruments.

Chapter 6 deepens the understanding of feature-feedback systems by solidly anchoring them in chaos theory. Several methods of investigating parameter spaces are introduced which are useful for understanding some of the dynamics of feature-feedback systems and not to just be amazed by it. Such methods prove useful as tools for the deliberate design of autonomous instruments, a topic that is dealt with in Chapter 7. There we focus on how to generate non-stationarity, that is, how to avoid static or cyclically repeating behaviour, which can sometimes be quite hard to do. Then, some more remote

applications of feature-feedback systems are briefly discussed; we show how to generalise the ideas to note-level composition and concatenative synthesis. The results from an internet questionnaire about preferences and complexity evaluations of sounds made by autonomous instruments are reviewed in Chapter 7, and further discussed in the last chapter. Finally, the last chapter also raises questions about how to generate or compose music with autonomous instruments; why improvisation is virtually necessary in performance with semi-autonomous instruments, and some problems that face musicologists trying to analyse such music are discussed.

Frequently used notations and abbreviations are explained in Appendix A. Several sound examples have been included to illustrate the synthesis models. The sound examples appear as follows:

Example 1.1. The electronic version contains direct links to the sound examples on the internet. Readers of the printed version may wish to bookmark the web page[3], where the sound examples appear ordered by chapter and with the same numbering as in the heading of the examples.

In view of the discussion in the beginning of this chapter, this thesis does not try to pursue research *in* the arts; hence, the accompanying sound examples should be taken for what they are: illustrations, rather than compositions in their own right.

[3]http://folk.uio.no/ristoh/adapt/sndex.html

Chapter 2

Feature Extraction and Auditory Perception

Audio feature extraction has its applications in music information retrieval, in computer music and in studies of voice quality and timbre perception. By definition, feature extractors are one of the components of feature-feedback systems where the generated output signal is monitored so that the signal generator's parameters may be updated automatically in response to the recent output. Therefore, feature extractors will play a prominent role in most of the following chapters. The point of using feature extractors in general is to provide compact descriptions of various perceptually salient traits of the signal. This chapter reviews several feature extractors and addresses questions about the relationship between auditory perception and signal descriptors.

Pierre Schaeffer introduced many useful concepts for describing sound and for talking about something as ephemeral as the attention in the act of listening. Although there are many studies in the field of music psychology of timbre perception, we will see that the most restricted notion of timbre, pertaining only to the steady-state spectrum of a sound, is by itself too restricted to allow for a fair treatment of the whole gamut of possible sounds, whereas the typomorphology proposed by Schaeffer can be a useful complementary aid for sound description.

However useful it may be to be able to attach semantic labels to sounds in a fairly systematic manner, it is the application of automatic feature extractors that will be of greater concern to us. Yet, no feature extractor will be very useful unless it differentiates between perceptually distinct qualities in a consistent way.

The bulk of this chapter is a review of various feature extractors and ways to implement them. Even the simplest low-level feature extractors can be very powerful tools when used in parallel. They make concatenative and feature-based synthesis possible, and they are crucial parts for machine listening; furthermore, feature extraction can be used in the analysis of music that lacks a notated representation and as tools for exploring parameter spaces of synthesis models. Before dealing with the relationship between synthesis parameters and feature extractors in the next chapter, we first need to relate audio features to perceptual qualities.

2.1 Dimensions of sound

Studies of timbre have often focused on pitched sounds, but what are we to make of the differences in sonic qualities between unpitched sounds? Anyone with normal hearing has no trouble telling apart sibilants such as /sh/ and /s/, neither of which has any pitch. A vast range of low-level feature extractors are available for analysing recorded or synthesised audio signals, but it is not always clear what perceptual correlates the analysed features have. On the other hand, Schaeffer's typomorphology is a comprehensive system for the classification and description of sound, although the acoustic correlates of its categories largely remain unclear. We are faced with the problem of relating objective descriptors to subjective experience. To keep things clear, the terms "feature" and "signal descriptor" are reserved for objective measurements of signals, whereas "sonic character" or "trait" refers to the subjectively perceived qualities.

2.1.1 Measurements and percepts

Instruments for measurement may be said to extend our senses, and may allow us to gather information more finely tuned than we are capable of directly perceiving, or beyond the confines of our sensing organs (ultrasonic vibration, radio frequency electromagnetic radiation), or otherwise inaccessible to direct observation. Still, some instruments of measurement do not precisely extend our senses, but rather they allow us to study the relationship between physical stimuli and perceived quality. In his essay *The Function of Measurement in Modern Physical Science*, Kuhn made the following observations:

> Many of the early experiments involving thermometers read like investigations *of* that new instrument rather than investigations *with* it. How could anything else have been the case during a period when it was totally unclear what the thermometer measured? Its readings obviously depended upon the "degree of heat" but apparently in immensely complex ways. "Degree of heat" had for a long time been defined by the senses, and the senses responded quite differently to bodies which produced the same thermometric readings. Before the thermometer could become unequivocally a laboratory instrument rather than an experimental subject, thermometric reading had to be seen as the direct measure of "degree of heat," and sensation had simultaneously to be viewed as a complex and equivocal phenomenon dependent upon a number of different parameters (Kuhn, 1977, p. 208, italics in original).

When we develop novel measures for sensory qualities, it will be useful to remember this dual direction of investigation. Here, the investigation of a novel measuring instrument and of its measure's correlation to perceived qualities is of primary interest. When the instrument is better understood, it may be employed to read off values carrying some meaning from the data, such as in checking the thermometer for appropriate clothing before leaving home.

Audio feature extraction deals with representations of audio signals, and does not exactly extend the range of our senses. Instead, it may be regarded as an information reduction from the richness of complex audio signals to simplified descriptors. Feature

extraction usually involves the smoothing of input data by taking averages over moving temporal windows, hence it implies a data reduction by decreased bandwidth or sample rate. If a sufficiently high number of feature extractors describing independent audio attributes are combined, this ensemble of descriptors could qualify as a representation of the signal. This representation would probably not allow perfect reconstruction, i.e. it would be a lossy representation. So the purpose of feature extraction is not a complete description of the signal to the level where it can be reconstructed, but a data reduction that keeps a few descriptors corresponding to perceptually salient traits.

Most feature extractors to be discussed in what follows analyse an audio signal and return a vector of numbers, corresponding to how that signal attribute changes over time. Now, these tools can serve the purpose of identification or classification of signals in terms of some of their perceptual qualities. This is what a pitch tracker, an envelope follower or a spectral centroid analyser achieve. They may be more or less apt instruments for measuring their respective perceptual correlates, though measurement instruments may as well be developed without any precise knowledge of what exactly is its perceptual correlate. For example, some descriptive statistical measure may be applied to the phase spectrum of a sound segment, but what perceptual quality, if any, does this measure capture? Finding the perceptual correlate of a measure is not necessarily trivial. Even in cases where reliable measures have been developed to meet precise needs such as the classification of voices into healthy and pathological, their interpretation may become highly problematic as soon as they are employed outside their original domain (cf. Section 2.3.8).

2.1.2 Schaeffer's theories on listening

Pierre Schaeffer's single most important theoretical contribution, his *Traité des objets musicaux*, is often cited for its innovative classification of sounds into a typology. Yet, his theorising of the act of listening (as exposed in Book II of Traité) is even more significant. The detachment of sound from sound-producer, known as the acousmatic listening condition, became a ubiquitous experience with the introduction of radio broadcasts and recorded sound on phonographs. Then, years of extensive studio experience led to the idea of *l'objet sonore*, the sound object.

Schaeffer is very explicit about what the sound object is *not* (Schaeffer, 1966, pp. 95-97): It is not the instrument or the sound source which made the sound. Neither is it to be found on a few centimetres of magnetic tape that carries the recording of a sound, because it needs to be heard by a listener in order to become a sound object. Although originating in the physical world, it is entirely contained in the listener's consciousness. Moreover, the same piece of magnetic tape may contain several different sound objects, say, if the tape is played at different speeds. However, and this is a rather subtle point, the listener's will to compare those different renderings of the same recorded sound at different transpositions may suffice to make of it a single object, observed from different perspectives, as it were. For another listener, those different versions of the same recording may as well constitute unique sound objects. Finally, the sound object is not a mental state of the listener, despite being qualified as situated in the listener's consciousness. It remains the same object regardless of who listens to it and how one's attention fluctuates. Hence, the

sound object should not be seen as something merely subjective, in the sense of being the unique and private experience of each individual; rather it is seen as an object which can be analysed and on whose properties one can reach intersubjective agreement.

Reduced listening (*l'écoute réduite*) is another central notion in Schaeffer's conceptual apparatus, which we will return to in Chapter 8. In our daily life, we need to identify sound sources, and often interpret the meaning of sounds. The attitude of reduced listening is to turn this natural curiosity concerning the causes and meanings of sounds towards the inner properties of the sound itself. In particular, the sound object is constituted in an act of reduced listening.

Experimental music (or musical experimentation) was the term that Schaeffer ended up with as a characterisation of his musical research. Whereas the experimentation that was being carried out in psychoacoustics made use of simple stimuli such as sinusoids and white noise, Schaeffer's experimental approach began with the kinds of stimuli that one would encounter in actual music (Schaeffer, 1966, pp. 168-169), particularly in the electroacoustic music of the day. In other words, Schaeffer's approach is to begin with ecologically valid stimuli, in this case short sounds that might be used in tape music, and to experiment with the perception of these sounds.

More recently, Plomp has argued in favour of a similar approach to research on sound perception and cognition, mostly in the context of speech perception (Plomp, 2002). Among the prevailing habits among researchers in hearing and psychoacoustics, Plomp lists the following: Simple stimuli, such as sinusoids, are studied instead of more complex sounds that meet us in our daily sonic environment. A "microscopic" approach is favoured over a "macroscopic" one, by which Plomp means making controllable laboratory experiments where a single parameter may be isolated and varied, and its effect studied—but experiments with pure tones will reveal nothing about the perception and cognition of spoken language. Further, there is a predilection for psychophysical aspects of perception: studying sensation instead of interpretation. Finally, the "dirty conditions" of everyday life are abstracted away as much as possible. Laboratory tests are typically conducted in noise-free settings, where disturbances are eliminated; this, however, stands in stark contrast to the everyday conditions in which listening operates.

These shortcomings notwithstanding, many useful studies of timbre perception have been carried out in the past years, some of which will be reviewed below. The criticism raised by Plomp does not really affect the value of these studies, which are useful inasmuch they make the relationships between acoustic parameters and perceived qualities expressly known. But even these timbre studies are quite limited in scope if we think about the vast canvas of possible sounds that might some day be studied. To this end, Schaeffer's typomorphology may serve as an inspiration.

Whereas the immediate aim of psychoacoustic research has been to study the relationship of stimuli with known and precisely controllable acoustical properties to their perception, the kind of project outlined by Schaeffer goes in the opposite direction. Starting with complex stimuli, one tries to find verbal labels, which may be refined over repeated exposure to the sound object. A further extension, still awaiting serious studies, would be to seek the acoustic correlates of the Schaefferian typological and morphological categories. Such an endeavour would be extremely complicated, for at least two reasons. First, there has to be semantic labels that are accepted by a broad enough community,

and there has to be a certain consensus about what these labels represent sound-wise. In the case of the typomorphology, it means that listeners have to be familiar with that particular system of classifying sounds. Second, and perhaps more seriously, there is an inherent vagueness and intended flexibility in Schaeffer's terminology. He did not provide clear, unambiguous acoustical definitions of the terms in the typology and morphology, which instead were to accommodate for varying listening intentions. "Nous pensons en effet que le principe de notre classification permet d'assigner au même objet diverses cases selon l'intention d'écoute. La recherche d'une typologie ' absolue ' est illusoire" (Schaeffer, 1966, 433). Looking for an absolute or objective typology is wasted effort, since the classification allows one to classify the same sound object in various ways depending on the listening intention. If this is taken seriously (and there is no reason not to do so), the idea of trying to relate Schaefferian terminology to acoustic attributes in a rigid way might as well be dropped (see also Holopainen, 2009). However, the flexibility of listening intention that Schaeffer primarily seems to have in mind is to consider longer or shorter segments as a sound object; then, depending on its length, a prolonged sound object may fall into a different category than the sub-objects that it may contain.

Nevertheless, the typomorphology does provide some useful terminology for describing sound in a metaphorical way. Similar metaphorical descriptions of sound can be found in some user interfaces to synthesisers or effects, where the knobs may carry labels such as "brightness", for example. This can be very useful, since the sound's brightness may be influenced in several different ways that need not concern the end user.

As we learn more about signal representations, such as the spectrogram, and their correlations with perceived characteristics of sound, the more we may borrow precise terminology for the description of sounds from these representations instead of resorting to more impressionistic circumlocutions. Musicians who are familiar with using spectrograms may be able to tell the sound engineer what frequency to turn down on the EQ. As familiarity with low-level feature extractors grows, perhaps it will become more common to describe sounds in terms of their centroid, zero crossing rate, spectral slope, and so forth.

2.1.3 A general music theory of the sound-world

In Schaeffer's theory, classification into categories is the first step. The typology classifies sounds according to three pairs of criteria. First, there is *mass*, which is related to how a sound occupies the spectrum (such as its pitch, if any) and *facture*, related to the shape of the sound over time. The mass may be fixed, with or without an identifiable pitch; or it may be variable as in glissandi and diphthongs; further, this variation may be simple and organised, or disorganised and unpredictable. The facture can be either prolonged and continuous, impulse-like, or iterated as in a series of repeated impulses. The second pair of criteria are the *duration* and *variation*. These are tightly linked, because perceived duration is related to the amount of variation. *Balance* and *originality* are the third pair of criteria, which are used to distinguish sound objects that are redundant, balanced, and "excentric" or too complex. For a readable short introduction to the typology, see Thoresen (2007).

The typology is further extended with several morphological criteria that allows for

finer descriptions of sounds that have been classified as belonging to a certain type. Seven criteria are distinguished in the morphology, which are partly overlapping with the criteria already used for the typology (the conceptual complexity of Schaeffer's theories must not be underestimated). The morphological criteria are *mass*, *dynamic*, *harmonic timbre*, *melodic profile*, *mass profile*, *grain*, and *allure*. Chion describes the utility of these criteria (briefly summarised below in Section 2.1.4) as follows:

> La notion de critère morphologique, plus générale que celle de valeur, s'impose lorsqu'on veut faire un Solfège général du monde sonore, et qu'on doit renoncer à utiliser la notion de timbre et les valeurs musicales traditionnelles, qui ne se justifient que pour le cas particulier des musiques classiques occidentales. La notion de timbre renvoie en effet à l'*identification instrumentale*, comme perception synthétique d'un certain nombre de caractères associés dans le son, plutôt qu'elle n'aide à décrire et à percevoir en eux-mêmes ces caractères. Or, avec la musique de studio, il n'y a plus d'instrument (Chion, 1983, p. 143).
>
> [The concept of morphological criterion, which is more general than value, is essential if we want to build a general Music Theory of the sound-world and must give up using the concept of timbre and traditional musical values, which are only relevant to the particular field of Western classical musics. Indeed, the concept of timbre is bound up with instrumental identification as a synthetic perception of a certain number of associated sound characteristics, rather than an aid to describing and perceiving these characteristics in themselves. Now, with studio music, there is no longer an instrument.][1]

This is exactly the weakness of the usual notion of timbre (see also Section 2.1.5). The "valeurs musicales" referred to are such as pitch classes, which are amenable to being organised into scales.

The very idea of a *solfège* of sound objects (Schaeffer and Reibel, 1998) borrows its name from the interval naming that is taught at elementary level ear training and music theory classes. Since there is usually no doubt about the objectivity of the intervals that musicians are asked to name, one may wonder why not the classification into typomorphological categories should be as straightforward. However, the aim of the typomorphology is different; its purpose is not to identify abstract values such as pitch classes that can be ordered into scales, but to classify and describe sound in its concrete diversity. As with traditional solfège, this practice would require consensus and training, but the consensus is somewhat undermined by allowing the leeway of different listening intentions to influence the categorisation.

Sensory evaluation is important in quite different domains, such as product testing of foods, perfumes etc. (Meilgaard et al., 2007). Test procedures have been developed to deal with the uncertainties caused by individual variations among test subjects, which may be an even greater problem in the olfactory and gustatory senses than in hearing. But here too, the education of test panellists is stressed, so that they will function as reliably as possible as a test instrument.

[1] English translation by John Dack, whose renderings of Schaeffer's terminology we have tried to follow here. The complete translation of Chion's *Guide des Objets Sonores* is available at: http://www.ears.dmu.ac.uk/spip.php?page=articleEars&id_article=3597

Perhaps the problem of reaching consensus in descriptions of sound objects is primarily one of translation from the auditory modality to verbal labels or other types of indications. In studies of voice quality ratings, Gerratt and Kreiman (2001) meticulously synthesised one second voice fragments to match recordings. The synthesis model allowed control of the relative balance of harmonic and noise components. Listeners who had experience in evaluating voice disorders judged the perceived noisiness in twelve voice recordings and indicated their evaluation on a visual-analogue scale. In another experiment, the listeners had to adjust a synthesis parameter controlling the amount of noise so that it matched the original voice recording. The visual scale ratings revealed high variance of responses, while the synthesis task had low variance in 9 of 12 voice stimuli. Closer inspection of the three cases with disagreeing judgements revealed that despite the great variance, listeners had responded within a difference limen.

This study of Gerratt and Kreiman points to the difficulty of reaching agreement when the task involves mapping the perceived sound to other modalities, be they the position of a point along a line segment or a verbal label. Nonetheless, as long as the task is to adjust a single synthesis parameter so as to produce the sound which is as close as possible to the target, there is good intersubjective agreement. This of course is a cumbersome method that cannot easily be generalised to larger sets of sounds, which would require more complicated synthesis models and a large number of adjustable parameters.

2.1.4 The typomorphology

Here we briefly summarise the criteria used in Schaeffer's typomorphology. Any synthesis model that is capable of producing a rich and varied set of sounds will vary along several of these dimensions.

Grain As an inter-modal concept, grain applies to hearing as well as to vision and touch. Often there is a correlation, so that what looks and feels granular, such as a grated surface, will also produce a granular sound when dragging a stick over the surface. There may be many disparate sources of a granular sound, which is quite in accordance with the reduced listening intention that puts the sound's physical origin into brackets and facilitates a focus on morphological similarities between sounds of different origin.

Although there is no definition in acoustical terms of what grain is, there are a few evocative examples in the *Solfège de l'Objet Sonore* (Schaeffer and Reibel, 1998, first published in 1967). Low bassoon notes have a coarse grain, which becomes finer with increasing pitch; the sound produced by scraping over a rough surface will be granular and have aural qualities that are directly analogous to the surface structure; the sound of a cymbal has a fine scintillating grain. In acoustical terms, the common trait of granular sounds seems to be rapid modulation, and amplitude modulation in particular.

Allure Vibrato serves as the model for allure; it is a generalised slow modulation of pitch, loudness, or timbre. Allure corresponds quite well with the concept of Low Frequency Oscillators (LFOs) in synthesisers, which control the periodic modulation of assignable synthesis parameters, typically frequency, amplitude, filter cutoff frequency or the pulse width of rectangular waveshapes. Schaeffer further distinguishes three classes

of allure, according to their manner of variation. There is the mechanically regular, the more relaxed periodicity of a human agent and the unpredictable irregularity of natural processes. As with grain, the speed and amplitude of allure can also be described.

Grain and allure both relate to the sustainment (*entretien*) of the sound. Grain pertains to a finer level, while allure deals with slower change. Both however are distinguished from the general shape or envelope of the sound object.

Harmonic timbre and mass The criterion of harmonic timbre deals with specifying the colour of pitched sounds; essentially then, it contains those timbral dimensions that correspond to spectral shape and balance of partials. In Schaeffer's system, the criterion of mass complements that of harmonic timbre. It applies not only to pitched sounds, but describes how the sound occupies different pitch regions or tessituras. Related to mass, there is a scale of sounds ranging from a pure sinusoid to white noise, with a typology of seven positions in this continuum. A synthesis model which has a noise-to-sinusoid parameter will be detailed in the next chapter.

Dynamics and other profiles The criterion of dynamics describes the sound's temporal intensity profile. As such, it is related to the form or shape of the sound, and particularly important in this respect is the attack. The attack typology refers to various modes of excitation such as plucking with a plectrum, pizzicato, striking with a felt-clad hammer or fading in gradually.

Melodic profile applies most obviously to pitched sounds, but also to any temporally variable sound where the entire mass moves in the same way, so as to trace out a trajectory in pitch register. The neumes from Gregorian chant are borrowed to symbolise some of these profiles.

Mass profile is often linked to both melodic and dynamic profile, but an example where they are independent occurs with the use of a wah-wah effect or similar dynamically variable filtering.

Although it is safe to say that Schaeffer's typomorphology has not very often been related to feature extractors, there are some scarce examples. Peeters and Deruty (2008) developed a search-engine that uses descriptors inspired by the typomorphology. Morphological sound description mainly using pitch and loudness profiles was proposed by Bloit et al. (2009). These examples show that it is possible to begin with qualitative or metaphorical sound descriptors and work backwards to feature extractors. For this to be possible, any qualitative terminology first has to be translated into unequivocal acoustic and signal processing terms that can be implemented as feature extractors.

2.1.5 Timbre studies

Definitions of timbre have proven to be difficult to agree upon. For instance, McAdams (1999) defines it as that perceptual attribute which distinguishes two sounds otherwise similar in pitch, loudness, duration and spatial position[2], but this only says what timbre is not. To say that timbre is simply a multidimensional phenomenon is not very informative

[2]McAdams is clearly alluding to the often quoted 1960/1973 ANSI definition of timbre, only adding duration and spatial position to the list. For some examples of the various timbre definitions that have

either. Instead of proposing an alternative definition, it is revealing to look at some of the psychological experiments on timbre perception that have been carried out in the past years. In this research, various timbral dimensions have been isolated and sometimes coupled with acoustic features or semantic labels. Test subjects have rated perceived distance or similarity between pairs of sounds, and by multidimensional scaling (MDS) some low dimensional timbre space has been fitted to this data. In many cases, a three-dimensional timbre space has been found. Acoustic correlates of the resulting dimensions have often involved the attack time (or log-attack time), the spectral centroid, and spectral flux (McAdams et al., 1995).

In one of the early timbre studies utilising MDS, Grey (1977) used single Eb ($\approx$ 311 Hz) tones from string instruments, woodwind, and brass. This study featured a normal, a muted sul tasto, and a sul ponticello tone from a cello, and two nuances (p and mf) from a saxophone. Indeed, when normalised for loudness, different nuances on the same instrument will sound different; the one originally played with louder articulation will typically be brighter. Grey found a MDS solution with three dimensions, the first related to the spectral energy distribution, and the second related to low-amplitude energy in the high frequency range in the attack segment, and the third related to synchronicity in transients in higher harmonics together with spectral fluctuation over time. It is also noteworthy that clustering of the instruments revealed that perceived similarity was not necessarily based on instrument family membership; thus the flute is found in the same cluster as the strings, and the bassoon and trumpet are also very close.

An investigation of the relative importance of the attack portion of a tone versus the remainder by Iverson and Krumhansl (1993) used a collection of woodwind and brass instruments, violin, cello, piano, vibraphone and tubular bells. Similarity judgements of complete tones led to a two-dimensional space with static spectral attributes on one axis and dynamic attributes on the other axis. The tubular bells occupy a position quite far removed from the other instruments, and one may speculate whether its inharmonic spectrum contributes to this. Comparisons were made of the MDS solutions for attack segments only (first 80 ms) with the complete tones, and the remainder where the attack was removed. All three conditions led to similar spatial arrangements, leading to the conclusion that tones that had similar attacks also tended to have similar amplitude envelopes in the rest of the tone.

Krumhansl (1989) studied instrumental tones and hybrid instruments created with FM synthesis, and found a three dimensional timbre space with dimensions of brilliance, rapidity of attack and spectral flux. Others have later repeated the study, and found that the third dimension is actually better described as spectral irregularity or fine structure of the spectral envelope (Krimphoff et al., 1994). Spectral irregularity turns out to be a distinguishing feature in the recognition of car horn sounds, together with the fundamental frequency (normally around 400 Hz). Other timbral dimensions that have been found to distinguish car horn sounds are the spectral centroid and degree of roughness (Lemaitre et al., 2003).

In studies of simultaneous tones from pairs of wind instruments (oboe, trumpet, clarinet, alto saxophone, and flute), Kendall and Carterette (1993) initially found three

been proposed from Helmholz and on, see the compilation of G. Sandell:
http://acousticslab.org/psychoacoustics/PMFiles/Timbre.htm

dimensions by MDS: degree of nasality, brilliance vs. richness, and strong and complex vs. weak and simple. The third dimension was dropped, since the two first dimensions gave about as good a fit to the data as three dimensions. Oboe and flute in combination produced the most nasal timbre, whereas saxophone plus clarinet were the least nasal. On the second dimension, dyads including trumpet were more brilliant, while those including saxophone were more rich. For unisons, the degree to which dyads blend is inversely correlated to the identifiability of the instruments. If the two tones differ in the use of vibrato, it will be easier to pick out the individual instruments, as is known from the principle of stream segregation in auditory scene analysis (Bregman, 1990).

Vowels depend on their formant frequencies, relative strengths, and bandwidths. It is customary to draw a two-dimensional vowel space with the first two formants as co-ordinate axes, although a third formant is also important for the distinction of certain vowels (Ladefoged, 2005).

Nevertheless, there are apparent lacunas in the mapping of timbre space. After this short review of a few timbre studies, a picture emerges of research that has focused on pitched, harmonic sounds from acoustic instruments (plus a few studies on other sources, for example vocal sounds). What has been much less prioritised are crush tones on string instruments, multiphonics, most of the percussion family, non-western instruments, and countless other sound sources and sound types that find use in parts of contemporary music. However, a study by Lakatos (2000) combined harmonic tones and percussive sounds. A two-dimensional space was found as a solution to the combined set of sounds, with dimensions logarithm of attack time and spectral centroid. For the percussive sounds alone a third dimension was found, although its acoustic correlate remains obscure. As Lakatos admits, two dimensions are not at all sufficient if one wants to synthesise realistic timbres; hence it appears likely that other dimensions must play some role. Those other dimensions might or might not be continuous; there is a possibility that categorical perception occurs in distinguishing different kinds of excitation such as blowing, bowing, striking (Donnadieu, 2007) and different resonant materials and shapes such as strings, membranes, wood, or metal plates.

2.1.6 Discussion

The almost universal finding that spectral centroid and attack duration correspond to the two most important timbral dimensions may very well reflect shortcomings of the MDS method or biases in the collections of sounds used in experiments. It is conceivable to make an experiment where not only pitch, loudness, and duration are normalised, but also the perceived brightness and attack length. Lakatos (2000) mentions this possibility, at least that one could select sounds that were similar in centroid and rise time, but one could go further and normalise the centroid by filtering the sounds and creating an artificial attack by fading in. Then, the sounds would not differ in their centroid or attack time, but perhaps they would still differ in other aspects which might then come to the foreground. Such a study, if conducted, might reveal other timbral dimensions than those that are usually found.

Another possible explanation for the frequent mention of centroid and log attack time as primary timbral dimensions may be that only a small number of signal descriptors were

used in early experiments. As more feature extractors are used, the chances of finding one, or a combination of several that matches the experimental findings increases. Some MDS studies of timbre have been reanalysed, and better acoustic correlates have been found (such as Krumhansl's study, re-examined by Krimphoff et al. (1994)).

In timbre similarity experiments, it may be wise to use a set of sounds that are similar in as many respects as possible. But the notion of timbre as that which is neither pitch, loudness, or duration may function as a straightjacket, in that the range of sounds chosen for studies may become more restricted than necessary. There is an implicit requirement that stimuli used in timbre studies should be pitched in the first place, since timbre is supposed to be a distinguishing factor in sounds with identical pitch. On the other hand, it may not be as revealing to involve heterogeneous collections of sounds in one study. The problem is one of a trade-off between general, coarse descriptions that pertain to large classes of sounds on the one hand, and detailed, customised descriptions that capture subtle differences in a limited class of sounds on the other hand. A practical matter also puts limitations on MDS studies, namely that for a study of N different sounds, $N(N-1)/2$ comparisons will have to be made, assuming the order of presentation is irrelevant, which it has been found to be (McAdams, 1999). If sounds should also be compared to themselves (which can be useful for validating the test subject's responses), even more comparisons have to be made. An upper limit is given by the time it is reasonable to ask test participants to spend listening attentively, an hour being longer than most test participants can stay focused (Hajda, 2007, p. 259). Therefore, it is not surprising that most timbre similarity experiments use about 10–20 stimuli.

Further refinements of MDS techniques include the introduction of specificities and latent classes (McAdams et al., 1995; McAdams, 1999). It may be that certain sounds are unique in some respect, for instance, the hollow sound of a clarinet or the offset of the harpsichord. These traits are captured by incorporating specificities into the MDS model. Latent classes apply to the test subjects. These classes have been found to run across groupings by degree of musical training, but it is not known what determines which class a subject belongs to.

Another approach to timbre is to regard it in relation to source identification. An instrument does have a timbre, now understood as a qualification that serves to identify it as that particular instrument. Even though different notes across its register, and varying dynamics and articulations will sound different, they all share some timbral property that allow a listener to recognise the collection of sounds as emanating from the same instrument.

Further complications of timbre studies include the question whether timbre perception is continuous or categorical. If the attack time of a string sound is varied, a rapid attack may be classified as plucked, whereas a slower attack would sound as if bowed. Similar effects have been observed with sounds from a struck or bowed vibraphone that has been resynthesised with varying attack length (Donnadieu, 2007). Although Donnadieu found evidence that there is a perceptual continuum as a function of attack length, other studies disagree on this point; hence, the question as to whether the perception of attack quality is categorical or continuous remains open.

The concept of timbre is limited, not so much by its various definitions, as by its actual use in "language games", to borrow a term from Wittgenstein. A broader conception of

morphology is needed to capture more aspects of variability among sounds. Schaeffer's attempt to introduce a classification scheme, the typology, and a finer scheme for further description, the morphology, remains valuable resources for thinking about sound.

2.1.7 Multidimensionality of perception

Acoustic features stand in complex relations to perceived qualities. Nonlinear relations such as Fechner's or Stevens' law relate physical intensity to perceived magnitude. Steven's law is expressed as $P = kI^v$, where P is a perceived magnitude, I is the physical intensity, k is a constant depending on the units and the exponent v depends on sensory stimuli (Meilgaard et al., 2007, p. 48 ff). For instance, the exponent for perceived duration is 1.1, for loudness of a 1 kHz tone 0.67, for tactual roughness 1.5, and for visual length 1.0. The convenient fact that visual length perception is, for once, a linear relation has made it popular among experimental psychologists for collecting subjects evaluations of various other stimuli. Apart from this kind of nonlinear warpings, which Schaeffer quite appropriately termed *anamorphoses*, there may be a cross-influence from more than one acoustic dimension on a single perceived attribute.

When two auditory dimensions are varied simultaneously, the variation of one dimension may cause interference in the other. Pair-wise interactions between timbre, pitch and loudness were studied by Melara and Marks (1990), using classification tasks where subjects had to categorise stimuli as rapidly as possible into one of two possible classes. For timbral variation, they used two rectangular waveforms with different duty cycles, one described as "hollow" and the other as "twangy". They found that timbre and loudness were easier to classify when they were positively correlated (with hollow corresponding to soft dynamics and twangy to loud dynamics) than if they were negatively correlated or independent. Loudness was classified slightly faster than timbre. Similar findings were obtained with pitch and timbre.

Pure tones with varying amplitude produce different pitch percepts: for tones below 1000 Hz, the pitch decreases with increasing intensity, whereas for tones above 2000 Hz, the pitch rises with increasing intensity. This effect varies considerably between listeners (Houtsma, 1995). While this example may serve as a good demonstration of the interdependence of acoustic dimensions, it plays a marginal role in most music. Conversely, the equal loudness contours display the dependence of perceived loudness on the frequency of pure tones. These are of greater importance in auditory modelling. Loudness depends not only on intensity, but also on spectral content. In the case of bandpass filtered noise, loudness depends on the bandwidth. Up to the critical bandwidth loudness remains equal, but then, with increasing bandwidth the loudness increases too. Duration also influences loudness to some extent; the best way to increase the loudness of very short sounds in a mix is simply to increase their duration.

Detection thresholds for sinusoids or noise both show a time-intensity trade-off: the shorter the note, the louder it needs to be in order to be detected. The signal level at the threshold of detection as a function of the stimuli's duration follows a power law such that, when plotting the data with logarithmic scales on both axes, the test data fall on a line with slope of about $-3/4$ (Eddins and Green, 1995). This relationship holds for a range of durations up to 0.5 seconds. Theoretical models of this relationship assume

that loudness perception involves a temporal integration of the recent past signal. Such temporal dependence is automatically incorporated in RMS measurements of amplitude. It is worth pointing out that the temporal window of integration may be chosen freely when designing RMS (or similar) amplitude measures, but too long or too short windows may not match the ear's resolution very well.

The duration of stimuli has consequences for the detectability and differentiability of practically all perceptual attributes of sound. One might conceive of various levels of complexity of perceptual attributes grounded on the observation that more complex attributes take longer time spans to grasp. In this hierarchy, vibrato is at a higher (more complex) level than pitch, and melodic contour of an entire phrase on a higher level still.

As Schaeffer very convincingly demonstrated in *Solfège de l'Objet Sonore*, the envelope may be a stronger determinant of the timbre than its spectral characteristics. A piano tone in high register and a flute tone with an exponentially decreasing envelope imposed on it can be made to sound very similar, as can the two instruments when provided with flat envelopes. The sound examples (Schaeffer and Reibel, 1998, CD 2, tracks 25-26) are striking, although there is a possibility that the technical standards of recording and treatment of these sounds conceal some of the cues that might otherwise have been used to identify the flute and the piano.

There is also a complex relation between perceived length and the measured duration of sounds, depending on the amount of complexity or information in it, as Schaeffer already noted: "*La durée musicale est fonction directe de la densité d'information*" Schaeffer (1966, p. 248). Using as sound examples recordings of a metal sheet struck once or repeatedly, followed by its natural resonance, listeners might come to think that the attack portion if prolonged beyond just a single strike was longer than the resonance, whereas in fact the opposite was true. Schaeffer also made revealing observations on reversed sounds that originally consisted of an attack followed by resonance (Schaeffer, 1966, pp. 250–251): The density of information is better spread out over the sound's duration in the backwards sound; the listening becomes more abstract (since the sound source is harder to recognise); furthermore, such sounds are illogical by not being physically plausible. Concepts such as the complexity or information content of a sound are not easy to define; however, some attempts in this direction will be discussed in Chapter 5.

2.2 Feature extraction: An overview

Audio features can be divided into three categories: temporal, spectral, and spectro-temporal. Spatial features may be added to the list such as apparent position or stereo width. Those are less often discussed, and we will not deal with them here; however, a few spatial features will be introduced in Section 6.3.3. Moreover, features are either global descriptors of an extended segment of sound, or they are time-varying, producing new values at some uniform sample rate. Time-varying features are used in feature-feedback systems, but we will also propose a global descriptor that can be used in the evaluation of the timbral variation of a sound (see Section 7.3.2). Further, statistics of time-varying features are often useful, such as their averages and standard deviations.

Temporal attributes specify such things as attack time and temporal centroid. It should be noted that the analysis of many temporal attributes depends on a segmentation of a continuous sound stream. The attack time (i.e., duration of the attack phase) is an attribute that only pertains to an identified note onset or beginning of an event. Spectral attributes include a rich variety, such as pitch, spectral centroid, degree of dissonance, inharmonicity, and spectral roll-off. Some spectral features are calculated on the basis of the Fourier transform, while others make use of Mel frequency cepstral coefficients (MFCC), or a decomposition into spectral peaks and a stochastic residual (as is done in SMS, spectral modelling synthesis, which is briefly discussed in the next chapter). Wavelets are less frequently encountered, but may also be useful for feature extraction (Kostek, 2005). Finally, the spectrotemporal features describe the variation of the spectrum over time. Spectral flux, calculated as the overall difference between two consecutive FFT frames, is the prototypical example, but vibrato and other periodic or irregular modulations also belong to this category. The spectrotemporal set of attributes may be related to grain and allure in Schaeffer's typomorphology.

Many spectral features may be computed using different signal representations, such as the amplitude spectrum, the power spectrum, harmonic sinusoidal components or auditory models (Peeters et al., 2011). These different signal representations however yield correlated features.

2.2.1 Levels of audio features

There is already a large number of audio feature extractors, developed mainly for music and speech description, classification and source identification. Many fields of research have contributed to the proliferation of audio features, e.g. medical studies of voice quality, and psychoacoustics; but most notably, feature extractors find important applications in music information retrieval, typically aided by machine learning. In music information retrieval, the goals are different than those of adaptive sound synthesis or machine listening in interactive music. Nonetheless, some of its terminology can adequately be borrowed.

Three levels of audio features can be distinguished (Polotti and Rocchesso, 2008, ch. 3-4):

- Low-level descriptors are close to the signal and are typically computed with standard signal processing operations such as the Fourier transform and simple statistical processing.

- Mid-level descriptors can make decisions about tonality or provide segmentations into relevantly grouped chunks, such as music / speech distinctions or segmentations of different parts in a song. On this level, machine learning, e.g. artificial neural nets, K-nearest neighbours or similar techniques are used for classification tasks (e.g. Herrera-Boyer et al., 2003; Kostek, 2005).

- High-level descriptors are also known as semantic descriptors. These should provide descriptions that are meaningful for the user, such as characterising the mood of the music as happy or sad.

A slightly different classification is provided by Verfaille (2003), but there too, low-level features are considered close to the signal, and high-level, including pitch, those that are closer to perception. Those low-level features that will be our primary topic can be further classified according to how the signal is processed. The following overview is based on McDermott et al. (2006).

To begin with, there are *time domain* and *frequency domain* features. Several useful features can be obtained from the amplitude spectrum after doing an FFT; those features belong to the Fourier-transform domain. Further, if spectral peaks are extracted, *partial domain* attributes relating to the spectral composition of a sound can be analysed. Another term for partial domain attributes is harmonic attributes (Peeters, 2004). Moreover, there are trajectory attributes among which the temporal centroid occurs; periodic attributes, or descriptors of vibrato and similar modulations with respect to their rate and depth; statistical attributes, such as ratios of high to low values or standard deviation of features.

Further, attributes may be distinguished by their temporal extension. Most of the attributes that we find useful for musical applications capture the dynamically varying aspects of sound, but another strategy, often employed in voice quality research (as will be discussed in Section 2.3.8), is to use average measurements over a number of pitch periods. Such averaging may yield more certain results, but at the cost of ignoring temporal variation, or by demanding that the analysed signal segment be stationary.

For musical signals, most features will vary over time, although some may be relatively static over a unit such as a note or phrase. Thus, it is well motivated to begin with a segmentation of the signal into chunks that are relatively homogenous, and then each of these chunks can be submitted to feature extraction. This approach has been elaborated by Rossignol et al. (1999). As an example, a monodic melody may be segmented into notes, but first vibrato needs to be extracted and suppressed since the pitch deviation may otherwise confuse the algorithm. For more robust segmentation, several features are used in parallel.

The problem of making the transition from low level signal processing combined with machine learning to the humanly understandable level of musical meaning and value is referred to as bridging the semantic gap (Polotti and Rocchesso, 2008). A similar semantic gap also remains between the computer code that specifies an autonomous instrument (or any digital musical instrument for that matter) and its sonic output. One way of narrowing this gap is by extensive experimentation with the synthesis model.

For our present purposes, the low-level audio features are quite sufficient as components of autonomous instruments. Music information retrieval, in contrast, needs to use tools that makes it possible to search for characteristics that distinguishes one popular tune from another, and needs to take higher levels into account. These higher levels include tonality, melodic profile and rhythmic patterns, alongside timbral aspects. Thus, there is a strong attention to what Wishart (1996) called *lattice based composition*, in other words, music with pitches and note onset times locked to a grid or scale of possible values. A number of composers have developed systems that use machine listening with a clear focus on pitch and rhythm (e.g. George Lewis' Voyager system). These are after all of primary importance for score following. However, machine listening may be designed with different purposes in mind, where, for example, pitch and inter-onset intervals are

of lesser significance than a whole array of timbral attributes.

As we will use them, the output of a feature-feedback system does not come chunked into convenient packages corresponding to musical notes or events. If such discrete events take place, the only way to find them automatically is by segmenting the continuous audio stream with an onset detector. However, we will briefly outline some note-level generalisations of feature-feedback systems, which necessitates the development of note-level feature extractors, but we will save that discussion for Chapter 7.

2.2.2 Implementation perspective

The division of audio features into temporal, spectral and spectrotemporal is probably best motivated from a perceptual stance, even though there may be some overlap or unclear cases, but this division can also be carried out as pertaining to the implementation of the analysis algorithm. Then there is the time domain / spectral domain division, and spectral domain techniques can be further divided into those operating on the amplitude or power spectrum, on spectral peaks plus residual, on Mel frequency cepstral coefficients and so on. Further statistical measures (mean, variance, higher order moments) may be extracted from any of these representations.

Certain attributes can be accessed both through time domain and frequency domain methods. Autocorrelation, which is useful for pitch extraction, and the spectral centroid, may conveniently be calculated in either domain. Verfaille (2003, p. 125-6) actually lists at least three qualitatively different implementations of the spectral centroid and some further minor variations.

For the most part, we will make use of feature extractors that are low level, easy to implement, reasonably efficient and that are realtime compatible in the sense that they operate on local segments of the input signal. Although the feature-feedback systems in Chapters 6–7 are not implemented as realtime programmes, the feedback loops make it preferable to introduce as little latency as possible in any processing stage. It is better to have the flexibility to insert a delay between the feature extractor and the feedback loop if necessary.

Block-based algorithms such as the FFT process a group of contiguous samples in a single function call and are frequently used in feature extractors. Apart from the analysis window length, the hop size, being the number of samples between two adjacent analysis windows, is one of the most important parameters since it determines the feature extractor's output sample rate.

If the hop size is a single sample, the algorithm will be called *sliding*, after the sliding DFT (Jacobsen and Lyons, 2003) which has already begun to be used in audio processing (Bradford et al., 2005). The computational complexity of an N-point sliding DFT is $\mathcal{O}(N)$, in contrast to the DFT which is $\mathcal{O}(N^2)$ and the FFT with $\mathcal{O}(N \log N)$. Since the hop size must be one sample, the sliding DFT nevertheless ends up with rather high computational demands. Parallel hardware implementations have been proposed as a solution; this would make it feasible to implement sliding versions of feature extractors that normally would use the FFT. The sliding feature extractors that will be discussed in this chapter and further put to use in feature-feedback systems rely upon time domain processing, which leads to particularly simple processing schemes, since the input sample

rate of a sliding feature extractor is the same as its output sample rate.

When the purpose is to analyse sound, feature extractors should exhibit an intuitive and clear relationship between the perceived sound and the feature value. If this were of no concern, one might as well extract arbitrary features from the phase spectrum, which would help distinguishing among different signals that might however sound indistinguishable to the ear. As will be discussed later, there are seemingly more sensible features that sometimes turn out to be more problematic than suspected (see section 2.3.8), yet, in feature-feedback systems, the need for feature extractors that tightly follow psychoacoustic principles should not be overemphasised. For example, a simple RMS envelope follower may be just as useful for our purposes as a psychoacoustically motivated loudness estimator. Understanding what the feature extractor does to the signals it receives nevertheless simplifies the construction of feature-feedback systems. When a feature extractor is applied to the output of a signal generator, the choice of features to extract should be related to the range of sounds that the signal generator is capable of producing.

Block-based processing, as FFT and most other transform-based analysis implies, introduces a latency which has important consequences for the feature-feedback system's dynamics. Even sliding feature extractors need a certain temporal support. RMS amplitude may be estimated in several ways, such as calculating it from spectral bins, or using a one-pole filter or a moving average filter. The two latter implementations use time domain filters, but the filter's impulse response provides a temporal window of integration which is finite in the moving average case, and nominally infinite when using a one-pole filter. When a feature extractor is inserted in a closed loop, as in a feature-feedback system, such implementation details may cause markedly different behaviour.

There are a number of libraries and toolboxes for feature extraction, sometimes combined with higher level tasks such as segmentation and classification, for example Marsyas (Tzanetakis and Cook, 2000), MPEG-7 (Peeters et al., 2000), or the MIR toolbox (Lartillot and Toiviainen, 2007). The feature extractors used in this thesis are custom made, and written in C++. After all, low-level feature extractors are relatively easy to implement, and the more advanced functionality such as classification and identification is not needed for most of our purposes.

A design issue worth considering is whether a large collection of attributes should be implemented as stand-alone functions that may be called separately or calculated collectively all at once. If a fixed set of attributes are to be calculated, the processing may be optimised by performing several feature extractions in the same loop. In contrast, it may be the case that only a few attributes are needed, and so calculating several others is unnecessary. For the purpose of experimentation it is preferable to have the flexibility of individually accessible functions that calculate only one attribute each. In this way, it is also possible to use different window lengths simultaneously.

2.2.3 Overview

In the feature-feedback systems presented in later chapters, only a small number of feature extractors have been used. The ones that have been tried out are listed in Table 2.1. Crosses in the columns marked time domain or frequency domain indicates whether or not

Feature extractor	Symbol	Range and units	Time domain	Frequency domain	Where discussed
RMS amplitude envelope	A_{RMS}, $\hat{a}$	$[0, 1]$	X	X	section 2.3.1
Instantaneous amplitude	$\hat{a}$	$[0, 1]$	X		section 2.3.2
Instantaneous frequency	$\hat{f}$	$[0, f_s/2]$ Hz	X		
Zero Crossing Rate	*ZCR*	$[0, 1)$	X		section 2.3.3
Fundamental frequency	f_o, $\hat{f}$	$[0, f_s/2]$ Hz		ACF	section 2.3.5
Voicing (harmonicity)	$\hat{v}$	ca $[0, 1]$		ACF	
Formant 1 to 3 balance	*a1a3*	dB	X	X	sections 2.3.7 and 2.3.8
Spectral centroid	$\hat{c}$	$[0, 1]$	X	X	section 2.3.4
Spectral entropy	$\hat{H}$	$[0, 1]$		X	section 2.3.7
Spectral flux	$\hat{\Phi}$	$[0, 1]$		X	section 2.3.10

Table 2.1: List of feature extractors that will be used in autonomous instruments in chapters 6 and 7. ACF (the autocorrelation function) is obtained by an FFT.

an FFT has been used in the current implementation. Apart from that, the separation into time and frequency domain features is not so clear-cut, and feature extractors with marks in both columns have been implemented both ways. A few other features are occasionally used, as well as special purpose combinations of the above items. In the rest of this chapter, the listed features and several others will be described in more detail.

The feature extractors in Table 2.1 are either time domain (RMS, instantaneous features, ZCR, a1a3), based on autocorrelation (fundamental frequency and voicing) or use the amplitude spectrum. As mentioned, some of them may operate about as easily in the time domain as in the frequency domain, although computational efficiency and accuracy may differ. The partial domain features have been left out, but will be treated separately later.

In fact, most partial domain features are inadequate for the analysis of noisy, unpitched sounds, wildly changing pitch contours or very irregular spacing of harmonics. Although this makes them unsuitable for use in many feature-feedback systems, we will take a closer look at some of them in the context of additive synthesis in the next chapter. Until then, let us just summarise some partial domain features and how to extract them.

First the sound is analysed with a spectral modelling scheme, usually a phase vocoder

(Beauchamp, 2007), then a number of partials are identified and tracked to produce time-variable amplitudes and frequencies for each partial. This harmonic part may be subtracted from the original signal, leaving the residual consisting of noise and transients. Several attributes can only be defined in terms of partials (Peeters et al., 2011), or are at least easier to calculate from partials.

- Most importantly, the fundamental frequency is the frequency that fits best to the (harmonic) spectrum.
- Noisiness is the ratio of the residual to the total energy.
- Inharmonicity measures deviation of the partials from a perfectly harmonic spectrum.
- The odd to even ratio is the balance between odd and even partials.
- Tristimulus is a triplet of attributes that measures the relative contribution of the first, the second to fourth, and the remaining partials (Pollard and Jansson, 1982).
- The spectral irregularity may also be calculated on the basis of partials.

It should be noted that most partial domain attributes are easily controllable in sinusoidal additive synthesis models, although some require the synthesis model to be augmented with a noise component. This opens up the prospect of resynthesis and transformation of analysed sounds.

2.3 Low-level feature extractors

Some of the most common, and for our purposes the most useful audio features will be reviewed, but this is in no way a comprehensive collection. Still, the feature extractors listed here are more than enough for our purposes, and we can only make use of a handful of them in later chapters. A few time domain alternatives to common spectral domain methods will be introduced, since this is practical for use in feature-feedback systems.

There is no definite way to split the set of features into time and frequency domain, very low or medium level, and so on. But we progress from more signal-related features to those that are more perceptually motivated.

2.3.1 Amplitude, loudness

Several amplitude detectors have been devised both in analogue and digital domains (Zölzer, 2002, p. 83): The half-wave and full-wave rectifiers, the squared signal and the instantaneous envelope detector. For an input signal x_n, the full-wave rectifier returns the absolute value $|x_n|$, the half-wave rectifier is $\max[0, x_n]$, and the instantaneous envelope is calculated from a Hilbert transform pair x, y: $e^2 = x_n^2 + y_n^2$. All of these detectors find applications in dynamic processing, so they are potentially useful for adaptive effects and synthesis models as well. The instantaneous envelope is useful in many other contexts,

but as a measure of perceived amplitude it is far too fast in its tracking of change. It is however easy to improve it by smoothing its output with a lowpass filter.

The RMS amplitude, measured over a window of N samples, and defined as

$$A_{RMS}(n) = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1} x_n^2} \tag{2.1}$$

is very useful for amplitude measurements. As noted above, there are various possible implementations. In practice, the RMS amplitude (2.1) with arbitrary window length can be efficiently calculated with a moving average filter as follows:

$$\begin{aligned} s_n &= x_n^2 \\ y_n &= s_n + y_{n-1} - s_{n-N} \\ A_{RMS}(x_n) &= \sqrt{y_n/N} \end{aligned} \tag{2.2}$$

Another common variant is to use a one-pole lowpass filter:

$$\begin{aligned} y_n &= x_n^2 + by_{n-1}, \quad 0 \ll b < 1 \\ A_{RMS}(x_n) &= \sqrt{y_n/N} \end{aligned} \tag{2.3}$$

If an FFT is calculated, the RMS value can also be obtained from the power spectrum by taking the average over all bins. However, the benefit of using either the moving average filter or a one-pole filter is that this leads to a sliding algorithm. With the FFT, which usually comes with the constraint that window sizes must be a power of two, arbitrary window lengths N may be obtained by zero padding to the next higher power of two. The flexibility of window length is highly desirable for experimentation with feature-feedback systems. Clearly, the algorithm (2.2) is simpler than taking an FFT and much more computationally efficient than a sliding DFT version would be.

A measure related to the peakedness of the waveform is the *crest factor*, which is the ratio of the peak amplitude to the RMS amplitude:

$$\text{crest} = \frac{\max(|x_n|)}{A_{RMS}(x_n)} \tag{2.4}$$

RMS amplitude is correlated with loudness perception, but does not take the frequency dependence into account. Accurate loudness models usually proceed from filter banks, summing up loudness contributions across the audible frequency range. It is conceivable to design a simplified loudness analyser by inserting appropriate highpass and lowpass filters before the RMS calculation. This might work for a restricted amplitude range since the equal loudness contours are not parallel over widely separated amplitude levels.

2.3.2 Instantaneous estimators

When the signal is known to be a sinusoid or to occupy a narrow band, then the instantaneous amplitude and frequency may be retrieved. The instantaneous frequency is

obtained through a Hilbert transform (e.g. Proakis and Manolakis, 2007) of the signal $x(t)$, which results in the so called analytic signal $z(t) = x(t) + iy(t)$, where $y(t)$ is a version of $x(t)$ whose phase is shifted 90° at each frequency.

The Hilbert transform is implemented as a FIR filter, in this case with 256 taps (due to symmetries and the fact that even filter coefficients are zero, the number of filter coefficient multiplications reduces to 1/4 of the filter order; this and other optimisations makes the filter reasonably efficient). At a sampling rate of 44.1 kHz, the length of a 256 tap FIR filter is about 6 ms.

Usually, frequency is understood as the number of oscillations per some unit time. Instantaneous frequency is a quite different concept that is very useful if the frequency changes over time. The idea behind instantaneous frequency is that a real valued signal $x(t)$ is modelled as a slowly varying amplitude envelope modulated by a complex exponential, $a(t)e^{i\phi(t)}$. If $x(t)$ can be expressed this way, then the instantaneous frequency can be found by taking the derivative of the instantaneous phase $\phi(t)$. Instantaneous frequency may be viewed as the average of the frequencies that make up a compound signal at any moment. Hence an instantaneous bandwidth may also be defined (Boashash, 1992a).

Among a large number of estimators of instantaneous frequency of discrete time signals (Boashash, 1992b), one that is easy to implement is

$$\hat{f}_n = \frac{f_s}{2\pi(x_n^2 + y_n^2)} \left(x_n \Delta y_n - y_n \Delta x_n\right) \tag{2.5}$$

using the Hilbert transform pair x, y and a first order approximation $\Delta x_n = x_n - x_{n-1}$ of the derivative. Locally, this estimate is not very good, but on average, this method follows the actual frequency rather closely. As an example, for 10^5 samples of a constant sinusoid at 1 kHz and $f_s = 44.1$ kHz, the average of the estimated instantaneous frequency is 996.6 Hz, but the RMS error is 246 Hz (these values were calculated after removing an initial transient corresponding to the length of the filter). Better estimators can be made by increasing the Hilbert transformer's length, and by using higher order approximations of the differentiators in (2.5). But the interesting aspect of an instantaneous frequency estimator in this context is that it has the shortest window length one can get away with, while still capturing frequency variations well on average. The assumption is that the signal be a slowly varying sinusoid; for complex sounds this method breaks down, which may be a problem if fast frequency modulation occurs. In Chapter 6, this method of instantaneous frequency estimation will be used in a feature-feedback system, which admittedly is to stretch its domain of intended use beyond safe limits, since fast modulation cannot be ruled out.

Figure 2.1 shows that the instantaneous frequency estimator has a time lag corresponding to exactly half the Hilbert transformer's length. It starts at a zero value and jumps abruptly to a value far above the true frequency after the time lag. The presence of oscillations indicate that this estimator is not accurate without some subsequent smoothing. Similar behaviour is observed for the instantaneous amplitude estimator (figure 2.2), except for a gradual fade-in of amplitude instead of the sudden transition that happens for the frequency estimator.

The instantaneous amplitude is easy to calculate, as already mentioned it is

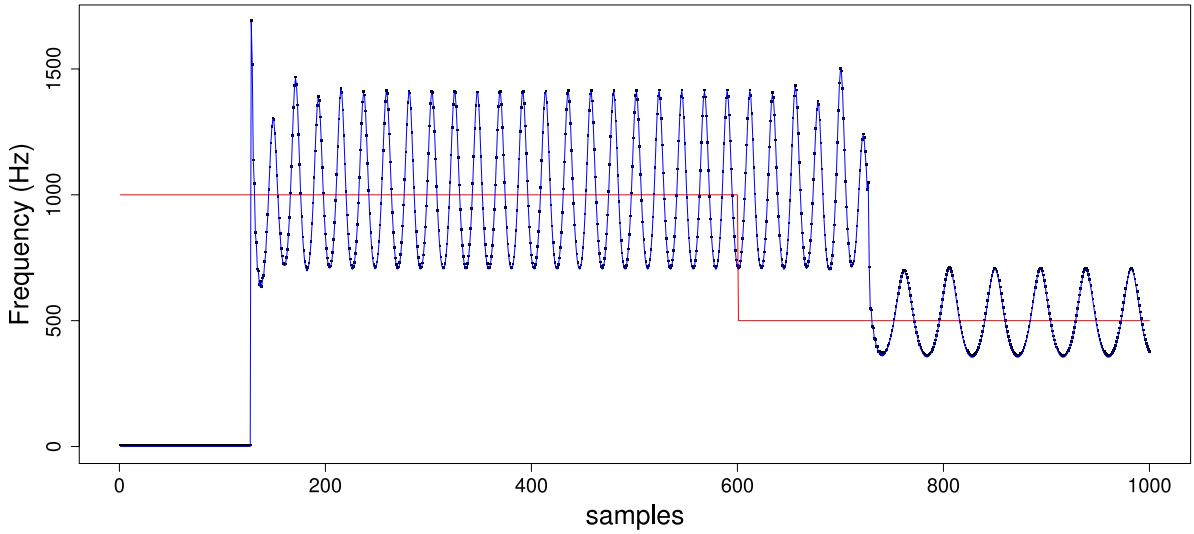

Figure 2.1: Estimation of instantaneous frequency. The red line is the true frequency, 1 kHz for the first 600 samples, then suddenly it changes to 500 Hz. Notice the time lag in the estimated frequency (blue curve) and its oscillations around the true frequency. The oscillations are clearly slower and have smaller amplitude for the lower frequency.

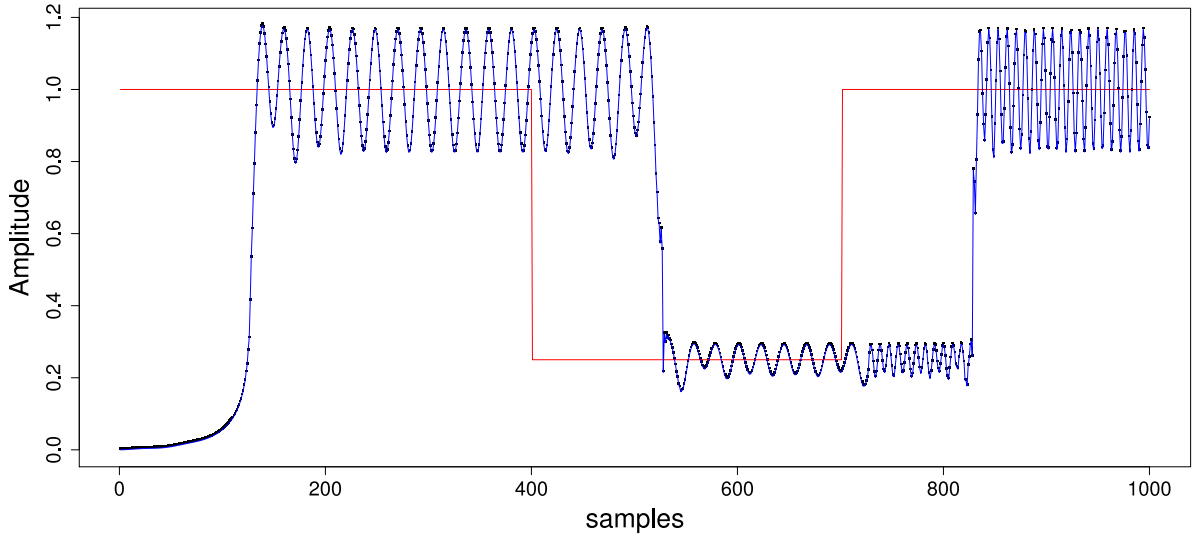

Figure 2.2: Instantaneous amplitude estimation (blue curve) and true amplitude (red line). Note the gradual fade in at the beginning and the varying speed of oscillations; the lag is the same as for the frequency estimator.

$$\hat{a}_n = \sqrt{x_n^2 + y_n^2} \tag{2.6}$$

where x and y are again the Hilbert transform pair of the signal. Since RMS amplitude will be used later, it is appropriate to point out that these are different measures even if the instantaneous amplitude is smoothed. The instantaneous amplitude corresponds to the peak amplitude, which in general is different from the root mean square of a waveform. For example, a sinusoid has peak amplitude 1, but RMS amplitude $\sqrt{2}$, whereas a square wave happens to have the same peak and RMS amplitude.

2.3.3 Zero crossing rate

This is one of the simplest signal attributes. If the signal is a sinusoid, the zero crossing rate (ZCR) corresponds to a frequency $f = \text{ZCR} \cdot f_s/2$, and in general, for a waveshape with p zero crossings each period, ZCR is proportionate to p times the fundamental frequency. Therefore, we shall often use it as a simple pitch estimator in feature-feedback systems.

ZCR gives an average number of zero crossings over a window of length N:

$$\begin{aligned} \text{ZCR}(x_n) &= \frac{1}{N}\sum_{i=0}^{N-1} c_{n-i} \\ c_n &= \begin{cases} 1, & x_n \cdot x_{n-1} < 0 \\ 0, & otherwise \end{cases} \end{aligned} \tag{2.7}$$

For a more temporally accurate analysis, a small N may be chosen. There is a time-frequency trade-off here; however, since the range of ZCR is always $[0, (N-1)/N]$ there will be an increasingly coarse-grained quantisation the smaller N becomes. ZCR and the spectral centroid are both correlated to the presence of high frequency content and thereby to perceived brightness. If they were two perfectly correlated measures of the same thing, plotting the one against the other would always result in a straight line; however, this is not the case.

It can easily be deduced that uniformly distributed white noise has an average ZCR of 0.5. Since ZCR is an indicator of high frequency content in the signal, and since the high frequency content in most music is less than that of white noise, typical ZCR values for music should be rather low as compared to the theoretical maximum of 1. Values higher than 0.5 should only be expected in some synthetic sounds (although this also depends on the sample rate). Such considerations of the theoretically obtainable range of feature values and the expected range given some particular sound source are very important in the design of both synthesis models and adaptive effects that rely on feature extractors.

If the waveform is constant and the signal is monophonic, then the ZCR may be used as a pitch follower. It is only necessary to know how many zero crossings one period of the waveform has. A short segment taken from Xenakis' S.709 is analysed with respect to ZCR using two different window lengths in figure 2.3. Locally, there are quasi-repeating waveforms, but over larger time spans the waveform is constantly metamorphosing. As

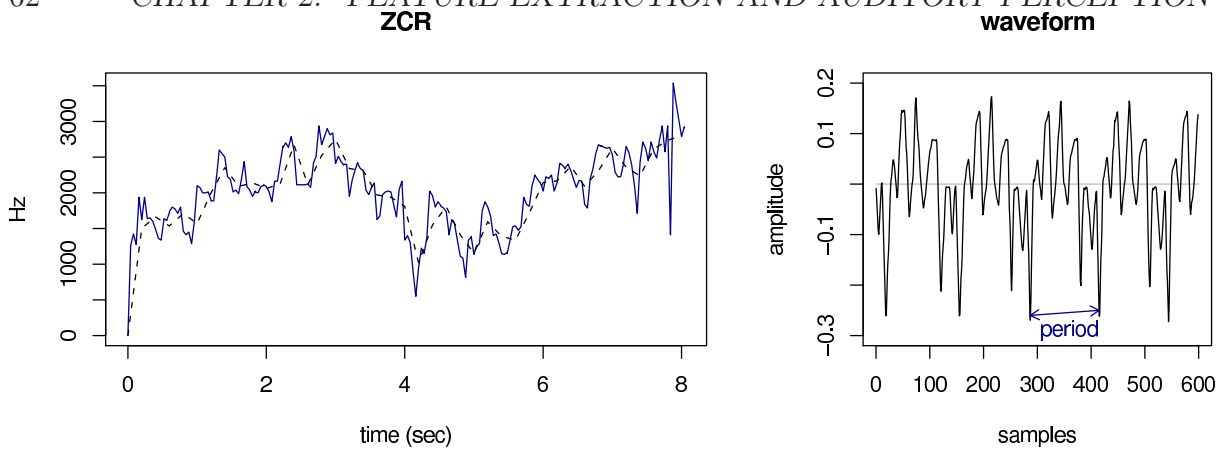

Figure 2.3: Left: ZCR contours of a phrase from S.709 by Xenakis (beginning at about 3'17). The solid blue line uses a 40 ms window, and the dashed line shows a 200 ms window. Right: typical waveform with one period marked. Since there are several zero crossings during each period, the ZCR would overestimate pitch.

can be seen, the longer window traces a smoother curve, but does not capture the rapidly shifting pitches characterising this passage as well as the shorter window does.

The variation of ZCR or high zero-crossing rate ratio (HZCRR) has been shown to be a good measure for distinguishing speech and music (Lu et al., 2001). It is defined as the relative proportion of frames where the ZCR is 1.5 times greater than the ZCR averaged over one second, and its value has been found to be greater for speech than for music.

2.3.4 Spectral centroid

A few different definitions and implementations exist for the spectral centroid. Its most straightforward definition is as the linearly weighted and normalised sum of the bins of the amplitude spectrum:

$$\hat{C} = \frac{\sum_{k=0}^{N/2+1} k|A(k)|}{\sum_{k=0}^{N/2+1} |A(k)|} \tag{2.8}$$

Here and in the following we use the notation $A(k) = \sqrt{X_{Re}(k)^2 + X_{Im}(k)^2}$ where X_{Re} and X_{Im} are the real and imaginary parts of the complex spectrum obtained from an N point discrete Fourier transform. Alternatively, the centroid can be calculated efficiently in the time domain by making use of the fact that the derivative of a sinusoid has linearly increasing amplitude as a function of its frequency:

$$\hat{c} = \frac{A_{RMS}(\frac{d}{dn}x_n)}{A_{RMS}(x_n)} \tag{2.9}$$

Furthermore, if spectral peaks have been identified, the centroid may be calculated on these only. If so, it may have units of (fractional) partial number, otherwise its unit is either Hz or normalised to the unit interval $[0, 1]$. As has been shown (Zölzer, 2002; Verfaille, 2003), various implementations of the centroid yield similar, if not perfectly identical results. The derivative in (2.9) can be approximated by the first order filter

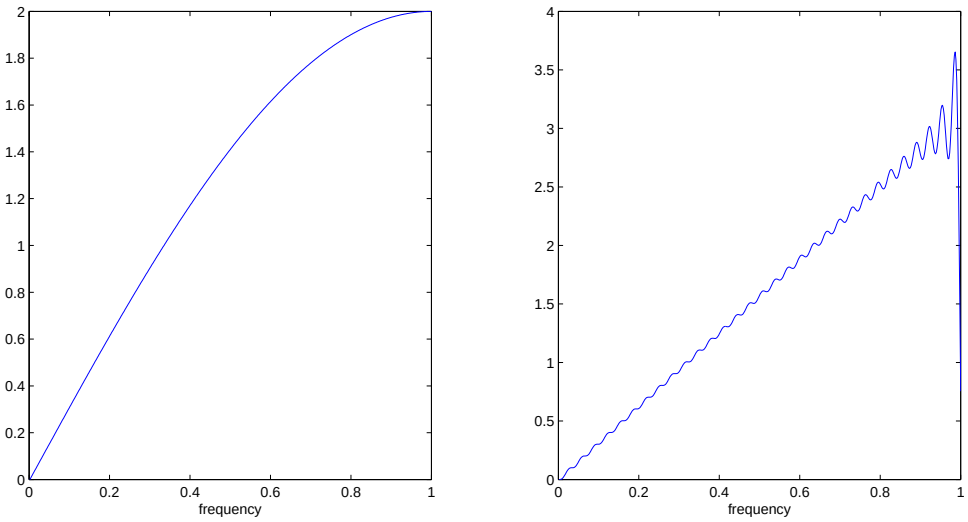

Figure 2.4: Differentiator frequency responses. Left: 2 points, right: 125 points, using a rectangular window.

$y_n = x_n - x_{n-1}$. Better approximations are possible with higher order differentiators (Figure 2.4). Ideal differentiators have a frequency response that is proportional to frequency, and an infinite impulse response

$$h(n) = \frac{(-1)^n}{n}, \quad -\infty < n < \infty, \quad h(0) = 0$$

which can be truncated to any suitable length (and preferably windowed by some function that has better properties than the rectangular one, as shown in Figure 2.4).

Other weightings of the spectrum have been suggested, e.g. the High Frequency Content (HFC) (Masri and Bateman, 1996), which excludes the two lowest bins, and, in contrast with the centroid, is not normalised:

$$HFC = \sum_{k=2}^{N/2+1} k\,|A(k)|^2 \tag{2.10}$$

Among other things, High Frequency Content may be used for onset detection by comparing the HFC in the current frame with that in the last frame. The formula (2.10) also suggests how new features can be designed by defining arbitrary weighting functions that highlight chosen portions of the spectrum. In mixing and mastering, certain frequency registers are identified, e.g. bass, mid-range, presence, air, etc. The energy in these bands may be compared to the overall RMS for a set of descriptors of spectral balance.

2.3.5 Features from autocorrelation

The autocorrelation function is useful for the extraction of some attributes. Since autocorrelation is a way to discover repetition in a signal, it can be used as a pitch follower.

Another attribute is voiciness, which is used in speech analysis to distinguish voiced and unvoiced sounds. We will make frequent use of both of these extractors, which are calculated in the same function call. The partial domain attribute odd to even ratio may also be obtained from the autocorrelation.

As the autocorrelation

$$r_{xx}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x_n x_{n+k}$$

measures the self-similarity of a signal at a given lag k, it has the greatest peak at zero lag, and a local maximum corresponding to the signal's periodicity, if there is any. Pitch estimation based on the autocorrelation is a simple technique, but may fail for several reasons. Octave errors are often encountered, but there is also the time-frequency trade-off to take into account. The autocorrelation function is linear in period length, which means that low frequencies are not well resolved, but with increasing frequency, the resolution improves. Therefore interpolation based on the amplitudes surrounding the peak of the autocorrelation function is used to improve the periodicity estimation.

Voicing is the normalised amplitude at the first local peak (p) of the autocorrelation function:

$$\hat{v} = \frac{r_{xx}(p)}{r_{xx}(0)} \tag{2.11}$$

$\hat{v}$ attains high values, close to 1, for all kinds of steady harmonic complex tones and sinusoids, low values (about 0) for white noise and slightly higher values for pink noise. Sometimes it is called *harmonicity* (Zölzer, 2002) since a harmonic tone will have a high value (close to 1), whereas complex inharmonic tones attain medium values of voicing. Speech in particular is characterised by its rapid fluctuations between voiced and unvoiced sounds, which can easily be identified as peaks and troughs on the envelope of this attribute, provided its temporal resolution is sufficiently high.

The simplest method to distinguish voiced and unvoiced sounds is to set a fixed voicing threshold, typically $\hat{v} = 0.4$. A more elaborate scheme may take the voicing in previous frames into account, thus avoiding possible errors in boundaries between voiced and unvoiced sounds (Markel, 1972).

With so clear correlations to different types of sounds, voicing turns out to be a very versatile attribute. For example, Verfaille (2003) suggests it can be used in an adaptive effect to produce a vowel suppresser, akin to de-essers that suppress sibilants.

The odd to even ratio of the partials may be estimated from the autocorrelation by observing that the even partials have twice the fundamental frequency of the full set of partials. Hence the even partials will contribute to the autocorrelation function at lag $p/2$, where p is again the peak that corresponds to the fundamental frequency. According to Verfaille (2003, p. 127), the odd to even ratio can thus be estimated as $r_{xx}(p/2)/r_{xx}(p)$.

2.3.6 Pitch extraction

Perception of pitch is a complicated process, not least since physical stimuli of quite different character may produce pitch percepts. Harmonic spectra have a clear pitch,

inharmonic spectra may or may not be pitched, bandpass filtered noise is pitched if the bandwidth is sufficiently narrow, amplitude modulated noise has a pitch on the modulation frequency up to at least 100 Hz, and with decreasing precision up to 300 Hz (Houtsma, 1995). Several pitches can be heard simultaneously, as in chords, and in sounds with widely spaced partials, individual partials may become audible, each with its own pitch. Foolproof pitch extractors are by no means trivial devices.

If some constraints are imposed on the signal, pitch detection may become much easier. At the lower end, a single sinusoid with a continuously variable frequency can easily be tracked, either with the ZCR method, or by using a Hilbert transformer as described above.

The autocorrelation method of pitch extraction has already been mentioned. Cepstrum analysis may also be used; the method is similar to autocorrelation in that one looks for the peak of the cepstral coefficients. A third method is the simplified inverse filter tracking. For a discussion of these methods, see Markel (1972).

Spectral peaks are also useful for pitch estimation. Naive methods, such as picking the peak with the highest amplitude, are bound to fail as soon as the fundamental is not the loudest partial. The average distance between spectral peaks should correspond to the fundamental in harmonic tones, whereas frequency shifted harmonic tones may have a pitch that does not exactly correspond to the inter-partial interval. This method is not immune against spectral compositions with skipped partials, such as spectra with only odd partials. Another method is spectral template matching, where the spectrum is matched against a template consisting of a small number of partials. The best correlation of the spectrum with transposed versions of the template gives the estimated pitch. Spectral template matching is somewhat similar to the gathered log-spectrum method (Képesi and Weruaga, 2006), which sums the logarithm of the energy in harmonics of different fundamental frequencies, and the maximum is taken as the estimated fundamental.

The extraction of several simultaneous pitches (multipitch analysis) is a much harder problem. Among several methods in use, let us briefly mention one that was introduced quite recently (Klapuri, 2008). This method is based on an auditory model, which first splits the signal into subbands with a gamma-tone filter bank, then submits each band to dynamic range compression, half-wave rectification and lowpass filtering. These signals are then Fourier transformed and summed to form a combined spectrum. From this representation, a most prominent pitch can be identified and extracted (literally, it is removed from the mixture). Then this extraction process is repeated a number of times based on polyphony estimation, i.e. an appreciation of the number of concurrent pitches in the sound. Multipitch analysis is a necessary requirement for automated transcription of audio, but it has many other applications such as machine listening in the context of interactive music.

At this point we should add that when we refer to pitch extractors or pitch followers in the following chapters, a more correct expression would be “fundamental frequency extractor”. There are two levels of difficulties that one needs to be aware of. First, the estimator may not always yield the correct fundamental frequency; second, even if it does, this may not always be the perceived pitch. Similar caveats apply to other estimators as well.

2.3.7 Attributes for spectral shape

Several attributes exist for describing the general shape of the spectrum in just one scalar value. The spectral slope is the slope of the linear regression line of the spectrum. It could conceivably be calculated on the basis of linear amplitude and frequency scales, but logarithmic scales are better suited for perceptually relevant characterisation. Some common families of waveshapes have spectral slopes that decrease with a constant number of dB per octave; those with discontinuities in the waveshape (square wave and sawtooth) have a -6 dB/octave slope, while those that are continuous but have discontinuous first derivative (e.g. triangle wave) have a -12 dB/octave slope. Various types of noise are also qualified by their spectral slope, most notably pink noise with -3 dB/octave, brownian noise with -6 dB/octave, and white noise which is specified to have a flat or zero slope. The spectral slope, as well as the centroid, are correlated with amplitude in many acoustic instruments, since high partials tend to become more prominent with increasing dynamics.

The spectral roll-off is defined as the frequency below which some fixed proportion of the signal's energy is contained, which is often taken to be 95 %. To calculate the roll-off, first the energy of the spectral frame is computed, and then the energy is summed from the lowest bin until the sum crosses the 95% level (Peeters, 2004). Neither the centroid, the slope nor the roll-off tells everything about the spectral shape, but they all give general measures of the balance between low and high frequencies. Other spectral shape descriptors include the higher order statistical moments; the spectral spread, or variance around the mean, and the third and fourth moments, skewness and kurtosis.

White noise and a sinusoid at $f_s/4$ should have roughly the same centroid, but the sinusoid has minimal spread and the noise has maximal spread. Skewness measures deviations from a symmetric distribution. Negative skewness indicates a spectral distribution with more weight at high frequencies, and the opposite for positive skewness. Kurtosis indicates the degree of flatness or peakedness of a distribution. Another feature related to kurtosis is the spectral crest factor. It is the ratio of the maximum amplitude to the average amplitude of spectral bins within a specified frequency range (Peeters, 2004).

In the context of voice quality research, Ishi (2004) introduced a simple way to characterise the balance of energy in the first and third formant areas. The formant areas were, somewhat arbitrarily, defined as F1: $100 - 1500$ Hz, and F3: $1800 - 4000$ Hz. A measure called A1A3 gives the difference in dB between the spectral peaks with the highest amplitude in each band. For positive values, the lowest formant is the strongest, as is the case in many pitched instrument and vocal sounds. For noisy or whispered vocal sounds, the higher formant may be the most prominent. A1A3 is easy to calculate, and its meaning is readily interpretable. A similar measure, let us call it a1a3, can be implemented in the time domain. Instead of taking the maxima of two formant regions from an FFT, we apply two bandpass filters (BP) centred in the middle of the formant regions, and take the ratio of their RMS amplitude envelopes:

$$\begin{aligned} a1a3(n) &= 20 \log_{10} \frac{a_1 + \epsilon}{a_3 + \varepsilon} \text{ [dB]} \\ a_i &= RMS(BP_i(x_n)) \end{aligned} \tag{2.12}$$

Both filters should have the same Q-factor, i.e. the bandwidths should be a fixed fraction of the centre frequencies. A small number $\varepsilon > 0$ is added to avoid zero division. For use in synthesis models, it is often preferable that the range of an attribute is constricted to a standardised interval, such as $[-1, 1]$ or $[0, 1]$. A logarithmic scale is suitable for this measure, and can still be used although its units are no longer dB. Further parameterisation is possible for this attribute; the two centre frequencies and corresponding bandwidths may be varied, as well as the integration time for the RMS amplitude.

All of the preceding attributes describe general trends of the spectrum. On the other hand, the local deviation from smoothness of the spectral envelope may be worth investigating. Spectral irregularity can be calculated either from a collection of spectral peaks (Krimphoff et al., 1994), or directly from the FFT (Verfaille, 2003). There are various formulations; the one given by McDermott et al. (2006),

$$Irr = \frac{\sum_{k=1}^{K}(a_k - a_{k-1})^2}{\sum_{k=1}^{K} a_k^2} \tag{2.13}$$

uses two adjacent partials, whereas Verfaille used three adjacent spectral bins. This is a good illustration of the current lack of standardisation of feature extractors. The different standards make it hard to compare results across studies.

Another measure that deals with spectral irregularity is the spectral entropy. This is the Shannon entropy calculated over the bins of the amplitude spectrum, which should be normalised to sum up to 1 in order to be interpretable as probability masses. As usually defined, the Shannon entropy of white noise grows depending on the analysed sequence length, but we will use it normalised to the interval $[0, 1]$ instead:

$$\hat{H}(x) = -\frac{2}{N}\sum_{k=1}^{N/2} A(k)\log(A(k)) \tag{2.14}$$

Spectral entropy is highest for a flat spectrum (such as white noise, but also a single impulse or linear chirp) and lowest for steady sinusoids.

Formant frequencies and bandwidths not only distinguish vowels, but they also characterise acoustic instruments. Several methods for spectral envelope extraction and representation are in use (Rodet and Schwarz, 2007). Spectral envelopes can be found from the first cepstral coefficients, or from an autoregressive LPC model.

2.3.8 Features in voice quality research

Phonetics and medical voice quality research have contributed a number of audio features developed for their particular domains. Strictly speaking, some of these attributes only apply to vocal signals, although many are of potential interest in general musical applications as well. A common purpose for many of these analysis methods is to find acoustic and physiological correlates of the audio features. Klatt and Klatt (1990) discuss these correspondences for pressed, modal (normal) and breathy voice. In pressed voice, the glottal pulse is comparatively narrow, which produces a rich spectrum of harmonics. Breathy voice is caused when the vocal folds do not close completely, causing a strong fundamental and steeper spectral slope than in other phonations. A constant

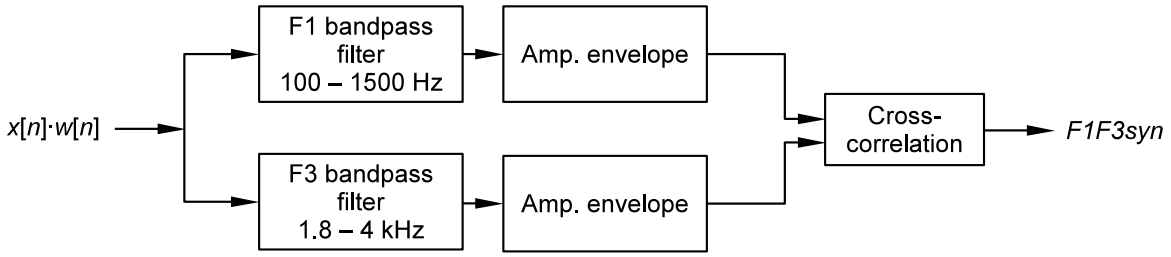

Figure 2.5: F1F3syn processing: the signal is windowed and split into formants. The amplitude envelopes of the two bandpass filtered signals are cross-correlated.

leakage of air also results in turbulent noise when the air stream passes through narrow constrictions. Nasal quality is also characterised by a strong fundamental component. Some languages, such as Gujarati, use the contrast between modal and aspirated vowels (Ladefoged, 2005).

Ishi (2004) introduced a method for the automatic analysis of aspiration noise. The basic idea is to combine two features, the balance between the lower and higher spectral registers, as well as a measure of aspiration noise, which would show up as noise in the third formant area. Since a stronger fundamental is indicative of aspiration noise, Ishi's A1A3 attribute (discussed above) is used for this part. For the analysis of noise in the third formant area, Ishi introduced an attribute called F1F3syn, which is a measure of the correlation of bandpass filtered waveforms from the first and third formants, respectively. Here the same formants are used as in A1A3, i.e., $100 - 1500$ Hz for the first, and $1800 - 4000$ Hz for the third formant. The signal is split into these two bands, and each band's amplitude envelope is calculated with a Hilbert transformer. These instantaneous amplitude envelopes are smoothed with a 1 ms von Hann window. Then F1F3syn is obtained by taking the cross-correlation of the two envelopes (as shown in figure 2.5). Apparently Ishi used Pearson's correlation coefficient:

$$F1F3syn = \frac{\sum_n (x_n - \bar{x})(y_n - \bar{y})}{\sigma_x \sigma_y}$$

where x and y refer to the amplitude envelopes of each band, and σ is the standard deviation. Since the bandpass filtering is carried out in overlapped FFT windows, the attribute is calculated on a frame (Ishi used 32 ms), and advanced with a suitable hop size.

If vocal sounds are analysed with respect to A1A3 and F1F3syn, and these two attributes are displayed in a scatter plot, it turns out that those sounds with perceived aspirated quality tend to focus in a region with relatively low values of both attributes, while the distribution of non-aspirated sounds is centred around higher values. These two regions also overlap slightly. A creaky voice typically produces high values of F1F3syn, while whisper produces low values. But there is often a large spread in the range of values. Now, the question is not so much if this analysis method is good for aspiration noise detection, but whether it is useful in musical applications.

For the purposes of feature-based synthesis, it is desirable to know how to synthesise sounds with a given value of a certain attribute. We have found one such model that

allows a rather precise control of F1F3syn. Two bands of harmonically related sinusoids, F_1 and F_3 in the first and third formants, are generated as follows:

$$F_i(n) = \sum_{k=K_i}^{L_i} \cos(k\omega n), \quad i = 1, 3$$

where K_i and L_i are the lower and upper bounds of the two bands. Now, one of the bands is delayed with respect to the other by a fraction of the waveform's period, so that

$$x_n = F_1(n) + F_3(n - \delta)$$

with $\delta \in [0, T]$ for period $T = 1/f$. With this model it is possible to generate signals with foreseeable values of F1F3syn: zero delay (or a full period) gives the maximum value 1, while half a period's delay gives a minimal value of -0.5. A theoretical minimum value of -1 should be obtainable by other means. Unfortunately, these phase variations do not result in audible differences, at least if the amplitude envelope is ramped in the beginning and end of the tone. This negative result, together with the rather high variability of this attribute in the analysis of other signals, gives reason to doubt its usefulness outside its intended domain.

Hadjitodorov and Mitev (2002) list several attributes in use for pathological voice screening: Fundamental frequency and its standard deviation; amplitude and pitch perturbations (shimmer and jitter); ratio of harmonics to noise energy; degree of hoarseness; normalised noise energy; turbulent noise index; ratio of the energy of the first partial to the rest of the partials; duration ratio of non-vocalised to vocalised part of the signal. While the inferences drawn from collections of these attributes are used in medical diagnostics, the same attributes could be used to analyse virtually any sound. It should be noted, however, that some of these attributes assume a vocal source, with the identification of glottal pulses as a first step. Algorithms for glottal pulse identification will probably work for any pitched signal, with varying degrees of success. Hadjitodorov and Mitev suggest the following procedure, slightly simplified: Beginnings of glottal pulses are identified as the first zero crossing before the maximum amplitude in a segment of the signal corresponding to the longest waveform period, and successive points are determined by finding a maximal autocorrelation within a range of delay.

A measurement technique for hoarseness was suggested by Yumoto et al. (1982). Curiously, it had previously been common practice to evaluate hoarseness from visual inspection of spectrograms of sustained vowels. The cues to look for were noise components in the main formants, high-frequency noise, and loss of energy in high frequency partials. Increasing hoarseness is indicated by the appearance of noise, which replaces the harmonic structure. The measurement assumes that the voice signal can be modelled as a periodic signal with additive noise. First an average waveform is calculated by summing a number of consecutive pitch periods. In this average waveform, the noise component should be cancelled. The quantity of harmonic energy H is calculated from this average waveform. Then the noise energy N is estimated as the energy of the difference of the average waveform and each individual pitch period. The hoarseness measure is then given by the harmonics to noise ratio H/N. Interesting though this method may be, it is not robust against pitch variations such as jitter, and it requires preprocessing over a num-

ber of pitch periods, which presents an obstacle for use in real-time applications. More direct measurements of harmonic to noise ratio can be useful in musical applications of sound analysis and synthesis. However, the calculation of residual in spectral modelling synthesis (see Chapter 3, and Serra and Smith (1990)) is hardly any simpler than this hoarseness measure.

2.3.9 Dissonance, roughness

Consonance, and its dual concept dissonance, have been defined in various ways in different musical contexts. Sethares (2005) lists five distinct uses of the term consonance: melodic, polyphonic, contrapuntal, functional, and psychoacoustic consonance. The first four concepts are closely related to traditional music theory, while the last one (related to sensory dissonance) is independent from, and partly even at odds with music theory. Dissonance is a term that is frequently associated with the simultaneous combination of tonal, pitched and often, but not necessarily, harmonic sounds. But sensory dissonance is a concept that applies to all kinds of sounds, even unpitched noises.

Sensory dissonance is caused by the beating of closely spaced partials. Sethares develops a dissonance measure that builds upon the work of Plomp and Levelt (1965), who studied the dissonance of two simultaneous tones, both sinusoids and complex harmonic sounds. Plomp and Levelt found that the degree of dissonance is a function of the interval as measured in units of the critical bandwidth over a large region of frequency.

In Sethares' model, first the dissonance of each pair of partials in a spectrum is calculated according to the Plomp and Levelt dissonance curve. Next, the contribution of each pair of partials is summed to yield the intrinsic dissonance of a sound. While the dissonance curve for two simultaneous sinusoids is empirically motivated, it is less clear whether sensory dissonance adds up by linear summation. The calculated result will also depend on the spectral peaks and the chosen method for picking them out. Dissonance curves show the degree of dissonance of one spectrum against itself as a function of their transposition ratio (dissonance curves for two different spectra are also possible). Evidence for the soundness of this dissonance measure comes from the agreement between positions of local minima of the dissonance curve of harmonic spectra, and their corresponding positions according to music theory.

A time domain method has been proposed for the calculation of sensory dissonance (Sethares, 2005, Appendix G). Beats between partials is the phenomenon to be captured. First, the input signal is decomposed with a filter bank approximating the critical bands. Next, an envelope detector is applied to each band, followed by a bandpass filter. A half wave rectifier followed by a lowpass filter is used for envelope detection, a construction that is also used in other auditory models. The bandpass filters are tuned to the beating frequency which contributes to roughness, and the output from all channels is finally summed up to produce the overall sensory dissonance measure.

Roughness, being related to sensory dissonance, may be described as a fast modulation in the range $15 - 300$ Hz. A simplified procedure for its calculation is to take the RMS of the highpass filtered RMS amplitude envelope (McDermott et al., 2006).

2.3.10 Spectral flux

Spectral flux differs from other spectral features in that it compares two adjacent frames. The flux measure is close to 0 when there is little variation in the sound, and close to 1 if successive spectra differ much. There are basically two different ways to conceive of flux. It may be thought of as the complement of correlation between adjacent windows,

$$\Phi_c(m) = 1 - \frac{\sum_k A(m-1,k)A(m,k)}{\sqrt{\sum_k A(m-1,k)^2}\sqrt{\sum_k A(m,k)^2}} \tag{2.15}$$

where $A(m,k)$ denotes the amplitude in frame m, bin k (Peeters, 2004). The other perspective on flux is to consider it as the magnitude of the time derivative of amplitude in each bin, averaged over frequency:

$$\Phi_d(m) = \frac{\sum_k |A(m-1,k) - A(m,k)|}{\sum_k A(m-1,k) + \sum_k A(m,k)}. \tag{2.16}$$

Flux may also be calculated on a logarithmic amplitude scale, as Lu et al. (2001) have proposed.

Since flux measures variability over a certain time span, the distance in samples from one frame to the next is a crucial parameter. Overlapping windows must be avoided since using partially the same set of samples will introduce spurious correlations. The standard option is to take adjacent windows. Then the flux measure is parameterised by window length. It can be interesting to compare flux measures taken with different window lengths. Another alternative could be to study how flux depends on the spacing between analysed frames, allowing gaps between them. This idea is related to self-similarity and recurrence plots, both of which are useful tools for the analysis of formal aspects of longer signals such as recordings of complete musical pieces (see Section 7.1.1).

For stationary stochastic signals such as white or pink noise, the level of flux does not depend on the window length or type. White and pink noise have average values $\Phi_d \approx 0.29$ over window lengths at least from 0.01 seconds to over one second, whereas $\Phi_c \approx 0.21$ for both kinds of noises. The variance in average flux is greater for pink than for white noise. In general, the two flux measures differ and one has $\Phi_c < \Phi_d$.

Completely different results are obtained with non-stationary signals if the time-averaged flux is plotted against window length. An analysis of flux dependence on window length is shown in Figure 2.6, where Xenakis' piece S.709 is again used for illustration.

As is customary in scaling analysis of this kind, a log-log plot is used. The plot shows two separate scaling regions: for windows shorter than about 0.1 seconds, the flux increases with an almost constant slope, whereas for windows longer than 0.2 seconds the flux reaches a saturated level. The reason for using a log-log plot is that if a power law $y \propto ax^r$ is assumed, then this will look like a straight line on a double logarithmic graph. Note that the hop size in Figure 2.6 is identical with the length of the window applied to the signal; zero padding to the nearest greater power of two is used for the FFT. Finally it should be mentioned that the analysis is performed in two passes, where the second pass starts from half the hop size in order to reduce potential effects of spurious alignments of the window with periodic changes in the signal. Notice that if the correlation-like flux

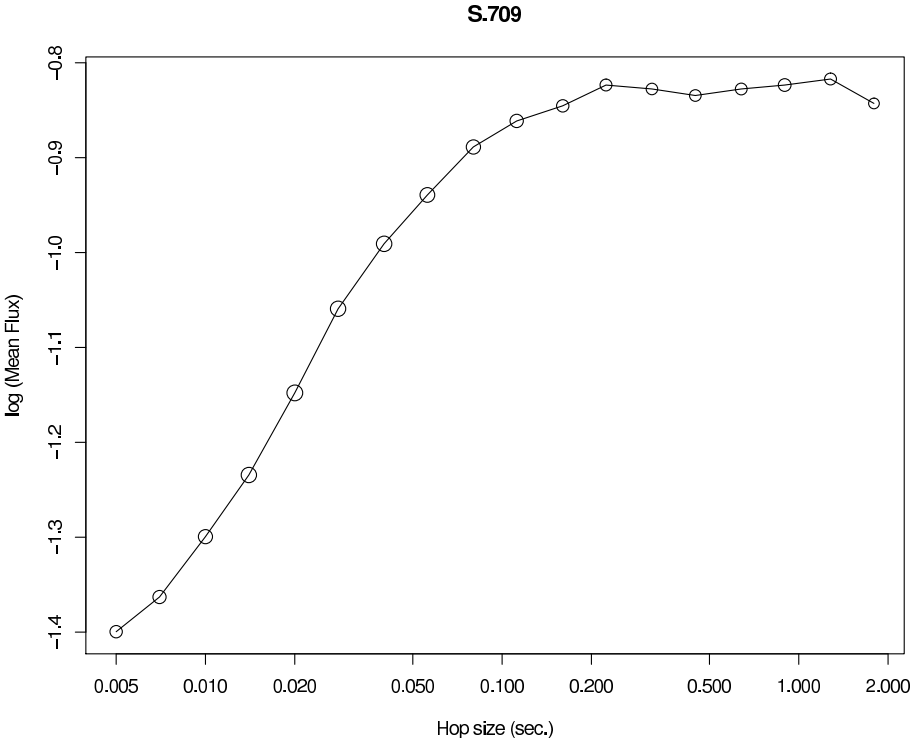

Figure 2.6: Mean flux (Φ_d) of S.709 by Xenakis as a function of hop size levels out for windows longer than 0.2 seconds.

$\hat{\Phi}_c$ is taken as the measure, then the figure gives essentially the same information as the analysis of mean autocorrelation—all one has to do is to flip the figure upside down.

Although flux is most obviously defined in terms of the short time Fourier transform, it is possible to construct a sliding version in the time domain by using a filter bank. In that case, one can easily take advantage of the flexibility of spacing of centre frequencies, and set up a bank of logarithmically spaced bandpass filters, e.g. with octave intervals. Ideally, the filters should have a flat magnitude response when all their outputs are combined, otherwise certain frequencies will contribute more than others to the flux estimation. This leads to a constant-Q version of spectral flux, in which the lag between the current and the delayed analysis windows scales as the inverse of the filter's centre frequency. Some preliminary experimentation has been carried out with this constant-Q spectral flux, but its properties and implementation details need to be further investigated.

Spectral flux appears to be the only commonly encountered spectrotemporal feature; however, other features such as glissando and vibrato are best measured over two or more spectral frames (see also Section 7.1.9).

2.4 Concluding remarks

The review of low-level feature extractors has focused on descriptors that are related to spectral shape, relative strength of partials, pitch, amplitude and the degree of noisiness. These are all such attributes as may be extracted continuously from the signal. Other

attributes pertain to temporal aspects of delimited segments of the signal, such as the attack length of a tone. Onset detectors are needed for the proper extraction of temporal features. For the purposes of building feature-feedback systems, however, we will rely upon feature extractors that deliver a signal at a constant rate. Nevertheless, onset detection and segmentation could be valuable additions to the tool-box of feature-feedback systems, at least insofar as their output contains any onsets to be detected.

The correlation of acoustical properties of sounds to feature extractors and to the perception of sound has been discussed from different angles throughout this chapter—from Schaeffer's typomorphology to more controlled experiments with timbre perception and what feature extractors can reveal about sound. In fact, there is a very simple and enjoyable way to explore these relations, which we turn to now.

2.4.1 How to listen to feature extractors

Concatenative synthesis may be used to explore collections of sounds by retrieving them according to certain criteria. The idea behind concatenative synthesis, originally used in voice synthesis, is to string together short excerpts of sounds into a coherent signal that matches some target sound (Schwarz, 2006, 2007; Sturm, 2004). There is a data base, the corpus, from which matching fragments are fetched. The target sound is analysed over short segments, and corresponding segments are sought from the corpus given certain constraints, especially that of minimising some distance function between the target and corpus segments.

Instead of using a target sound, it may be interesting to browse the sounds in the corpus by sorting them according to values of some chosen features such as centroid, spectral entropy, etc. By sorting the fragments this way, one can in effect audition a transition from low to high values of that feature. Such listening experiments can prove very illuminating in developing an intuitive understanding of what the features "sound like". Very different sounding audio fragments may share the same value of some feature extractor. If the feature extractor has an unclear relation to perceived sonic qualities, or if it only works for a certain class of sounds, then one might sometimes experience that two sound fragments that sound similar turns out to have widely differing feature values (see the discussion of F1F3syn in Section 2.3.8).

In some cases, there are several slightly different feature extractors that perform almost the same task, or put differently, are more or less good at some specific tasks. The spectral centroid is known to be correlated with the zero crossing rate, although it could be interesting to study their difference. This can be done by sorting the same corpus in turn with the centroid and the ZCR and listening to the two output sound files.

The balance between noise and pitched sound can be measured with voicing. But spectral entropy turns out to be another efficient way to distinguish noisy sounds from pitched sounds. This works since noise has a flat spectral envelope, whereas pitched sounds have spectral peaks. If one wants to know the difference between spectral entropy and voicing, then one would have to select sounds that are more or less constant in one attribute and sort the corpus according to the other.

Example 2.1. If one sorts a heterogenous collection of sounds by increasing spectral entropy, those with clearest pitch come first, succeeded by the most noisy sounds at the

end.

Sorting according to a single feature is straightforward. Two or more features may be combined, and then sorted according to the level curves of their sum or difference. For instance, the variation of combined features such as centroid and spectral entropy may be investigated by sorting according to $\hat{c} + \hat{H}$. Other sorting criteria could be used, such as requiring that the distance between two features is always within some limit.

Notice that low voicing should correspond to noisy or aperiodic sounds, whereas in contrast, low spectral entropy is associated with few and sharp spectral peaks. These are not mutually exclusive properties; a sound may, for instance, have two sinusoidal partials at an inharmonic ratio, which would result in low $\hat{H}$ and relatively low $\hat{v}$. Sounds with simultaneously high voicing and spectral entropy can also be found, such as spectrally rich bass notes.

Of course the sound examples one may construct by this procedure depend on the corpus. It would be very interesting and useful to investigate the psychoacoustic correlations of various features this way. It is not known today what kind of scale would be useful to measure linear increasing amounts of perceived noisiness if measured by voicing or spectral entropy. Most likely, the just noticeable differences will have unequal sizes depending on the value of the feature. Influences from other features than the one under study would need to be reduced as far as possible.

It is an illuminating experience to listen to the result of sorting a corpus into an increasing sequence of some feature, particularly so if the corpus contains a rich and varied collection of sounds.

2.4.2 Acoustics and feature extraction

New features and their analysis algorithms may be constructed by combinations of existing features. What makes certain instrumental sounds characteristic is probably not so much which values they happen to take on certain acoustical dimensions, but the relation that links one dimension to another. In flutes, increasing pitch is followed by increasing loudness; piano spectra are rich in the bass register and have fewer harmonics in the higher registers, etc. Couplings of this kind are very easy to incorporate in synthesis models once they are known. Therefore, analysis of such relations may be an effective way to find relevant design principles for synthesis models. This is true even if the goal is not actually to model existing sound sources.

Feature extractors may be strongly correlated (either positively or negatively) for two reasons. First, they may measure almost the same thing, but in different ways. For example, the ZCR, centroid, spectral roll-off and a1a3 are all in various ways sensitive to the presence of high frequency energy. If two descriptors measure exactly the same thing but in different ways, then there is little reason to use both, but otherwise they are often useful for discriminating among similar sounds. Second, some sets of features may seem to be correlated simply because most musical or speech sounds happen to be correlated across different acoustic dimensions. With some ingenuity, synthesis models may be constructed that allow for the independent control of those features. The well-known correlation between amplitude and brightness or spectral richness in most acoustical instruments does not even require very clever solutions for their independent control.

Verfaille (2003, p. 145) briefly discusses both of these reasons for correlations, and points out that some sets of feature extractors are even redundant. This is the case in particular with different implementations of the same feature, such as the frequency and time domain versions of centroid. The study of Peeters et al. (2011) investigated correlations and redundancies among a large set of features by analysing a highly varied set of musical instrument sounds. They recommend to select features that are mutually independent and suggest using four groups of features as a minimum, including a measure of the central tendency of the spectrum (such as centroid or roll-off); a measure of temporal variability (spectral flux); a feature related to the temporal energy envelope; and one feature related to pitch or periodicity.

Feature extraction of simultaneous sound sources is a hard problem if one wants to access the features of the individual sources rather than the mixture. In human auditory perception, we usually have no difficulty assigning different sound sources, each to its own stream. This is a basic requirement in order to be able to perceive more than one timbre at once. Most of us would probably have no trouble characterising, say, the voice of a person as nasal, and the timbre of a flute as breathy, even if the two were heard simultaneously. If similar achievements were to be made in automated feature extraction, advanced techniques such as blind source separation (e.g. Kostek, 2005) would have to be the first step. It should be noted that the feature extractors discussed in this chapter (with the sole exception of the multipitch extractor) assume a single input source, lest the input from several sources be blindly treated as if emanating from the same source.

Orchestration, on the other hand, is very much about creating novel timbres by blending the sound of individual instruments and making them indistinguishable in the mixture. If the problems of timbre perception are combined with those of auditory scene analysis—stream integration and separation—then it seems that very much is still unknown about our auditory perception.

Among the different purposes of feature extraction, one is to make inferences about the sound source. This is clearly seen in medical voice diagnostics, or in tasks of instrument recognition. In feature-based synthesis, audio features may be used to find a set of synthesis parameters that match the analysed sound. Feature-feedback systems include feature extractors by definition, but they are useful for describing the dynamics of any complicated synthesiser in terms that are often easier to relate to auditory perception than the synthesis parameters are.

For the analysis of sounds, a set of complementary audio features is useful. In the next chapter we discuss an application of audio feature extraction, namely, how to map analysed features to synthesis models. The analysis—transformation—resynthesis paradigm is one way to use analysed features, but less rigourous strategies will be exploited in which the resynthesis of sounds is not the goal.

2.4.3 Conclusion

The conception of timbre as that which is neither loudness nor pitch was argued to be inadequate for the description of sounds. As Bregman (1990, p. 92) has remarked, either timbre must be restricted to sounds that have a pitch, or there is something wrong with the definition. If only pitched sounds can have timbre, then sounds such as the

scraping of a shovel in a pile of gravel or the sound of a tambourine have to be left out, as well as the sounds of many a nonstandard synthesis technique. Typomorphological criteria such as mass profile, dynamic, grain and allure are much more specific than the almost all-encompassing wastebasket term timbre. It is precisely on this level that Schaeffer's typomorphological terms must be understood: as general musical dimensions alongside with pitch, loudness and duration. From the typomorphology, the idea of redundant, balanced and excentric sound objects is well worth retaining when we discuss the complexity of music in Chapter 5. Also, the concept of reduced listening will re-enter the discussion in Chapter 8.

In parallel to a narrow conception of timbre as pertaining to the steady-state spectrum, low-level feature extractors operate locally on short time frames where they provide a very restricted view of the signal. Musical processes that span over longer durations, from a few seconds to minutes, are not directly accessible at this level without further processing. This is something to bear in mind when feature extractors are put into feedback loops and given the task of making observations of the generated output signal of a feature-feedback system. An additional layer of higher-level feature extractors may be helpful in the design of autonomous instruments. Even in order to detect that the instrument's output is rather static over time, one needs to combine feature extractors from more than a single temporal frame, as will be demonstrated in Chapter 7.

The intuitive understanding of the relation between audio signals, various signal descriptors and the perception of sound has been discussed through this chapter. Obviously, it is useful to know what the feature extractors do and what they imply in perceptual terms. Maybe the time has yet to come when we read off a few feature extractor values and get an idea about the sound they represent, in the way we read off the temperature on the thermometer. Nonetheless, feature extractors are useful for studying how synthesis parameters affect the sound in various ways.

Both sliding (time domain) and block-based (using the FFT) feature extractors have been implemented for efficient use in feature-feedback systems; sometimes the same feature can easily be analysed either way. Sliding feature extractors in particular fit neatly into feature-feedback systems. In all cases, it will be useful to be explicit about the implementation details. This is necessary in order to be able to calculate the range of possible values of the feature and typical values given certain classes of signals. In Chapter 3, more will be said about feature extractors and their relation to signal representations.

Chapter 3

Synthesis Models

The central component in a feature-feedback system is a signal generator, in other words a basic synthesis model that generates the output signal. After the previous discussion of feature extractors and their relation to perceived qualities, this chapter considers a few chosen synthesis models and how they relate to various features. In particular, additive synthesis and nonlinear models will be considered in this stocktaking of synthesis models that may be suitable for use as signal generators in feature-feedback systems.

It has become customary to broadly divide synthesis models into the three categories of spectral, physical, and abstract models. The first two categories have seen much development since the 1990s, while abstract models almost seem to have fallen into disgrace amongst researchers—with a few notable exceptions. *Nonstandard synthesis* is another term that is sometimes used for, as the label indicates, things that do not fall neatly into other categories (Roads, 1996). Xenakis' GENDYN program and Herbert Brün's Sawdust family of works are often relegated to the nonstandard category. Physical modelling, although an important category of synthesis models, will not be dealt with in this chapter; however, a brief example is given in Chapter 4.

Usually, synthesis using spectral models proceeds from the analysis of input sounds, but here we will instead consider "synthesis by rule" from specified feature values. By additive synthesis, sounds can easily be generated from prescriptions using spectral and partial domain features. However, some features are partly overlapping or correlated, which means that conflicting demands may arise. This problem is addressed below in Section 3.2.

Abstract models, including FM and waveshaping, are typically nonlinear in the sense that linear changes of control parameters do not correspond to linear changes in the amplitude of partials. Nonlinear synthesis models may sometimes be hard to relate to audio features. Still, they provide attractive alternatives because of their simple implementation and, in many cases, their powerful control parameters. Section 3.3 gives an overview of some nonlinear synthesis models including those that will be used in Chapter 6 and 7 as signal generators in autonomous instruments. Indeed, if feature-feedback systems may be thought of as nonstandard or abstract synthesis models, then, why not build them on top of abstract synthesis models used as signal generators?

The note concept is prevailing in most discussions of synthesis models. The sound is assumed to have a beginning, the attack, followed by a steady state portion, leading to a

final decay. This way of conceiving of sound synthesis is too restricted in the context of autonomous instruments and needs to be complemented by considerations of higher levels such as phrases or textures. This is particularly important in feature-feedback systems where the note concept is abandoned.

Most of the synthesis models reviewed in this chapter are well established and of widespread use. The basic recipes can be found in the classic reference work of Roads (1996); see also Tolonen et al. (1998) for a survey of spectral and physical modelling. The recent developments in sound synthesis consist mostly of slight amendments to long established synthesis techniques. However, one of the interesting frontiers of today's research deals with how to connect feature extraction and synthesis or audio effects. Some of those attempts will now be reviewed.

3.1 Synthesis with feature extraction

Several related strategies have been developed that unite feature extraction and synthesis models or audio effects. In some cases, such as concatenative synthesis or feature-based synthesis, the goal is usually to imitate a target sound, whereas adaptive audio effects uses the extracted features in order to control effect parameters in synchrony with the input signal.

It is no coincidence that feature extractors often involve a spectral model. The short-time spectrum is well suited as a basis for perceptually valid representations of signals, although it is not the only possible alternative. As spectral models are successively refined, they may also be used for representation of higher-level audio features. Some signal representations and their use in feature extraction will be discussed in Section 3.1.4.

3.1.1 Feature-based synthesis techniques

A very general way to relate sound analysis and feature extraction to synthesis is to allow any conceivable mapping from analysed features to control parameters. In other words, these mappings may occur within as well as across sonic dimensions, say, from brightness to the speed of amplitude modulation. Although the original sound may no longer be recognisable in the new synthesised sound, it is likely to carry traces of the rhythm of the first sound. On the other hand, several strategies have been developed for modelling a target sound using a given synthesis model.

Evolutionary computing has been popular as an aid in the search for adequate synthesis techniques and parameters capable of more or less closely reproducing an arbitrary target sound. In this case, the amplitude spectrum of a segment of the target sound may be chosen as the goal which the fitness measure is compared against. Andrew Horner (2003) noted the irony of the circumstance that as interest in synthesis techniques such as FM declined, researchers began using evolutionary computing methods to optimise the parameters of FM and wavetable models. Such optimisations has made it possible to obtain high quality resynthesis of some instrument sounds.

A related strategy is *feature-based synthesis*. Hoffman and Cook (2006) explain the motivation: While similar to imitative synthesis, where the goal is to recreate a target

sound, feature based synthesis uses a set of audio descriptors either extracted from a sound file or specified by other means. Then, any synthesiser may be used for resynthesis. Its parameter space is searched for sounds that match the input features according to some distance metric. Matching a single feature to synthesis parameters may be trivial, as Hoffman and Cook note, but as soon as several features are combined the problem becomes hard. Iterative search procedures are needed, which implies offline processing. Realtime performance may be accomplished by storing matching feature and synthesis parameter values in a data base (Hoffman and Cook, 2007). As an example application, they mention a simple synthesiser with four sinusoidal oscillators and bandpass filtered noise, which is matched to centroid, harmonicity and five cepstral coefficients. The first two of these features can be mapped to any two-dimensional controller, and they suggest taking the cepstral coefficients from a voice source, thus allowing a very immediate and intuitive control.

Similar work has been carried out by Park et al. (2007) under the name of *Feature Modulation Synthesis*. They developed an application that extracts a set of features from a sound, which can then be modified prior to resynthesis. Their model includes features such as the amplitude envelope, spectral centroid, spectral spread, harmonic expansion or compression, inharmonicity, shimmer, jitter, and spectral flux.

In fact, besides being a convenient way to transform sounds, feature modulation synthesis can be used as a tool for musicians to learn to articulate verbal descriptions of timbre (Park et al., 2008), much in the same vein as discussed in the previous chapter. However, the interdependence of features is something they acknowledge; it may not be possible to arbitrarily specify the value of one feature without affecting those of other features. This problem will be addressed in great detail below in Section 3.2. The modulation of a single feature may be quite simple, such as increasing the spectral irregularity by highpass filtering the amplitude bins of the spectrum as if it were a time domain signal (Park and Li, 2009). Cross-synthesis and morphing are other useful applications of feature modulation synthesis.

Poepel and Dannenberg (2005) introduced a technique they called audio signal driven sound synthesis, which uses feature extractors such as pitch trackers and envelope followers to control synthesis parameters in FM or subtractive synthesis. In their approach, the original signal may also be used directly in a delay line, where the delay length is controlled by the input signal itself. These ideas were then picked up by Lazzarini et al. (2007), who called it adaptive FM synthesis. They have followed up this work with more focus on FM and other nonlinear techniques rather than the feature extraction side (e.g. Lazzarini et al., 2009b). Adaptive synthesis is similar to adaptive effects in its use of input signals that are analysed. If one were to remove the input signal and replace it with a feedback loop from the signal output, then adaptive synthesis in fact becomes a feature-feedback system.

3.1.2 Concatenative synthesis

Concatenative synthesis also relies upon an analysis of the target sound, which is reconstructed from snippets of sound taken from a large sound data base (the corpus), where each fragment is matched according to analysed features or the spectrum at each moment

(Schwarz, 2007, 2006). Originally, the technique was developed for speech processing, but more recently (in the last decade) it has caught the attention of the music signal processing community (Schwarz, 2000). This technique is often restricted to resynthesis by sampling and possibly minor sound processings if this helps to bring the closest matching sound fragment even closer to the target. Sometimes, the succession of matched segments is taken into account in order to produce smooth transitions, which may be done by prioritising the use of adjacent segments from the corpus in the synthesis (Zils and Pachet, 2001). In fact, granular synthesis and sampling are related to concatenative sound synthesis although they typically do not use feature extractors. However, adaptive granulation where the grain selection depends on an audio feature can prove a forceful extension to basic granular synthesis. Other terms used for concatenative synthesis include *soundspotting* (Casey, 2009) and music mosaicing or *musaicing* (Zils and Pachet, 2001).

Apart from realistic sound rendering, concatenative synthesis has other, more artistic uses, as demonstrated by Bob Sturm (2004). By restricting the content of the corpus to certain kinds of recordings, the target can be re-orchestrated and given a new, coherent character. Some of Sturm's entertaining examples include rendering a speech by George W. Bush with howling monkeys, or re-orchestrating Schönberg's string quartets with solo recordings of Anthony Braxton on saxophone. Although the target might well be unrecognisable, some aspects of its morphology will be captured, such as loudness and pitch register, depending on which feature extractors are used. In fact, it has become common to trace the aesthetic history of concatenative synthesis back to John Oswald's plunderphonics and other examples of micro montage where very short sound fragments from various sources are assembled manually into an entire composition (Sturm, 2006). The main difference in artistic uses of concatenative synthesis is that it can be done automatically, thus alleviating the compositional process significantly. We will return to some applications of concatenative synthesis in Section 7.4.

As a matter of fact, variants of granular synthesis have long been popular among composers (Roads, 2001; Wishart, 1994). In particular, the granulation of recordings is one of the most frequently heard processings used by electroacoustic composers. Although possible already by means of tape splicing as in Cage's *Williams Mix* (1952) or Stockhausen's *Étude* (same year), the technique has proliferated since computers have automated the tedious scissors work. In its simplest form, the technique consists of selecting grains from the input sound file and applying a window function and then assembling them perhaps in a reshuffled order in the output sound file. The important parameters are the window length and the amount of overlapping in the output. Often the onset times of successive grains are more or less randomly perturbed. However, if the granulation parameters are applied blindly to the input one misses the opportunity to design the processing adaptively according to the features of the input. Different sounds may require different parameter settings for the granulation process to yield the desired result. Di Scipio has used granulation of recordings or live audio with processing parameters partly controlled by feature extractors in some of his works. His series of processed recordings from four cities, *Paysages historiques* (1998–2005), and the *Audible Ecosystemics* series, are good examples of this.

3.1.3 Adaptive digital audio effects

Although all audio effects process an input signal, whereas only some synthesis models begin with the analysis of audio signals, there is really no absolute split between synthesis models and effects. If the synthesis model requires huge amounts of control data derived from a source sound, then the difference from effects processing is merely one of degrees. Looking into modern hardware synthesisers, often a final layer of effects processing is integrated for the purpose of adding gloss to the sound. Reverberation or room simulation effects in particular serve this function. There is nothing in the way for the application of effects as components *inside* synthesis models as well. This is quite natural in any unit generator based synthesis language where different components may freely be patched together. In particular, adaptive effects may be used as components of synthesis models; if connected the right way, they will qualify as feature-feedback systems.

Adaptive audio effects, as previously mentioned, use feature extractors on the input sound to control effects parameters. Verfaille and others have developed a few adaptive effects (Verfaille and Arfib, 2001; Verfaille, 2003), but there are historical precedents. Dynamic processing, including compressors, expanders, noise gates and de-essers, are all good examples of the principle: an envelope follower determines the amount of gain to apply to the signal.

Apart from dynamic processing, there is another category of audio effects, arguably qualifying as adaptive, that were explored before the term was introduced. Trevor Wishart (1994) developed a group of techniques that took the *waveset* as its primary, "atomic" unit. A waveset is a segment of the waveform delimited by two consecutive zero crossings going in the same direction. These techniques rely upon the identification of zero crossings, which is a rudimentary form of signal analysis. Wishart introduced several techniques of transforming sounds based on this zero crossing analysis, most of which cause drastic distortions of the original sound. These include waveset omission, interleaving, timestretching and several others.

In *content-based transformations* (Amatriain et al., 2003), the idea is to bring in higher-level features in the tuning of effect parameters, although the approach is virtually the same as that of adaptive effects. The basic idea can be applied in flexible ways to yield the results that any specific musical situation requires. If the task is to modify certain parts of a sound file, say, to apply a flanger to those parts which are noisy, a noisiness feature extractor can be applied to identify those portions, and the flanger may then be applied with the modulation depth or wet/dry balance controlled by the degree of noisiness. So, instead of providing long lists of useful adaptive audio effects of proven value, it may be preferable to accommodate for the free combination of feature extractors and effects. From the user's point of view, it is likely better to have access to a whole range of effects and feature extractors to be combined ad hoc for any conceivable purpose.

Just to mention a few interesting ideas of what can be done with adaptive audio effects, we list the following:

- Adaptive tape transposition where the amount of interpolation depends on the signal's amplitude;

- Ring modulation with pitch tracking—making it possible to lock the modulation

frequency to the pitch;

- Time stretching of steady parts of the sound, leaving transients untouched;
- Change of prosody and voice quality (including gender and age) in speech and song.

It should be mentioned that there are further possibilities than just using the same input sound both for feature extraction and as input to the effect unit. As in side-chaining, commonly applied in dynamics processing, the sound controlling the effect parameters may be different from the one that is processed. Gestural real-time control of adaptive effects has also been considered; for more details on all of this, see Verfaille (2003).

3.1.4 Signal representations

Additive synthesis is particularly well suited for the control of partial domain features (see Section 2.2.3). The reason for this is that the features pertaining to partials are already defined in terms of the parameters of additive synthesis, namely the amplitude and frequency of each partial. Fourier analysis and resynthesis by means of an oscillator bank of sinusoids with time-varying amplitude and frequency is a basic model with wide-ranging uses in computer music, most notably in transformations such as pitch transposition and time compression or stretching. Extensions have been proposed such as Spectral Modelling Synthesis (SMS), wherein the sinusoidal part is complemented with a noise residual, which is found by subtracting the harmonic part from the original signal (Serra and Smith, 1990).

In SMS, first the spectral peaks are found and modelled with the tracking phase vocoder (Beauchamp, 2007). This forms the deterministic part of the signal. When the deterministic component is subtracted from the original signal, the residual, stochastic component remains. Hence, the SMS representation allows for the separate control and modification of partial domain and noise-related features. A further refinement to spectral modeling is Transient Modelling Synthesis (TMS), where the noise residual from SMS is further decomposed into transients and slowly varying noise (Verma et al., 1997).

Another development in signal representations are the so called atomic or dictionary-based models, in which signals are analysed using an overcomplete set of basis functions (Goodwin, 1998; Sturm et al., 2009). With the short-time Fourier transform, there is always a unique way to decompose a signal as the sum of harmonic series of complex exponentials (or sines and cosines). Overcomplete representations may be formed by taking an already complete set of basis functions and adding more basis functions to it. For example, one may include a (complete) set of Gaussian complex exponentials, and add chirps and wavelets to it. Consequently, there are more than one way to decompose the signal. This non-uniqueness of representation is exploited in applications such as efficient compression by sparse representations. Another application is to overcome the time-frequency limitations of traditional sonograms and accurately resolve both short clicks and long stable sinusoids. This has been achieved with the so called *wivigram*, which applies the Wigner-Ville distribution on a dictionary of sinusoids of varying frequency, phase and duration multiplied with Gaussian windows (Kling and Roads, 2004; Sturm et al., 2009). The signal decomposition is usually performed with some flavour of the

matching pursuit algorithm (Goodwin, 1998), where first the best matching basis function is found. This basis function is subtracted from the signal, which is then analysed again and the next best matching basis function is extracted, and so forth, until there is nothing left in the signal but, perhaps, a weak residual.

Audio feature extraction using sparse signal representations is a field that has only recently begun to be explored (Ravelli et al., 2008). An application to music genre classification was proposed by Henaff et al. (2011), using basis functions adaptively learned from the input signals. They used the constant-Q transform as the underlying signal representation and identified basis functions separately in each of four octaves, capturing different chordal intervals among other things.

On the sound synthesis side, overcomplete dictionaries and matching pursuit algorithms are related to granular synthesis. In effect, each basis function is a grain. Composite sounds may be assembled from grains translated in time and frequency, scaled, or modified according to any other parameter.

Spectral models such as SMS or TMS have their applications in sound transformations such as high quality time-stretching where transients are left untouched and morphing between source sounds. Overcomplete dictionaries may be used for sparse and perceptually meaningful representations of sound. All of these representations are more sophisticated and more complex to handle than the signal representation needed for most basic low-level feature extraction as reviewed in the previous chapter. Indeed, the FFT and a conversion to the amplitude spectrum is all that is needed for the set of low-level features described there, unless they are defined in the time domain.

Spectral modelling begins with an existing sound that is recorded and analysed, and whose time-varying parameters drive the synthesis algorithm. The manual specification of parameters in additive synthesis has been dismissed as too time-consuming although it is theoretically possible. In autonomous instruments where the synthesis parameters are algorithmically controlled additive synthesis may be worthwhile to try. However, next we will discuss additive synthesis more in detail primarily because of its close correspondence between synthesis parameters and partial domain feature extractors.

3.2 Additive synthesis from audio features

Spectral modelling with deterministic and stochastic parts, and possibly transients, are well researched methods to which nothing new will be added here. However, those features that SMS gives access to can be controlled in synthesis models without the analysis step by using synthesis by rule, which is the approach taken here, though more properly, it is a kind of analysis by synthesis. As described by Risset (1991, p. 18),

> a useful method to characterize a given type of timbre is *analysis by synthesis*, that is, building up a synthesis model, discarding aurally irrelevant features, and deciding from the subjective quality of the synthesis whether the simplified synthesis model retains the information essential to the identification of that timbre.

Knowing which features of the model are aurally irrelevant may however be difficult. Our approach will be to investigate parameterisations of additive synthesis models without using any source sound to be modelled. Nevertheless, the models for a sinus-to-noise continuum introduced below are approached in a typical analysis by synthesis manner.

Assume that we have analysed a sound with a sinusoidal model and extracted a number of partial domain features. Among the parameters we may have pitch, amplitude, inharmonicity, odd to even ratio, tristimulus and irregularity of the spectral fine structure, as well as more general descriptions of spectral envelope such as slope, roll-off, centroid, skewness and kurtosis (see Section 2.3). Now, it is possible to modify any number of these parameters and then resynthesise the transformed sound. Another option is to start from scratch; that is, using a synthesis model with these spectral attributes as control parameters, we may generate sounds by direct specification of all these control parameters. If this seems like a daunting task (indeed it can be), some sort of algorithmic control or data mapping from other areas than analysed sound signals may be applied. Let us first see what this synthesis model looks like.

For full flexibility, we do the synthesis by a bank of sinusoidal oscillators

$$x(n) = \sum_{k=1}^{N} a_k(n) \cos(\varphi_k(n)), \tag{3.1}$$

where $a_k(n)$ is the amplitude of the k:th partial at time n, $\varphi_k(0)$ are the initial phases, and the instantaneous phase of each partial

$$\varphi_k(n) = \varphi_k(n-1) + 2\pi f_k(n)/f_s \tag{3.2}$$

is determined by the time-varying frequencies $f_k(n)$. Variants of the sinusoidal oscillator with time-varying frequency f_n and sample rate f_s,

$$\begin{aligned} x_n &= \sin(\phi_n) \\ \phi_n &= \phi_{n-1} + \frac{2\pi}{f_s} f_n \end{aligned}$$

will be used often enough to merit the introduction of a shorthand notation that can be used when the frequency is variable and the phase does not need to be explicitly stated:

$$x_n = \mathrm{osc}(f_n).$$

To begin with, the amplitude, frequency and phase will be assumed to be constant over time. This restriction helps us to focus on a few timbral attributes pertaining to the spacing of partials and their relative amplitude. Then, noise will be added to this model. The control of timbral attributes has its limits; as we will see, when several partial domain features are specified simultaneously, conflicting demands may arise since they are not mutually independent.

3.2.1 Inharmonicity

Most of the nonlinear synthesis models that will be discussed in Section 3.3 may produce either harmonic or inharmonic spectra, and the same is obviously true for additive synthesis. Since a few of the nonlinear models are also used in feature-feedback systems, it may be illuminating to begin with a discussion of the nature of inharmonicity.

Given that a spectrum with partials that do not line up at harmonic frequencies is inharmonic, there are obviously several ways a spectrum can fail to be harmonic. Beauchamp (2007, 58 ff.) lists three types: sounds with nearly harmonic partials (e.g. piano), sounds with widely separated partials (vibraphone, chimes), and sounds with dense partials (cymbals, drums). The mode frequencies of plucked or struck strings take the form

$$f_k = k f_o \sqrt{1 + Bk^2}, \quad k = 1, 2, \ldots$$

for a fundamental (or more appropriately, lowest partial) f_o and inharmonicity index B taking small positive values. Bars, on the other hand, typically vibrate with modes at approximately

$$f_k = f_o(2k + 1)^2, \quad k = 1, 2, \ldots.$$

Frequencies for the modes of drum membranes can also be obtained from theoretical formulas, in this case as zeros of Bessel functions, although it should be remembered that spectra of real instruments always deviate from these simplified models.

If the partials f_k are assumed to be closely aligned to a harmonic spectrum, then inharmonicity may be defined as the deviation from a harmonic spectrum with fundamental frequency f_o. This raises some questions that are not always considered: Is the fundamental the lowest partial? What about spectra including odd partials only, or other highly irregular shapes?

Assuming that the spectrum contains only odd harmonics, it can easily be seen that a careless formulation of the algorithm for inharmonicity calculation will give absurd results: the second detected partial is already three times the fundamental (instead of two), and the difference grows for increasing partial numbers. In order to avoid unreasonable differences between an observed partial frequency and the corresponding perfect harmonic, the measured partial f_k should be matched to the nearest perfect harmonic kf_o. It will also be noticed that minor deviations in the estimated fundamental frequency will influence the inharmonicity measure considerably.

Rossignol (2000, p. 39) gives a formula for the calculation of inharmonicity for each partial separately. First, the fundamental is estimated from a small number L of the lowest partials,

$$f_o = \frac{1}{L} \sum_{k=1}^{L} \frac{f_k}{k}, \tag{3.3}$$

and an index of inharmonicity for each partial

$$H_k = \left| \frac{f_k - k f_o}{k f_o} \right|$$

is defined for $k = 1, \ldots, L$. Rossignol suggests using only three partials ($L = 3$), because decreasing amplitude in higher partials make frequency estimation less accurate. This may be sufficient in the context of audio segmentation by detection of discontinuities in the time function of certain descriptors, which is the problem Rossignol considers, but for a good overall estimate of a sound's inharmonicity it seems preferable to include a greater number of partials. Those partials that are resolved by the ear's frequency resolution are most important in determining perceived pitch of harmonic sounds, although higher, unresolved partials also contribute (Houtsma, 1995). For a fundamental frequency of 200-300 Hz, the number of resolved harmonics is about 11 or less. This gives an indication as to how many partials to include in the inharmonicity measure.

Then, to analyse global inharmonicity, the contributions of each partial considered is added up and weighed by their respective amplitude:

$$INH = \frac{2\sum_{k=1}^{L} a_k^2 \, |f_k - k f_o|}{f_o \sum_{k=1}^{L} a_k^2} \tag{3.4}$$

This is the form of inharmonicity calculation given in Peeters (2004), which should yield values in the range $[0, 1]$. It can be seen that perfectly harmonic spectra produce a zero value, whereas the maximum possible deviation from a harmonic spectrum would occur if the partials appeared exactly half-ways between the true harmonics. Then $|f_k - k f_o| = 1/2$, and the factor 2 in the numerator of eq. 3.4 ensures that the maximum value 1 is obtained. In practice, however, this line of reasoning does not hold, because shifting the partials of a harmonic tone up or down by $f_o/2$ results in a new harmonic spectrum. Moreover, the fundamental as estimated according to eq. 3.3 will tend to minimise the overall inharmonicity. Consequently, the highest practically attainable value of inharmonicity may be lower than 1.

3.2.2 Synthesis of inharmonic spectra

Additive synthesis models allow for the design of arbitrary spectral shapes and frequency relations of partials. Shifted spectra take the form

$$f_k = f_o(k + \delta), \quad k = 1, 2, 3, \ldots \tag{3.5}$$

For $\delta = -0.5$ the spectrum is harmonic with only odd partials and fundamental at $f_o/2$. A shifted spectrum with $\delta = -1$ is perfectly harmonic, but with a component at 0 Hz. Likewise, the spectrum is harmonic for $\delta = 1$, but now with a missing fundamental. For small shifts up or down, the perceived pitch of a shifted spectrum follows the direction of the shift. Similar effects are produced by detuning a single partial by a small amount. If the detuned partial deviates too much from its position in the harmonic spectrum, it segregates and can be heard in isolation (Bregman, 1990, p. 237 ff.). This, and similar phenomena, leads Bregman (1990, p. 242) to propose the name "partial pitch" for the pitch that is assigned to a single partial, and "global pitch" for the percept that belongs to a set of partials taken together. Taking such psychoacoustic effects into account would make inharmonicity measurement quite complicated.

Another variant is the family of stretched (or compressed) spectra

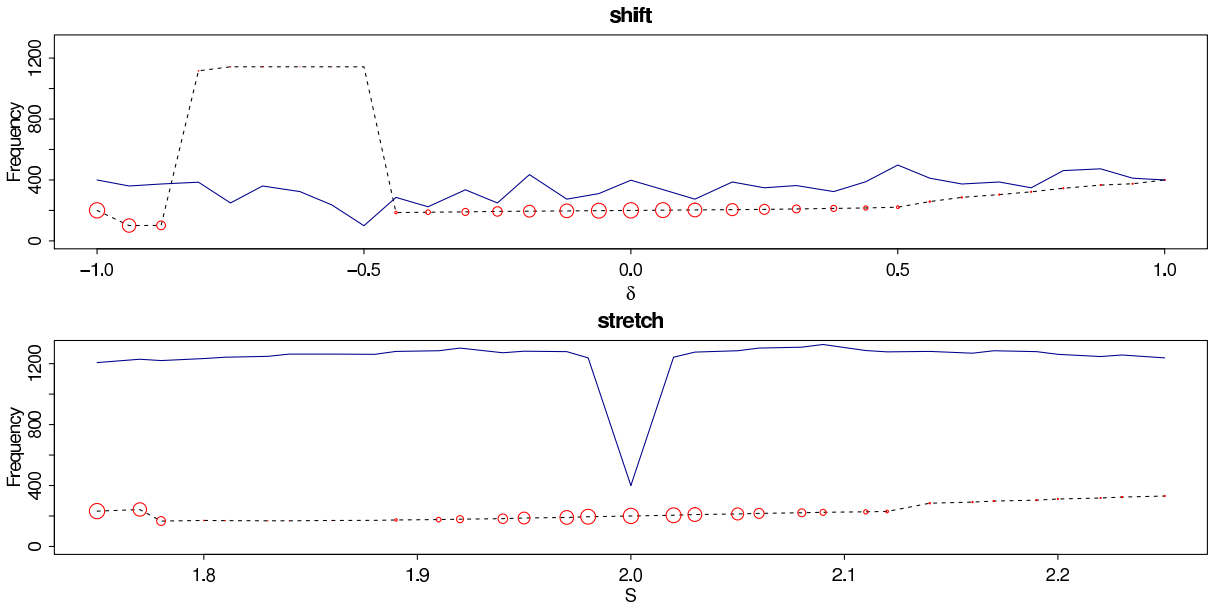

Figure 3.1: Estimated frequencies of inharmonic spectra. Top: shifted spectrum with $-1 < \delta < 1$; bottom: stretched spectrum with $1.75 < S < 2.25$. In both cases $f_o = 200$ Hz, and a lowpass filter with cutoff at 1.5 kHz has been applied to the signals before analysis. Solid blue lines: ZCR in units of Hz; dashed line with circles: estimated fundamental frequency using autocorrelation. The size of the circles indicate the voicing value; the bigger the circle, the better the frequency estimate.

$$f_k = f_o S^{\log_2 k}, \tag{3.6}$$

where the spectrum is harmonic if $S = 2$, compressed for $S < 2$, and stretched for $S > 2$. Sethares (2005) has discussed how to build scales with pseudo-octaves and other modified intervals that allow consonance in the modified consonant intervals when using spectra composed according to eq. 3.6. Two simultaneous tones built on a spectrum with $S = 2.08$ will sound very dissonant in a $1 : 2$ ratio, whereas the ratio $1 : 2.08$ sounds consonant. This is not so mysterious if the compound spectrum of both tones together is considered; if the spectral composition of partials does not match the interval between the fundamental frequencies, then there will be many closely spaced partials that add sensory dissonance.

Before we mention a few other ways to construct inharmonic spectra, let us take a look at the estimation of fundamental frequency from signals with shifted and stretched spectra (see Figure 3.1). If we generate signals using eqs. 3.5 and 3.6 with 25 partials of equal amplitude and apply a lowpass filter to the signal before analysis, we get the results shown in the figure. As a matter of fact, the autocorrelation based fundamental estimator fails badly even for perfectly harmonic spectra if all partials are of equal strength; under such conditions it returns an estimate close to the highest partial present in the signal. The shifted spectrum shows a peculiar jump in estimated fundamental frequency around $-0.8 < \delta < 0.5$ together with small voicing values (indicated by the size of circles). Voicing, being related to harmonicity (indeed, it is often called harmonicity), decreases

as the spectrum deviates from pure harmonic spacing. The predicted effect is clear in the bottom plot, where the voicing increases around $S = 2$, corresponding to the harmonic spectrum. As can be seen for both shifted and stretched spectra, the zero crossing rate estimator is independent from the fundamental frequency as estimated by the autocorrelation method.

Equidistant spacing on a logarithmic scale,

$$f_k = f_o 2^{kC/1200} \tag{3.7}$$

with a spacing C in cent, is another variant that typically produces inharmonic spectra. Octave spacing ($C = 1200$) is familiar from Shepard tones (Shepard, 1964), but any interval can be used. For particularly narrow intervals (on the order of 5 cents) and a sufficient number of partials, peculiar amplitude and frequency modulation effects occur (Holopainen, 2001).

Several nonlinear synthesis models may produce inharmonic spectra. Ordinary two-oscillator FM, it will be recalled, has partials at

$$f_k = f_c \pm k f_m, \quad k = 1, 2, 3, \ldots \tag{3.8}$$

where the harmonic index $H = f_c/f_m$ controls the type of spectrum. If H is a small integer rational number the spectrum will be harmonic, otherwise it will be inharmonic. Subharmonic sequences

$$f_k = f_o/k \tag{3.9}$$

may occur in chaotic systems with a subharmonic route to chaos, which will be discussed more in Chapter 4. In the period doubling cascade, which is the most familiar special case of subharmonics in chaotic systems, k in (3.9) is restricted to powers of two.

A counterintuitive fact is that the dissonance curves of a harmonic spectrum and its mirrored subharmonic spectrum are identical (Sethares, 2005, pp. 124-125). Since these spectral types are perceptually clearly distinct, one would suspect that Sethares' dissonance model is oversimplified.

As can be seen already from this sample of spectral types, there are several qualitatively different ways to produce inharmonic spectra. In addition to the assortment of spectra reviewed here, there are all kinds of irregular spectral compositions such as those found in bells and metal percussion instruments. The analysis with eq. 3.4 by itself cannot distinguish these types from one another. Inharmonicity as a graded perceptual attribute has a meaning only when is is possible to compare the spectral components to those of a perfectly harmonic spectrum. When there is any ambiguity as to the fundamental frequency or what partial numbers to assign to the frequency components, the concept of inharmonicity becomes hard to define other than as an all-or-nothing distinction.

The perceptual relation between inharmonic sounds and pitch is complicated. As Schneider (2000) has pointed out, church bells and carillons are tuned to scales and are used to play tunes despite the fact that each single bell on its own gives rise to an ambiguous pitch percept. Bells that are assigned higher scale steps can have a lower centroid than bells that have lower pitch, which may be a source of confusion in pitch

perception. Similar ambiguous pitch percepts do arise in the artificially constructed inharmonic spectra discussed here.

3.2.3 Partial domain features

Let us now turn to the descriptors of the spectral envelope and relative amplitude of partials. An interesting question is to what degree these are independent, or conversely, how correlated are the various features?

Tristimulus are the three components

$$T_1 = \frac{a_1}{A} \tag{3.10}$$

$$T_2 = \frac{1}{A}\sum_{k=2}^{4} a_k \tag{3.11}$$

$$T_3 = \frac{1}{A}\sum_{k=5}^{N} a_k \tag{3.12}$$

based on the relative strength of the partial's amplitudes a_k as given in (3.1), with $A = \sum_{k=1}^{N} a_k$. When two of the tristimuli are known, the third can be found since they sum to 1. Therefore, they are commonly plotted in a 2-D plot restricted to a triangular area. Originally, the tristimuli were calculated by Pollard and Jansson (1982) on the basis of a psychoacoustically motivated loudness model and used to plot the initial transients of various instruments. Here, as noted, we use the amplitude spectrum instead of loudness.

Odd to even ratio is another often used partial domain feature (Peeters, 2004). Clarinet tones are known to have a strong share of odd harmonics, which makes the odd to even ratio high. The formula, as calculated from the partial's power spectrum is:

$$OER = \frac{\sum_{k=0}^{N} a_{2k+1}^2}{\sum_{k=1}^{N} a_{2k}^2}. \tag{3.13}$$

A maximal odd to even ratio occurs in case the spectrum only consists of odd harmonics, whereas a minimal ratio can be observed if there are very weak odd harmonics—removing all odd order harmonics clearly has the effect of transposing the fundamental one octave up. It can be seen that at least the second tristimulus may be affected by the odd to even ratio, and possibly the other two as well.

The centroid and spectral roll-off can both be computed from the partials instead of a complete amplitude spectrum. A high centroid is intuitively seen to be positively correlated with T_3, whereas low centroids should be correlated with the first and perhaps the second tristimulus, provided the tones have the same pitch.

The spectral irregularity is the sum of differences between three adjacent partials (cf. eq. 2.13 on page 67); as such, its estimation can be accomplished in several ways. It is obviously correlated with the odd to even ratio. Thus, a spectrum with only odd partials yields high spectral irregularity, but the converse need not be true. Consider as an example a spectrum with every third partial removed, such that the partials have

amplitudes $1, 1, 0, 1, 1, 0, \ldots$; then the spectral irregularity is high, but the odd to even ratio may approach 0 dB.

From a synthesis perspective, these attributes are easily controlled taken one or a few at a time, but not so if several of them should be combined at once. Another application of tristimulus and centroid features to sound synthesis was considered by Sølvi Ystad (1998), who used waveshaping to generate the deterministic part of a flute sound. Flute tones, like many other instruments, have a rise in centroid as the dynamics increase, though the partials change in quite specific ways that the centroid on its own cannot capture. Ystad showed that by matching the waveshaping function to tristimulus attributes instead, a better match for pianissimo and fortissimo notes could be achieved.

3.2.4 Combining attributes

In additive synthesis, any of the attributes pertaining to spectral shape and the balance between partials can be directly controlled. Now, we would like to control several of them at once. Since there is some degree of redundancy or overlap across different attributes, this imposes a restriction on the other attributes after the value of the first has been set. This is reminiscent of the situation in overcomplete signal descriptions or frames, where there are several possible decompositions of the same signal (see section 3.1.4 above). But in contrast to overcomplete representations, here the representation is not necessarily complete, albeit overlapping. A fundamental result from linear algebra tells us that it takes N basis vectors in $\mathbb{R}^N$ to represent any point in $\mathbb{R}^N$. Thus, it takes N linearly independent attributes to describe all possible combinations of the amplitudes of N partials. However, the point of using a small number of attributes to describe a larger set of partials is precisely to attain a more compact description, and hopefully one that is more perceptually relevant than, say, an individual control of each partial's amplitude. This approach is similar to Group Additive Synthesis (Kleczkowski, 1989), which is an attempt to simplify additive synthesis by grouping partials that evolve in a similar manner with respect to amplitude and frequency.

It will be instructive to consider a simple synthesis model involving the three tristimulus attributes, odd to even ratio, and spectral slope. To begin with, each attribute shall be considered separately. Let the slope factor $S\,(\approx 1)$ that determines the partials' amplitudes be defined as $a_k = S^k$. Under the assumption of a harmonic spectrum, this formulation gives a straight slope in dB per Hz. An odd to even ratio equal to 1 should correspond to perfectly balanced odd and even partials, a ratio of 2 would imply that odd partials are twice as strong, etc. Hence, an OER of 1 signifies a balanced spectrum where odd and even partials have equal weight. However, dB units will be used for OER, with 0 dB corresponding to the balanced case and negative values to more prominent even partials. The tristimuli can be specified by two arbitrary numbers, since the third will be determined by the first two. For the sake of consistency, the tristimuli will also take units of dB. As we leave absolute amplitude out of consideration for the moment, the partials will have to be normalised by summing up the particular values that result from specific parameter settings.

Assume that tristimulus is calculated first, followed by odd to even ratio and lastly the slope is imposed on the partial's amplitudes. First, the user-supplied dB values are

converted to linear amplitude, and a vector of N partial amplitudes a_k are initialized to 1. Then the first, second to fourth, and remaining partial's amplitudes are assigned by multiplying the partial amplitudes a_k with the specified tristimuli values. Next, in order to impose the desired odd to even ratio, one multiplies the subset of even partials a_{2k} with some constant R. Finally, a slope S is imposed on all partials such that $A_k = S^k a_k$ after which they are normalised so as to sum to 1. Of course, one may obtain different results by performing these operations in a different order.

Now, the success of the model may be assessed by comparing the actually achieved feature values to those that were specified. For instance, setting a relatively high value of T_3 in combination with a steep spectral slope ($S \ll 1$) yields two conflicting demands. The tristimuli and odd to even ratio follow from eqs. 3.10-3.12 and eq. 3.13 respectively, and the estimated slope coefficient can be found by linear regression of amplitude in dB versus partial number. Figure 3.2 illustrates that the tristimuli cannot be separately controlled when specifying the odd to even ratio. The tristimuli were specified each to have zero dB, which means that the two curves should have been flat, had there been no mutual influence. The spectral slope, however, remains unchanged as the OER varies. Already this simple example should make it clear how optimistic it would be to assume that one can set up straightforward correspondences between synthesis parameters and resulting feature values so as to achieve the specified blend of partial domain features. This difficulty will only be compounded when the feature extractors are used inside feature-feedback systems.

Nevertheless, this synthesis model may be used to assess the relative perceptual salience and character of the chosen attributes. As would be expected, the spectral slope behaves similar to a lowpass filter for $S < 1$ and like a highpass filter for $S > 1$. Since this synthesis model produces the same waveform regardless of fundamental frequency, the filter analogy applies only as long as the fundamental frequency remains constant. Still using the filter analogy, the third tristimulus may be compared to a shelf filter.

The odd to even ratio requires rather high values before it becomes clearly audible. At +24 dB and other attributes set to neutral values, it definitely sounds hollow like a square wave, although slightly lower ratios may begin to evoke this impression (these observations are made on the basis of informal listening tests). For negative odd to even ratios, at a certain point the pitch an octave above the fundamental frequency becomes more prominent than the actual fundamental, and the perceived pitch may become ambiguous or jump up an octave.

As noted, once the attributes are set, the waveform is fixed. Hence, this synthesis model is practical for wavetable synthesis. Using separate wavetables for each of the tristimuli and for odd and even partials, a direct control of the features becomes possible. One of its limitations is the invariability of the spectral shape as the fundamental varies; in other words, it does not provide the specification of formants or a general spectral shape. This lack is easily amended, for instance with the addition of filters or by a technique that will be introduced in the next section. Synthesis with wavetables is also restricted to harmonic spectra, unless several wavetables be used at once (So and Horner, 2004). But at some point, as increasing numbers of wavetables are introduced, the benefits of using wavetable synthesis are outweighed by the complications and additive synthesis using an oscillator bank becomes the preferable alternative.

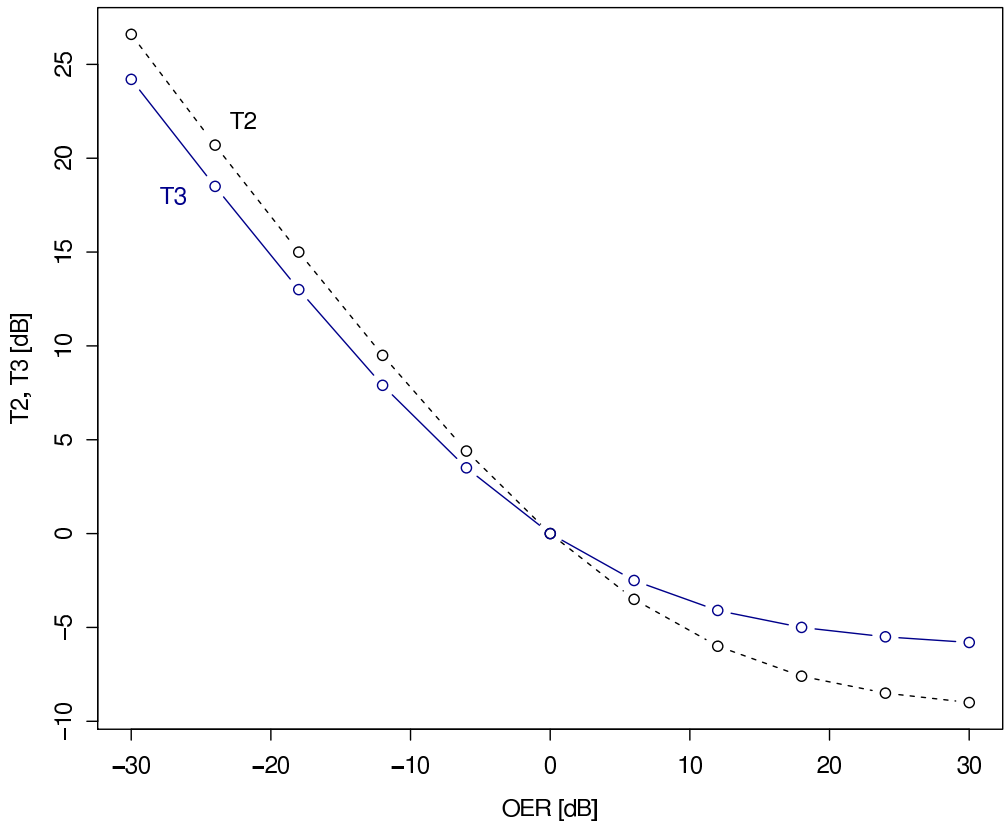

Figure 3.2: The resulting tristimuli T_2 (dashed black line) and T_3 (solid blue line) normalised so that $T_1 = 0$ dB are shown as a function of the specified odd to even ratio.

3.2.5 From sinusoids to noise

A general musical problem is how to produce convincing continuous transformations or morphings from one sonic character to another. For transformations in the continuum between a pure sinusoid and white noise, there are several solutions. The most crude technique of crossfading the noise and the sinusoid does not produce a convincing percept of the transformation of a single coherent sound source; rather it is heard as the mixing process it really is. Starting from white noise, successively narrower bands may be shaped with bandpass filters with increasing Q-factor. The limit of a pure sinusoid is only almost attainable this way, since slight irregularities in the amplitude will persist in the filtered noise.

A sinusoid with amplitude modulated noise produces *shimmer*, or random fluctuations of amplitude. Likewise, frequency modulation with noise produces *jitter*. AM and FM with variable amounts of noise can be combined into a single model

$$x_n = A_n \mathrm{osc}(f_n) \tag{3.14}$$

where

$$A_n = w\xi_n + (1 - w) \tag{3.15}$$

is the instantaneous amplitude, and

$$f_n = F_0(1 + Iw\zeta_n) \tag{3.16}$$

is the instantaneous frequency, and $\xi,\ \zeta \in [-1, 1]$ are two independent sources of uniformly distributed noise. The weight $w \in [0, 1]$ controls the balance between pure sinusoid ($w = 0$) and maximal noise modulation ($w = 1$). Furthermore, the FM index I in (3.16) is an important parameter. Setting it too low will not produce convincing results; then the pure sinusoid will be heard too clearly. On the contrary, setting I too high makes the white noise dominate over the transition as w varies between 0 and 1. We will use $I = f_s/20F_0$, which is a good compromise. The transition from pure sinusoid to white noise as it is measured by two feature extractors is shown in Figure 3.3. The voicing decreases quite smoothly, while the spectral entropy increases rapidly as soon as even a small amount of noisiness enters. Before the estimation of spectral entropy, the signal is windowed. The choice of window turns out to have a significant impact on the spectral entropy level for single sinusoids, which is understandable since the window introduces spectral sidelobes that affect the entropy measure quite markedly. In Figure 3.3 a von Hann window was used. Theoretically, a single sinusoid has zero spectral entropy.

Clearly, both the voicing and spectral entropy features reflect the sinusoid-to-noise transition very well, although the shapes of the curves are dramatically different (imagine one of the curves flipped upside down for direct comparison). If there were data from perceptual studies of this transition, it might follow a third curve—would it look more like the spectral entropy curve, or like the voicing curve? Apparently, small changes are easier to detect near the pure sinusoid extreme than near the white noise side, which indicates that the perceived noisiness might be captured better with spectral entropy than with voicing. This observation is only based on informal listening tests, though. When listening to a slow transition going from noise to sinusoid, before the pure sinusoid emerges the sound appears to segregate into a noise floor of diminishing amplitude and a gradually focusing sine tone.

Sinusoids and white noise are stimuli that have few perceptual dimensions—loudness for noise, pitch and loudness for the sinusoid. The sounds in-between are however much more malleable, since the path from a pure sinusoid to white noise may be traversed in so many ways. In the above model (3.14), the mixing effect is heard as a feeble noise floor near the sinusoidal end of the continuum.

Spectral modeling synthesis, or the decomposition of signals into a deterministic sinusoidal part and a stochastic part of coloured noise, implies a decoupled representation of these two components. In other words, the noise is added to the deterministic part as a regular mixing of two signals. Shimmer and jitter, on the other hand, are caused by the modulation of sinusoids. Modulation techniques belong to nonlinear synthesis models, and will be considered below.

When noise is used as a source to be mixed in together with sinusoidal components, the choice of its amplitude distribution may not matter as long as it is not too exotic. Uniform and Gaussian noise tend to sound undistinguishable, apart from any potential difference in loudness. A true Cauchy distribution however,

$$p(x) = \frac{\tau}{\pi(\tau^2 + x^2)}, \quad -\infty < x < \infty \tag{3.17}$$

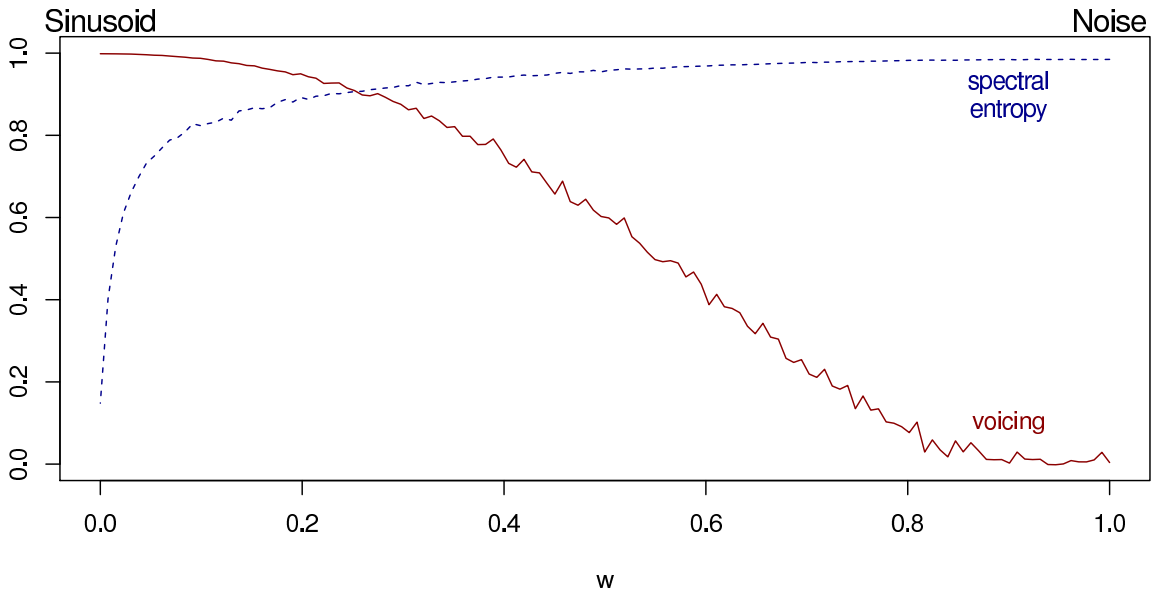

Figure 3.3: Sinusoid-to-noise transition with a steep rise in spectral entropy and more gradual change of voicing.

with spread τ, is not very practical as an audio signal unless its infinite amplitude range is limited. Unlike uniform or Gaussian noise, Cauchy noise is distinguishable by its intermittent rare spikes over a feeble noise floor. For an overview of these and other stochastic distributions that have been popular in computer music, as well as recipes for generating them, see Lorrain (1980).

Fractal distributions arise in chaotic iterated maps (Tél and Gruiz, 2006, see also ch. 4). These are characterised by highly irregular and inhomogeneous shapes, defined on a support (the attractor) which itself is a fractal, but with an added irregularity along this support. Again, the spectrum may be white, and perceptually indistinguishable from other, smooth random distributions.

The choice of distribution is however most important when the stochastic signal is used to modulate another signal. Xenakis (1992, ch. IX) considered various random distributions for waveform generation. His strategy was apparently to use one distribution for amplitude values and another (or possibly the same one) to determine the time points where new amplitude values were to be generated. In between, the function was held constant. This technique lends itself to the production of certain kinds of crackling or bursting noises.

This review of methods related to additive synthesis and additional noise modulation has shown how features can be used to control the balance of partials, or the balance between stable and noisy tones, thereby shaping tone colour. The synthesis models considered so far may be used as signal generators in autonomous instruments, although here they have only served as convenient illustrations of the relations between audio features and synthesis parameters. Nonlinear synthesis models offer a forceful control of global timbral qualities by a single parameter, something that makes them attractive for use in autonomous instruments.

3.3 Nonlinear models

Most of the models discussed in this section have their roots in early days of electronic music and exist in both analogue and digital forms. They include variants of waveshaping, frequency and ring modulation, phase distortion and discrete summation formulas. FM models with feedback will be further discussed in Chapter 4. Waveshaping appears as an indispensable part of many synthesis techniques. Indeed, as Marc Le Brun (1979) has demonstrated, waveshaping combined with ring modulation is able to simulate other nonlinear techniques including FM.

One might think that nonlinear models should be less intuitive than, for instance, additive synthesis, in the way control parameters relate to perceived sonic qualities. However, if the comparison is stated on the premise that additive synthesis parameters are not to be derived from the analysis of any sound, then the opposite would probably be true. Some nonlinear techniques such as feedback FM offer a simple and powerful control over spectral richness by a single parameter, the modulation index. Such global control of several partials at once is typical of nonlinear synthesis models. On the other hand, while sinusoidal models can recreate and modify any sound by well known analysis techniques, nonlinear models are more restricted and usually do not come with handy analysis techniques.

3.3.1 Basic modulation

Modulation is the dynamic change of a signal. Usually modulation is periodic, but it may also be irregular and aperiodic. Typical modulation rates lie in the sub-audio range, as is often seen in synthesisers that provide a specially devoted low frequency oscillator (LFO) to be used for such purposes, although higher modulation rates may also be useful. If the rate is in the audio range, the modulation will alter the timbre by introducing sidebands.

Modulation techniques that influence amplitude, frequency or spectrum have been common in electroacoustic music since the early years. Amplitude modulation (AM) takes a non-negative modulation signal, while ring modulation (RM) uses a bipolar (alternating positive and negative) signal. In frequency modulation (FM), the frequency of a carrier oscillator is influenced by a modulator signal. For audio rate modulators and carriers, this is the familiar synthesis technique introduced by John Chowning (1973). Occasionally the term FM is used also for sub-audio modulation, which is perceived as a vibrato.

AM and RM have a characteristic effect on the spectrum since the multiplication of time domain signals corresponds to the convolution of their spectrum. If the modulator is a sinusoid $x(t) = \sin(2\pi f t)$, the resulting spectrum will consist of two copies (sidebands) of the original spectrum, one shifted f Hz down (and mirrored), and one shifted f Hz up. Single Side Band modulation (SSB) is timbrally slightly reminiscent of RM, but as its name implies, it has only one sideband instead of two (provided the modulator is a single sinusoid). Hence, it produces exactly as many partials as there were in the original sound, and the effect is simply to shift every partials' frequency by the same amount.

SSB can be implemented with the Hilbert transform as follows. Let $z(t) = x(t) + iy(t)$ be the Hilbert transform pair of the input signal and let $w(t) = e^{i\omega t} = \cos(\omega t) + i \sin(\omega t)$ be the complex modulation signal. Then, the upper sideband u and lower sideband v are

obtained by taking the real parts of the multiplied signals

$$\begin{cases} u &= \mathrm{Re}(z\bar{w}) \\ v &= \mathrm{Re}(zw) \end{cases} \tag{3.18}$$

where $\bar{w}$ denotes the complex conjugate of w.

Demodulation is to undo the effect of modulation. Zölzer (2002, ch. 4) gives some interesting examples where single sideband modulation is used. First, the spectrum is shifted in one direction, then some filtering or vibrato is applied to the signal and finally the spectrum is shifted back to its original position. Inharmonic comb filtering can be realised this way. Practically any other sound synthesis parameter susceptible to temporal variation may be modulated. Common examples include filters with sweepable centre frequencies or cut-off and variable waveshapes as in pulse width modulation. Modulation often plays an important role in feature-feedback systems, at least if parameter update occurs at the audio sample rate or at a rapid control rate.

Although modulation often involve sinusoidal signals, noise sources may also be used as the modulator. Kristoffer Jensen (2005) proposed a general model which allows the independent control of both shimmer and jitter. In this model lowpass filtered Gaussian noise with variable bandwidth is used. The noise intensity and correlation between partials can be explicitly controlled. Shimmer is introduced by adding the noise to a constant amplitude value, which produces broad sidebands around the partial. Jitter introduces effects that can be understood by comparison with standard FM, where the carrier is modulated by another sinusoid. In that case, an infinite number of sidebands result, with amplitudes determined by Bessel functions. If instead the modulator is lowpass filtered noise, the sidebands' frequencies will be given by probability density functions. The modulation index then controls the jitter bandwidth.

The full model, with independent control of noise modulation in each of the K partials, is

$$x_n = \sum_{k=1}^{K} a_k(1 + \sigma_k^s s_k[n]) \sin(\theta_k[n]) \tag{3.19}$$

where σ_k^s is the strength of shimmer for partial k, and s_k is the lowpass filtered noise, which stems from separate stochastic processes for each partial; and

$$\theta_k[n] = \theta_k[n-1] + 2\pi f_k(1 + \sigma_k^j j_k[n]) \tag{3.20}$$

where σ_k^j is the strength of jitter, j_k is lowpass filtered noise for each partial, and f_k are the frequencies of the partials.

Furthermore, a powerful control parameter $c \in [0, 1]$ is added to this model, which provides an adjustment of the correlation between the partials. For shimmer and jitter, this becomes

$$s_k = (1 - c)r_0 + cr_k \tag{3.21}$$

$$j_k = (1 - c)r_0 + cr_k \tag{3.22}$$

with a common noise source r_0 and independent noise r_k for each partial. The parameter c then controls the balance between perfectly correlated and totally independent partials.

By controlling the parameters related to jitter and shimmer, a wide range of sounds can be produced. Jensen (2005) qualifies some of these sounds as "splashing", "almost screaming", "crackling", and "windy". Boiling or gurgling noises may be other ways to describe some of the sounds.

The use of noise in synthesis models is quite well understood. It can be a valuable addition to additive synthesis as well as to any nonlinear synthesis model. Instead of a white noise source, a chaotic map may be used to modulate an oscillator, which is something we will consider again in Section 7.2.2.

3.3.2 Feedback AM

Since feedback systems will play an important role in later chapters, let us consider some cases of feedback in nonlinear synthesis models. Feedback amplitude modulation is a simple and economical, albeit little known synthesis model. Five variations on this model are suggested by Lazzarini et al. (2009a), only two of which will be discussed here. The output from a sinusoidal oscillator is delayed by one or several samples and used to modulate the amplitude of the current sample. The amount of feedback is controlled by the parameter β, and the delay is D samples:

$$y_n = \text{osc}(f)(1 + \beta y_{n-D}) \tag{3.23}$$

Feedback ring modulation is obtained by removing the constant 1 from (3.23), but this is not very practical, since whenever the signal reaches zero, it would get stuck there. In dynamic systems, usually involving several coupled oscillators, this state of ceased oscillations is known as *oscillation death*, or *amplitude death* if the output amplitude becomes zero. The problem of avoiding oscillation death is something to be aware of in feedback systems, as one may unwittingly set up the system in such a way that its occurrence cannot be ruled out. In Section 6.4, we will deliberately construct an oscillator that suffers from amplitude death.

In the above formulation of feedback AM (3.23), the amplitude needs to be normalised. A suitable gain factor is $G = 1/(A+B\beta)$, for $A \approx 2$ and $B \approx 10$. The amount of feedback for which this system is stable is not obvious. For $\beta < 2$ it may be stable for a unit delay, but this also depends on frequency. For $\beta > 1$ the above gain factor is insufficient, but an adaptive gain control in the form of a compressor will guarantee that clipping is avoided. On the other hand, some of the character of this simple model is lost then. Harmonically spaced formant regions occur at multiples of f_s/D Hz, but these formants are suppressed if a compressor is inserted.

Another variant, capable of both harmonic and inharmonic spectra, includes ring modulation

$$y_n = \text{osc}(f_m)\text{osc}(f_c)(1 + \beta y_{n-1}) \tag{3.24}$$

with $f_m = Mf_c$. For integer values of M, the effect is to rebalance the strength of partials, for instance, $M = 2$ weakens the second partial. For $M = N + 1/2$ with integer $N \geq 0$,

only odd harmonics are retained. Note that transitions between differently weighted harmonic spectra (when M is dynamically changed) will traverse inharmonic spectra.

With two oscillators, cross-coupled AM and FM becomes possible (Valsamakis and Miranda, 2005). In these cases, the signal from the first oscillator modulates the amplitude of the second, which in turn modulates the amplitude of the first; similarly with FM. Hybrid cross-coupled modulation provides a third alternative, where one oscillator is amplitude modulated while the other is frequency modulated. In Section 6.2, a single feedback oscillator with cross-coupled AM and FM will be studied in detail, but it differs from directly cross-coupled oscillators by the use of frequency and amplitude feature extractors.

3.3.3 Extended FM models

Fast modulation is to be expected in many feature-feedback systems, where sliding feature extractors are mapped directly to synthesis parameters. In particular, if the frequency of an oscillator is under the influence of such feedback mapping, then FM of some kind will result. FM has been used for audio synthesis in many varieties (Roads, 1996; Horner, 2007a). We will consider two novel and experimental techniques; first, an adaptive version of exponential FM, where the resultant sideband frequencies are adjusted; and second, a combination of FM with waveset omission.

Several variations of standard two oscillator FM are possible. The standard FM formula can be written as

$$\begin{aligned} x[n] &= A\sin(\theta_c[n] + I\sin(\theta_m[n])) \\ \theta_c[n] &= \theta_c[n-1] + \omega_c \\ \theta_m[n] &= \theta_m[n-1] + \omega_m \end{aligned} \tag{3.25}$$

with carrier $\omega_c = 2\pi f_c/f_s$, modulator $\omega_m = 2\pi f_m/f_s$, amplitude A and modulation index I.

Roads (1996) describes exponential FM, which is implemented in some analogue synthesisers where the sinusoidal modulation is carried out over logarithmic rather than linear frequency units, as is customary. Hence, the instantaneous frequency is

$$\omega[n] = \omega_c B^{\cos(\omega_m n)} \tag{3.26}$$

so, with modulation index $\beta = \ln B$,

$$x[n] = \sin(\theta_c[n] \cdot \exp(\beta\cos(\theta_m[n]))) \tag{3.27}$$

where the phases θ are updated as in (3.25). This model produces other sideband frequencies than standard FM; in particular, the frequencies change with the modulation index. Some partials' frequency rise, whilst others fall. This irregular behaviour may be compensated for to some degree. Doing so can be conceived of as adapting the synthesis parameters to the spectrum of the output signal.

If we think of the modulation as a vibrato, it will give the modulated carrier (3.26) an average frequency $\hat{m}f_c$ which will increase with the modulation index (assuming $\beta \geq 0$). This mean frequency can be estimated by the numerical evaluation of

$$\hat{m} = \frac{1}{2\pi} \int_0^{2\pi} e^{\beta \cos(t)} dt \tag{3.28}$$

and then, from this estimate the carrier is adjusted to $f_c' = f_c/\hat{m}$. Whenever the modulation index β changes, the integral (3.28) has to be evaluated again. Then, if one wants to keep the $c : m$ ratio unchanged, the modulation frequency will also need to be adjusted in the same way, giving $f_m' = f_m/\hat{m}$. Now the resulting adjusted exponential FM model produces one stable partial at the carrier frequency, but still the other partials change with the modulation index. This time, most of the changing partials glissando downwards with increasing modulation, in combination with an increase in the number of prominent partials (see Figure 3.4).

Exponential FM in its original version as well as in its modified form introduce sidebands whose frequency vary as a function of the modulation index. This makes both synthesis models rather restricted, with usage probably best limited to special effects unless the modulation index is fixed at some constant value. However, the adjustment scheme proposed above may be generalised to other situations. In this case an analytic solution was found, but this is often too complicated. Then, feature extractors may replace the analytical solution or numerical estimation of synthesis parameters that need to be adjusted. The pitch control of nonlinear oscillators is one such application that will be introduced in the next chapter.

Wishart (1994) introduced a number of highly original audio effects, some of which were briefly mentioned above in Section 3.1.3. In particular, some of the effects that take the waveset as the atomic unit of sound are interesting to apply as a part of FM synthesis. Recall that the waveset is the smallest segment of the waveform between two consecutive zero crossings going in the same direction. In waveset omission, for every M out of N wavesets (for $M < N$) all samples are set to zero, which produces a granular effect.

When waveset omission is applied to the modulator in standard FM (3.25), the output signal will switch between an unmodulated carrier sinusoid and the fully modulated FM sound. Waveset omission of a sinusoid produces complex tones with fundamental frequency at the inverse of the total period. Given the modulator frequency f_m and an omission of M out of N wavesets, the resulting modulator will thus have its fundamental at $F_o = f_m/N$ Hz. This frequency can be enhanced by filtering the modulator signal with a bandpass filter that is also tuned to the frequency F_o. The result is more buzzing than regular FM and also more timbrally coherent as the modulation index is varied. To some approximation, this model is similar to regular FM with a complex modulator. The remaining differences are due to the bandpass filter, whose temporally extended ringing makes this a system with memory. Feedback variants of FM with waveset omission are also interesting; in particular, they begin to approach feature-feedback systems insofar as the waveset omission is a signal adaptive operation. Further examples of FM combined with filters are given in the next chapter (Section 4.3.4).

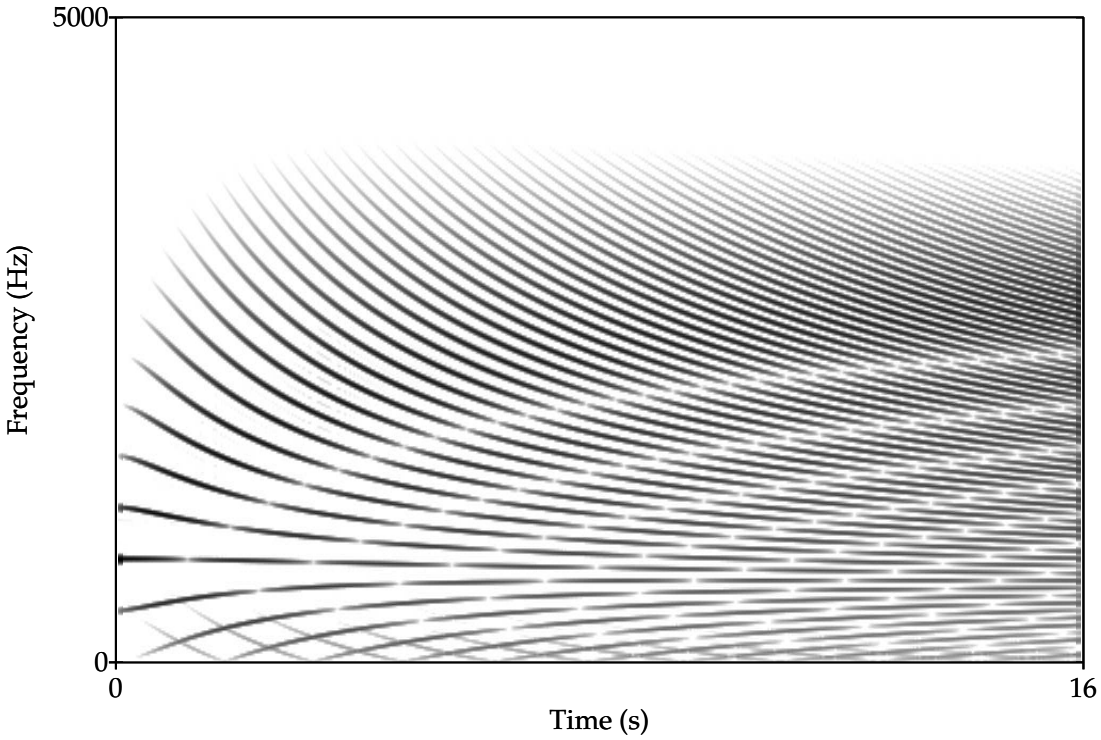

Figure 3.4: Exponential FM with adjustment, B increases from 1 to 26, carrier 800 Hz, modulator 400 Hz.

3.3.4 Wave terrain synthesis

Synthesis as a function of two variables (Mitsuhashi, 1982), better known as *wave terrain synthesis*, is a technique that at least merits from its intuitive geometrical interpretation. There is a path $\mathbf{c}(t) = (x(t), y(t))$ which travels through a "landscape" with varying height at different (x, y) coordinates. This terrain is defined by another function $f(x, y)$. The output signal is obtained from the composition $f \circ \mathbf{c}$. While the graphic image of a path circulating on a bumpy terrain is easy to grasp (as in Figure 3.5), it may be more revealing to consider this operation in algebraic terms. Assuming the two components of the path are each sinusoidal signals and the surface a low-degree polynomial, it will be easy to calculate the resulting spectrum.

Consider the wave terrain

$$z = f(x, y) = axy + b(x^2 - y^2) + c(x + y)^p \qquad (p \text{ odd}) \tag{3.29}$$

in which two sinusoids

$$\begin{aligned} x(t) &= A_1(t)\text{osc}(u) \\ y(t) &= A_2(t)\text{osc}(v) \end{aligned} \tag{3.30}$$

are the arguments of the function. An orbit with two incommensurate, slightly detuned frequencies trace out the wave terrain, as shown in Figure 3.5 which uses (3.29) for the

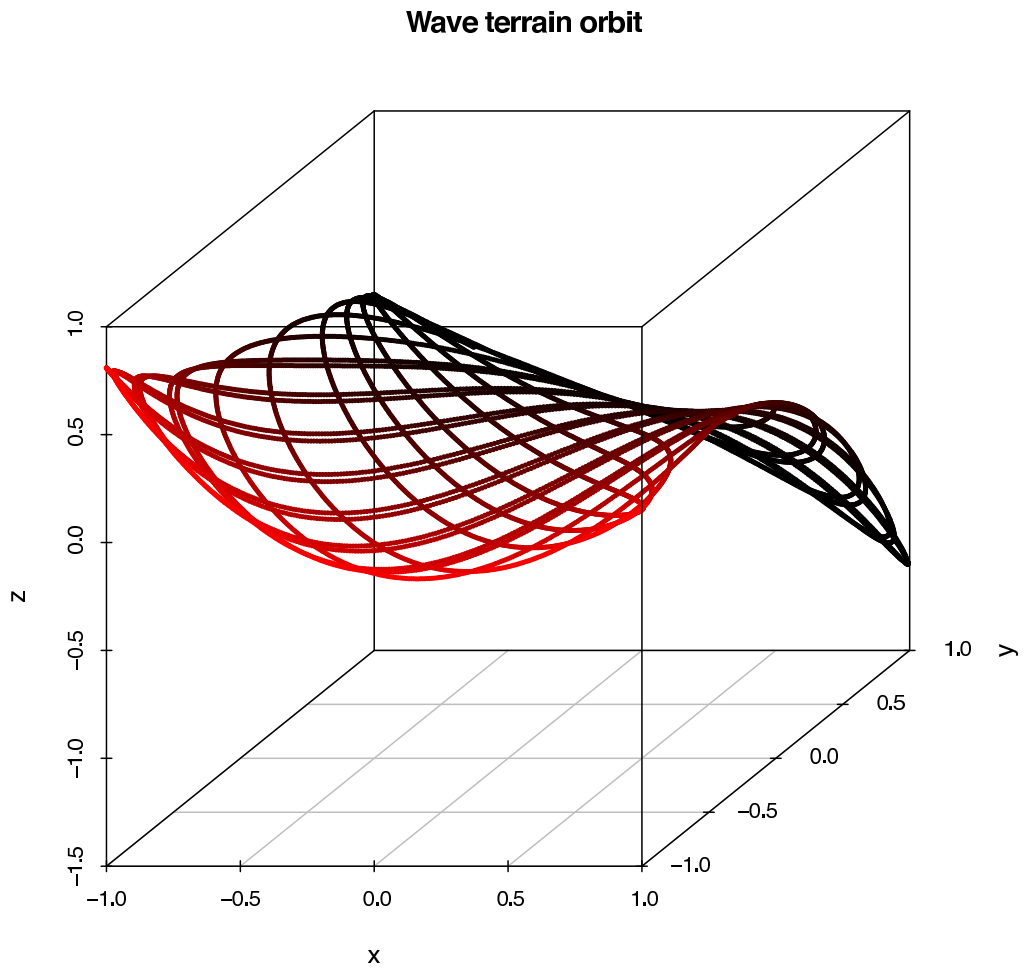

Figure 3.5: An orbit of two inharmonically related sinusoids traces out the wave terrain.

terrain. Had the oscillators been tuned to some simple ratio, the orbit would have been a closed loop that would not fill the terrain.

In (3.29), there is a cross-modulation term a, a saddle function component b, and an odd-degree waveshaping component c with cross-modulation. Now, the spectral shape will depend directly on the frequencies of the two sinusoids, while the amplitude relationships of its partials will be determined by the two amplitude variables of (3.30) and the coefficients in (3.29). Given the two frequencies u and v, the resulting spectrum can easily be derived. For simplicity, we will not consider relative amplitudes, but only the set of frequencies. The first term is simple ring modulation, which yields the frequencies $u+v$ and $u-v$. Next, there is standard waveshaping, where the squared terms contribute to frequencies $2u$ and $2v$, respectively (and a constant DC-offset if $A_1 \neq A_2$). Lastly, the sum of the two sinusoids raised to an odd power is given by the binomial expansion. For concreteness, take $p = 3$, and we have $(x+y)^3 = x^3 + x^2y + xy^2 + y^3$. This produces the partials u, v, $3u$, $3v$, $2u \pm v$, $2v \pm u$. Just as with the choice of harmonic ratio in FM, it can be seen that if u and v form simple ratios, the sound will be harmonic, and otherwise

inharmonic.

For richer spectra, the path components $x(t)$, $y(t)$ may be composed of several partials. It will be preferable to keep these bandlimited, so that the intermodulation and polynomial waveshaping in (3.29) will not introduce aliasing. Arbitrary paths may be designed, and, if they are periodic and contain a finite number of discontinuous jumps, they may be approximated by a number of Fourier coefficients, i.e., as sums of sinusoids in x and y. Likewise, smooth terrains of almost any shape may be approximated by a truncated two-dimensional Taylor expansion, which restricts the polynomial degree.

In ordinary waveshaping with a single sinusoid as input, spectral matching against a target sound is the usual design method. The solution is to use Chebyshev polynomials to determine each partial's strength. Whereas waveshaping of a single sinusoid is limited to harmonic spectra (unless the output is ring modulated, as proposed by Le Brun (1979)), the wave terrain technique includes inharmonic sounds as well. Presumably, it should be possible to match wave terrain synthesis to analysed sound spectra, but little if any research seems to have been carried out in that direction. Timbrally, the proposed wave terrain model is quite versatile, and with some effort, it might be capable of a crude simulation at least of certain woodwind tones or a piano.

Example 3.1. The spectrum can be designed by setting the ratio between the oscillator frequencies u and v in this wave terrain instrument. Here, first some harmonic ratios are used, then slightly detuned harmonic ratios, and lastly an inharmonic ratio. During each tone, the amplitude of both oscillators decreases exponentially.

The control parameters $(A_1, A_2, u, v, a, b, c)$ may all be time-varying. If the two frequencies u and v are identical, the model may be given an elegant formulation using a complex exponential oscillator and the variables $A, \omega, \alpha, \beta \in \mathbb{C}$:

$$\begin{aligned} z(t) &= Ae^{i\omega t} \\ w(t) &= \alpha z^2 + \beta z^3 \end{aligned}$$

Other terrain functions and orbits were originally studied by Mitsuhashi (1982). Later on, the scope of wave terrain techniques has been expanded by James (2005), who even suggested using time-variable terrains, possibly obtained from video images. In Chapter 6, we present an elaboration of the wave terrain model (eqs. 3.29–3.30) into a feature-feedback system with a large number of parameters.

3.3.5 Phase distortion

If a periodic waveshape is read with periodically variable speed, one gets phase distortion. For an introduction to the technique, as well as a novel take on phase distortion with allpass filters, see Lazzarini et al. (2009b).

If an oscillator is implemented by wavetable lookup, the indexing function would ordinarily be a straight line or a ramp function taken modulo the table size. Phase distortion uses nonlinear indexing functions to access the current position in the wavetable. The waveshape stored in the lookup table may be one period of a sine function, although

other waveshapes could be used. Then, the current index into the lookup table W of size N is given by a function $f : [0, 1) \rightarrow [0, N - 1]$, such that

$$x_n = W[f(\theta_n)] \tag{3.31}$$

and

$$\theta_n = \theta_{n-1} + \omega \,(\mathrm{mod} 1).$$

In practice, fractional indexing and an interpolating lookup table oscillator would be used.

Phase distortion then depends on the phase warping function, and whatever control one designs into it has to take the form of a parameter controlling the shape of that function. The immediate effect of departing from a linear function is to introduce harmonic overtones. Thus, phase distortion is suitable for use in instruments where partial domain (rather than spectral shape) attributes are involved. It should be noted that phase distortion is limited to harmonic tones. A common technique in synthesisers is to detune several oscillators slightly and mix their output to a single signal. Phase distortion is such a technique that works well with detuned oscillators. Slightly detuned partials cause slow beats that induce liveliness in otherwise static tones. Detuned oscillators is of course just as useful with any of the nonlinear techniques reviewed here, but in some other cases similarly detuned partials come for free when the parameters are set right.

Instead of the wavetable lookup implementation (3.31), which is primarily motivated by efficiency concerns, an explicit function call to some function w could be used, such that $x_n = w(f(\theta_n))$. This would be advantageous if the waveshape w needs to be changed dynamically. Amazingly, for harmonic tones, any distortion that may be accomplished by waveshaping may as well be performed by phase shaping (Timoney et al., 2010). Intuitively, this makes sense because waveshaping can be seen as stretching or squeezing a sinusoid in the vertical direction, whereas phase shaping involves similar deformations in the horizontal direction. For example, if the original waveshape is a sinusoid, then a triangular waveshape similar to the letter N may be obtained by appropriately speeding up and slowing down the phase increment. However, if further wiggles should be added to the waveshape, then the instantaneous frequency will sometimes need to be negative in order to change the sign of the waveform's slope.

In Chapter 7, we will rediscover phase distortion and lookup tables in an unfamiliar setting. There they will be used as parts of step sequencers instead of as oscillators (see Section 7.1.2).

3.3.6 Discrete summation formulas

A shifted harmonic spectrum as given by eq. 3.5 can also be obtained in, at least, two other ways than by additive synthesis. First, there are discrete summation formulas, introduced as a synthesis technique by Moorer (1976). Second, harmonic sounds can be shifted with the aid of the Hilbert transform, using eq. 3.18.

There exist a number of different discrete summation formulas, but their idea is to express trigonometric series in closed form. Among the formulas, one of the most useful for audio synthesis is

$$\sum_{k=0}^{\infty} a^k \sin(\theta + k\beta) = \frac{\sin\theta - a\sin(\theta - \beta)}{1 + a^2 - 2a\cos\beta}, \quad 0 < a < 1 \tag{3.32}$$

although a slightly more complicated formula exists for a finite sum. For synthesis, the right hand expression is used, but for interpretation of its meaning, the left hand side is consulted. The parameter a specifies a spectral slope, $\theta = 2\pi\frac{f_c}{f_s}n$ plays the role of a carrier, and $\beta = 2\pi\frac{f_m}{f_s}n$ acts as a modulator. With $\beta = 0$, this reduces to a sum of harmonic frequencies, but otherwise, the spectrum is shifted up or down for positive and negative β, respectively.

Example 3.2. Various carrier-to-modulator ratios produce harmonic or inharmonic spectra and sometimes beating, depending on the ratio. Here, the discrete summation formula is used with the modulation index ramped up and down in each of four tones using different ratios.

As Moorer (1976) notes, since the amplitude in (3.32) changes with a, it needs to be normalised somehow. The best thing, according to Moorer, would be to normalise according to the perceived loudness. At the time when this technique was introduced, this solution was seen as too costly in terms of computation. Today it is well within reach, but on the other hand, the appropriateness of loudness normalisation may be questioned. An inevitable property of most acoustic instruments is that as they are played louder, the spectral richness also increases. The same coupling comes for free in eq. 3.32. Still, it is necessary to apply some normalisation to ensure that the signal's amplitude is restricted. Although closed formulas for normalisation were also derived by Moorer, they need to take into account special cases where partials from negative frequencies may overlap with positive frequencies and interfere constructively or destructively. A lazy solution would be to insert a compressor in the synthesis algorithm.

The kind of inharmonic spectrum encountered in this synthesis model would be inappropriate to analyse using an inharmonicity feature such as (3.4) since it may be difficult to assign a fundamental frequency to the spectrum. Instead, the voicing (which is also called harmonicity for a reason) may be used to capture the variation between consonant and dissonant sounds that are produced as the modulator to carrier ratio $f_m : f_c$ is varied. This is illustrated in Figure 3.6 (right), where the voicing is high for integer ratios and for low modulation indices. As the modulation index increases, the curve becomes more irregular and tends to lower values of voicing, which may be caused by aliasing. Both the amplitude and the centroid increase dramatically as the modulation index approaches 1; the increasing centroid can be seen in the left part of the figure. A feature-feedback system using the discrete summation formula (3.32) will be described in Chapter 7 (Section 7.2.4).

3.3.7 The tremolo oscillator and Fourier series

Simple geometric waveforms such as sawtooth, triangular and rectangular waves were standard in analogue synthesisers and continue to be used in digital synthesis. None

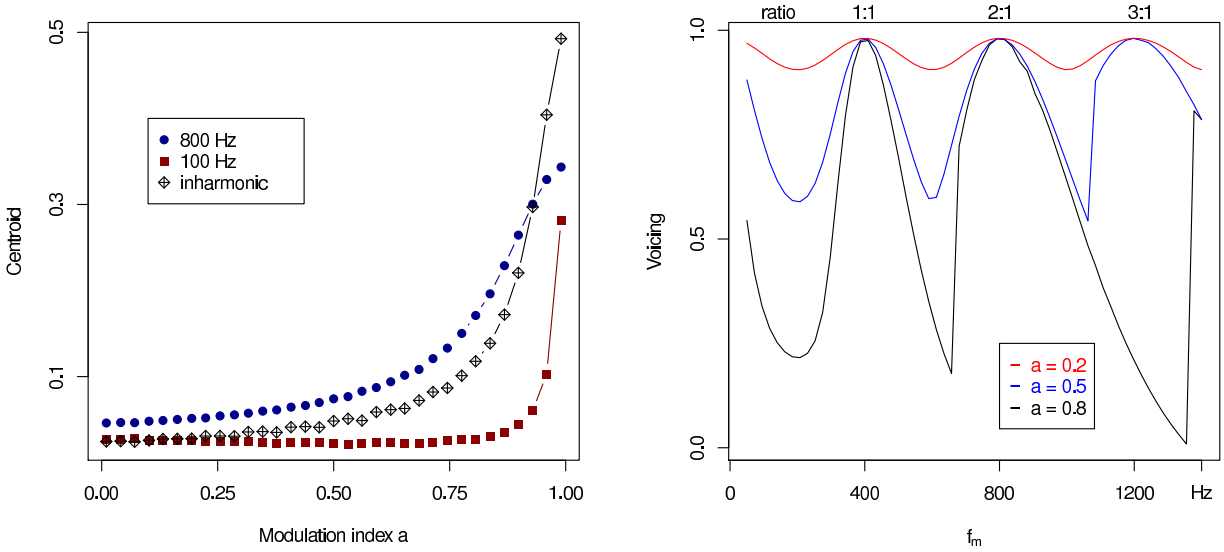

Figure 3.6: Feature extractor values of discrete summation formula synthesis. Left: the centroid as a function of modulation index for $f_c = f_m = 800$ and 100 Hz respectively, as well as for an inharmonic ratio. Right: voicing as a function of the $f_m : f_c$ ratio for three values of the modulation index a.

of these waveforms are strictly bandlimited, although the amplitude diminishes with increasing partial number. This causes problems with aliasing if the waveforms are adopted literally in the digital domain. For that reason, synthesis of bandlimited waveforms is still to some extent an active area of development. Various schemes for reducing aliasing have been proposed, such as storing an integrated waveform in the lookup table, which is then differentiated as the waveform is synthesised (Geiger, 2006).

Here we give a short resumé of the tremolo oscillator, previously presented elsewhere (Holopainen, 2010). Tremolo may refer to either a cyclic variation in amplitude, as a counterpart to vibrato, or to a rapid trill between two pitches. The tremolo oscillator generates a trill between two pitches, which may be thought of as FM synthesis with a rectangular modulator waveform. The modulator waveform has four parameters: two that specify the duration of each pitch level (t_1, t_2) and two for the pitches (F_1, F_2), as illustrated in Figure 3.7. The modulator waveform has a total duration of $T = t_1 + t_2$ seconds and a fundamental frequency of $f_c = 1/T$ Hz. If the two durations are equal, it makes sense to introduce an average frequency $f_m = (F_1 + F_2)/2$. Further, a frequency deviation corresponding to the modulation index in FM may be defined as $\delta = |F_1 - F_2|/2$. Writing the modulator waveform as

$$g_n = \begin{cases} F_1 & \text{if } \frac{n}{f_s} \,(\mathrm{mod} T) < t_1 \\ F_2 & \text{if } \frac{n}{f_s} \,(\mathrm{mod} T) \geq t_1 \end{cases} \tag{3.33}$$

the output of the tremolo oscillator is

$$x_n = \sin(2\pi f_c n/f_s + \frac{\delta}{f_m} g_n). \tag{3.34}$$

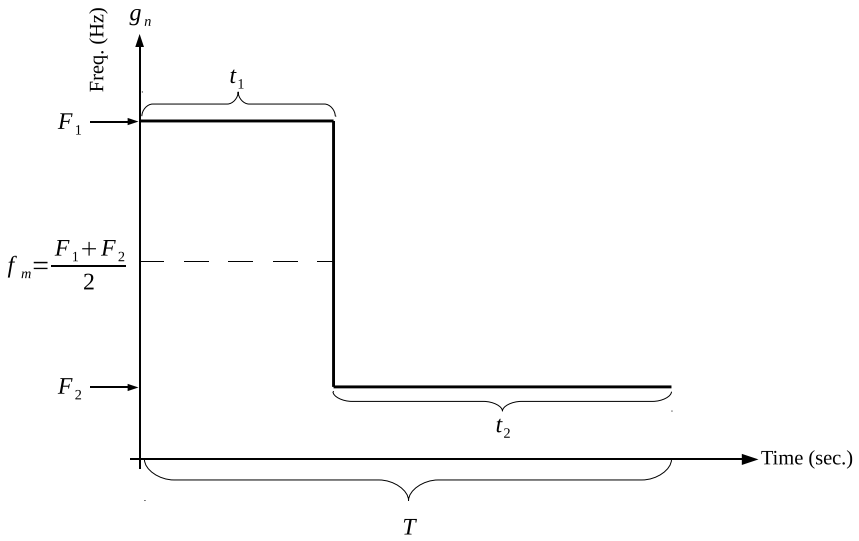

Figure 3.7: The rectangular waveform used for frequency modulation in the tremolo oscillator.

In the tremolo oscillator, the rectangular waveform g_n is smoothed with a one-pole lowpass filter before being used as modulator. Although problems with aliasing are somewhat mitigated by the smoothing filter, it is of course too late to remove aliased components once they have entered the signal.

The aliasing problem begins with the rectangular waveform (3.33) and is compounded by its use as modulator in FM synthesis. As is well known, FM synthesis by itself is not bandlimited and inevitably causes aliasing, although in many situations this is not even noticeable. The amount depends primarily on the modulation index, but for very high carrier and modulator frequencies disturbing levels of aliasing may occur already for very small values of the modulation index.

If one would like to reduce aliasing, the filtering solution is not theoretically correct, even though inserting the lowpass filter after the square wave actually improves the situation. Instead, a better way would be to use a bandlimited square wave for the modulator. Knowing the highest frequency that will be used, the Fourier series of the square wave may be truncated at some suitable partial number. For slow tremolo rates the effect of using a low order truncated Fourier series representation of the square wave is to introduce some wiggly glissando, which may or may not be the intended result. These wiggles, or the Gibbs phenomenon, are the unavoidable consequence of truncating the Fourier series, as is well known (Hamming, 1998).

Switching to continuous time variables, we have the output of the unfiltered tremolo oscillator

$$x(t) = \sin(\omega_c t + \delta \int g(t)dt)$$

using the periodic function $g(t) = g(t + 2\pi)$. Suppose $g(t) = 1$ for a fraction α/π of its period and $g(t) = -1$ the remaining time. Finding the Fourier series for $g(t)$ is easier if

$g(t)$ is taken to be symmetric around $t = 0$. Thus, we define

$$g(t) = \begin{cases} -1, & -\pi < t \leq -\alpha \\ +1 & -\alpha < t \leq \alpha, \\ -1, & \alpha < t \leq \pi \end{cases} \qquad \alpha \in (-\pi, \pi)$$

as an even function, so that we only need to find the cosine terms

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} g(t) \cos kt \, dt$$

of the Fourier series. Doing so, we have

$$\begin{aligned} a_0 &= \frac{1}{\pi}(4\alpha - 2\pi), \\ a_k &= \frac{4}{\pi k} \sin k\alpha, \quad k \geq 1 \end{aligned}$$

from which the truncated Fourier series can be synthesised as

$$\begin{aligned} \hat{g}(t) &= \frac{a_0}{2} + \sum_{k=1}^{N} a_k \cos kt \\ &= \frac{2\alpha}{\pi} - 1 + \frac{4}{\pi} \sum_{k=1}^{N} \frac{1}{k} \sin k\alpha \cos kt, \end{aligned}$$

shown in Figure 3.8 for $N = 33$ partials.

Note that this generates the waveform in normalised amplitude and frequency; it is thus suitable for use in wavetable lookup synthesis. That is, we make a waveform for wavetable lookup synthesis by sampling $\hat{g}(t), t \in [0, 2\pi)$ regularly in L points.

Evidently, it is not practical to change the parameter α (relative tone durations) while running the oscillator. Now, it should be clear why it is not such a bad idea to use the proposed structure with geometric square wave generation followed by lowpass filtering, rather than a theoretically more well-motivated version which may reduce aliasing, but at the cost of less flexibility of parameter changes.

Example 3.3. The tremolo oscillator sounds a bit different depending on how its square wave is generated. Here, first a truncated Fourier series is used, then a plain square wave which causes some roughness, and lastly a lowpass filtered square wave which sounds smoother, although it is also less brilliant than the first version. The modulation frequency increases from subaudio to audio range, displaying the interesting transition region.

The tremolo oscillator is capable of a wide timbral variety, which makes it interesting for use either as a regular synthesis model or as the signal generator in a feature-feedback system, as will be demonstrated in Section 7.2.1.

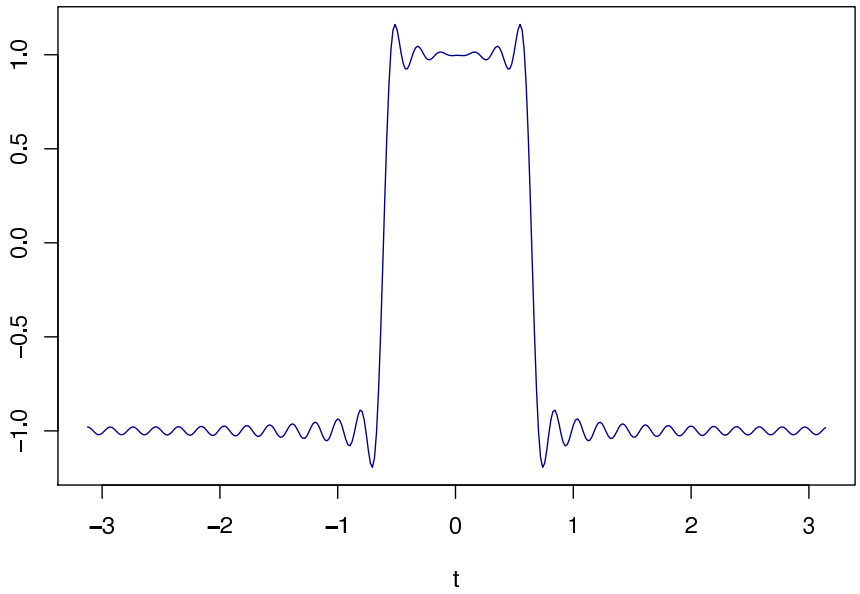

Figure 3.8: Rectangular waveform generated from truncated Fourier series.

3.4 Sound design

All the synthesis models presented so far will have to be fine-tuned and complemented with control functions that shape the sound's evolution over time. Control functions may be automatically generated envelopes that interpolate between breakpoints, they may come from gestural controllers or sensors, or from any arbitrary data in the case of sonification. In autonomous instruments of the kind that we will develop the control functions are generated within the system, beyond direct user control. In some feature-feedback systems the control functions contain rapid fluctuations which may be periodic or noise-like. Such fluctuations of synthesis parameters will introduce modulation effects that are not explicitly a part of the underlying synthesis model. With slower changing parameters, the signal generator will behave more as expected, whereas fast modulation influences its timbre sometimes in unsuspected ways.

We close this chapter with some thoughts on sound design and its limits, sound morphing and extensions of synthesis models by hybridisation, and give examples of higher level control taken from texture synthesis, phrase reconstruction and nonstandard synthesis. Finally, we discuss some criteria for the evaluation of synthesis models and their appropriateness in the context of autonomous instruments.

3.4.1 Timbral fine-tuning

Filters can be applied to the output of any signal generator. Any kind of effects processing may likewise be the final touch that is needed to adjust the output. When the signal generator is part of a feedback loop, then it is to be expected that it matters a great deal whether the final polish is applied within the loop or outside it as post-processing. Whatever modifications are made to the system within the loop may have significant effects on its dynamics, whereas post-processing is always safe. Although we usually do

not apply post-processing in the models presented here and in the following, it must be remembered that it is an important part of the timbral fine-tuning that is often needed to make something work in a mix with several other sound sources, or just to make it sound right on its own.

An important aspect of these final adjustments is the control of the spectral envelope. Equalisers or other filters may be used to shape the spectrum as in subtractive synthesis, or the signal generator itself may be designed so as to emphasise certain spectral regions. The control of the spectral envelope is complementary to the specification of individual partials, and is one of the important characteristics that provide coherence to acoustic instruments across their registers. Similarly to the control of partial domain features (as discussed above in Section 3.2), various features pertaining to the spectral shape may be specified. From this specification a spectral envelope may be generated and imposed on the signal.

A related idea is to group control parameters so as to make their variation partly interdependent. At first, one may think that it is a good idea to have the full flexibility of independent control over every single parameter. Sometimes it is, but with a large number of parameters such flexibility turns into the problem of how to control everything at once. In acoustic instruments one usually finds a positive correlation between amplitude and high frequency content. Many instruments, and notably the piano, exhibit an inverse relation between their spectral richness and fundamental frequency. It can be a fruitful sound design strategy to incorporate such couplings also in abstract synthesis models. Just because they are abstract does not mean that we have to throw any acoustic knowledge overboard.

The first part of this chapter reviewed some partial domain attributes. By weighting selected partials in various ways, sounds with qualitatively different timbre can be produced. Wavetable synthesis is a natural technique when one wants precise control over the partial's amplitudes. On the other hand, wavetable synthesis is restricted to harmonic tones, although weakly inharmonic tones can be synthesised using several wavetables (So and Horner, 2004). Inharmonicity, while by no means a goal as such, has occupied us for a large part of this chapter. The nonlinear group of techniques including FM, waveshaping with ring modulation, discrete summation formulas and wave terrain synthesis are all capable of inharmonic as well as harmonic sounds. This is the group of synthesis techniques that will be used for the autonomous instruments in Chapter 6 and 7 because they are simple to implement, yet they produce varied ranges of sound. Further, the nonlinear synthesis techniques have the advantage over additive synthesis that a single parameter can have a great impact on the whole spectrum, although additive synthesis can achieve a similar global control by the use of grouped partials, as we did in Section 3.2.4.

In an overview of different wavetable and FM techniques for spectral matching of instrument tones, Horner (2007a) concluded that the wavetable techniques generally performed better in producing close matches for a given number of wavetable lookups. In all these techniques static spectra from FM or wavetables are matched against spectral snapshots of the analysed tone and an error criterion is minimised. Unfortunately, restricting the comparison of synthesised and original tones to just one pitch is an artificial limitation. In real applications the entire pitch gamut of an instrument must be consid-

ered. The occurrence of strong formants in some instruments means that matching one single tone is good only for that tone, while other tones cannot just be transposed using the same spectrum. Anyway, the excellent resynthesis of single tones is a very different goal from ours, which is to have flexible, simple sound generators capable of producing a wide variety of timbral qualities. Nevertheless, if spectral matching of two quite different tones is put to use in a combined model, then timbral morphing becomes possible.

3.4.2 Sound morphing

Transformations of one type of sound into another is an important technique for composers working in the medium of sound. Some striking examples can be heard in Harvey's *Mortuos Plango, Vivos Voco*, where the sound of a church bell is transformed into a boy soprano. Other similar transformations can be heard in *Red Bird* and other works of Wishart. Having a synthesis model capable of producing the timbral extremes that are needed for such transformations can be very useful. However, creating convincing sonic transformations is more of an art than a technique. Perhaps these transformations or morphings are more familiar in the visual domain, with applications such as turning the face of one person into that of someone else by a smooth transition.

In sound synthesis there are two different ways to accomplish similar transformations or morphings. Either a delimited sound object or tone is transformed in several steps into another one, such as a piano tone into a banjo tone, or the transformation can be made gradually through time, beginning with one timbre and ending with another. The first type of transformations entail a representation of the temporal envelope of a tone which undergoes some kind of change, whereas the second type only morphs properties of the steady state spectrum (which is a misnomer since it then changes gradually).

Morphing between tones of different instruments has recently been studied by Caetano and Rodet (2010), who carried out the morphing on the spectral envelope. They also analysed the resulting spectral shape with features such as the centroid, spread, skewness, kurtosis and slope, and introduced a way to specify single feature values at various stages in the transformation. Suppose the two sounds A and B are to be morphed and a parameter $t \in [0, 1]$ controls the position between A and B of the transformed sound; then the sound at, say, $t = 0.5$ may take on a range of different values for the spectral shape features depending on how the morphing is carried out. There are several distinct paths that the transition between A and B can take with respect to signal descriptors, which may have their distinctive character.

Morphings from one kind of sound source into another do not always appear plausible. However, one may take two synthesis models and try to merge them into a single model that allows the user a parameterised control of the sonic continuum between their respective sounds. The transformations from sinusoids to noise (Section 3.2.5) is a simple demonstration of the principle, but it is applicable to a large class of synthesis models. When such hybrid instruments are feasible, they may solve the problem of timbral interpolation between two reference sounds. Hybridised synthesis models may also fulfill a need for flexible synthesis models with a large number of parameters. Such models with huge parameter spaces are suitable for automated optimisation of synthesis parameters by evolutionary computing if there is a target sound to be matched. This idea will be

further discussed in Section 6.5.

The noise modulation techniques discussed earlier may be a valuable addition to be merged with many other synthesis techniques. A hybrid flute model was introduced by Ystad (1998) which used waveshaping synthesis for the deterministic (sinusoidal) part and a stochastic component for the turbulent breath noises. In that case, the hybrid nature of the model aimed at producing a coherent instrument sound rather than, as we have proposed, to increase the sonic range. Instrument merging, while being a natural method for sound design, obviously yields a compound synthesis model which is more complicated than its parts. On the other hand, the search for synthesis models that are as simple as possible will make their study easier.

In the case of autonomous instruments, their behaviour will be controlled by parameters that are specified at the start. In contrast to the basic synthesis models that have been considered here, the typical autonomous instrument will not only exhibit a parameter dependence in its timbre, but also in the way the sound evolves over time.

3.4.3 Beyond the note level: Textures and phrases

In sound synthesis, it is far too easy to become captivated by the intricacies of a single note; how to control its timbre and its temporal envelope. In most papers where a new sound synthesis model has been proposed, the problem of how to connect notes into coherent phrases has been silently ignored. Moreover, as far as abstract models are concerned, questions of the temporal control of synthesis parameters are often simply left out. This is perhaps not surprising, since the temporal evolution of synthesis parameters is a problem that particularly concerns imitative synthesis, whereas abstract synthesis has the advantage of being applicable to whatever one can imagine, including the imitation of whatever seems feasible.

Although the phrase level is often neglected, an exception is the software synthesiser outlined by Lindemann (2007) which combines sampling and concatenative synthesis with sinusoidal modelling. The technique called *reconstructive phrase modelling* uses a sample library consisting of recorded phrases rather than individual notes. The phrases are decomposed into slowly varying control data for the amplitude and frequency of partials and a rapidly varying part of control data. The slow data is controlled by MIDI messages, whereas the fast variation, providing the tones with their richness and realism, is automatically added. If the user happens to play a phrase that does not match any of those that are stored in the data base, matching parts of phrases are spliced together by concatenative synthesis.

In texture synthesis, a broad range of synthesis techniques have been employed for the modelling of such diverse sound sources as rain, applause, fire, wind and traffic noise. Diemo Schwarz (2011, p. 221) characterises sound texture as "sound that is composed of many micro-events, but whose features are stable on a larger time-scale". In that sense, sound textures are similar to wallpaper. Schwarz distinguishes expressive texture synthesis from natural texture synthesis, the former being used interactively for musical composition or performance, whereas the latter finds its applications in computer games and installations. The dichotomy between realistic imitative synthesis and more abstract approaches appears again in the split between natural and expressive texture synthesis.

As it happens, feature-feedback systems often generate sounds that fit well into the description of textures; there is much action on short time scales, but the larger time-scale tends to be static. This will be evident from some of the sound examples that accompany chapters 6 and 7.

Apart from Di Scipio's suggestion to use iterated nonlinear functions for the generation of sound textures (Di Scipio, 1999), so called nonstandard synthesis models do not frequently figure among the attempts at environmental texture synthesis. Nonstandard synthesis, although no coherent approach, not only turns its back on acoustic models of sound production (Döbereiner, 2011), but also bypasses the note level altogether by focusing on manipulations of the waveform or individual samples. Nevertheless, what still makes these approaches fascinating today is that a higher level of sonic organisation emerges through the repeated operations of a set of rules. Paul Berg developed a collection of programmes called ASP, for Automated Sound Programs, in the 1970s (Berg, 2009, pp. 81–82). These programmes could run until interrupted, and there was a need for a mechanism that could provide change. Bit-level operations such as masking random numbers using the logical AND operation were used, but the value of the mask could then be determined (presumably less often) by a second mask which resulted in a long-term development which, according to Berg, was reminiscent of the random walks in Xenakis' Dynamic Stochastic Synthesis.

A similar working mode is applied in feature-feedback systems, where there is only one continuous stream of sound and no pre-established temporal boundaries such as beginnings and ends of notes or phrases. Consequently, the concern with timbral properties of synthesised sounds that one can afford to have in standard synthesis models becomes moot. Instead, the sound's evolution becomes increasingly important. Moreover, as we shall see, the timbre is no longer easily controllable in some feature-feedback systems.

3.4.4 Evaluating synthesis techniques

Jaffe (1995) introduced ten criteria for the evaluation of synthesis techniques. This evaluation was later followed up by Tolonen et al. (1998). The criteria deal with aspects such as how intuitive are the synthesis parameters; the physicality of parameters; the behaviour of the synthesis model under parameter changes; and the perceptibility of parameter changes. Further, there are criteria for the diversity and identity of the family of sounds obtainable from a single synthesis technique. Lastly, there are the implementation issues related to efficiency, computational load and memory requirements.

One group of Jaffe's criteria deal with aspects such as intuitively controllable parameters. The additive synthesis models discussed in Section 3.2 were designed so as to translate audio features directly to synthesis parameters, which should make them highly intuitive because of the relatively straightforward link between features and the perceived character. However, we saw that the simultaneous specification of groups of features may lead to mutually contradictory demands, in which case the controllability of all these features together is limited.

The implications of the control parameters for the perceived sound may become more complicated for nonlinear models in general, although many nonlinear techniques still have parameters that influence the sound in rather predictable ways. As long as the

number of parameters is small, it is easy to gain insight into the workings of any of the synthesis models discussed in section 3.3. The wave terrain system is the one with most parameters to define; among the synthesis models presented in this chapter it is without doubt the most complex to come to grips with. Nevertheless, so far we have been on safe ground. In the following chapters, chaos and self-organisation will be central topics, and we will see examples of more complicated synthesis models. If the criterion of intuitive parameters is important, then nonstandard synthesis and autonomous instruments will not do very well in that respect.

Another pair of complementary criteria is the identity of a synthesis model versus its flexibility. A versatile synthesis model such as additive synthesis is equally apt to generate whatever sound one would like, but this means that it is lacking a sonic identity of its own. In contrast, the sounds of standard FM or ring modulated tones will often be recognised as such, much as we recognise the sound of a saxophone. Nonstandard synthesis techniques in general are likely to be highly limited in their capacity to imitate arbitrary sounds, but instead they may have a strong sonic identity. This identity, however, may often be primarily manifested in the sound's temporal evolution rather than the timbre of a short representative segment.

The creation of arbitrary novel sounds and computationally efficient synthesis are listed by Tolonen et al. (1998, p. 85) as the main reasons for engaging with abstract synthesis models such as those discussed in this chapter. Powerful synthesis parameters that change the timbre globally is another reason for using nonlinear (abstract) models. The ease of implementation may be added as another reason to prefer abstract models over spectral or physical modelling.

Some of the nonlinear synthesis models introduced above (in Section 3.3) will be used as signal generators in feature-feedback systems. Then, these synthesis models will exhibit entirely new properties in the new context. Their control parameters will vary rapidly and irregularly. Instead of using direct gestural control of the synthesis models, the autonomous feedback control will shape their dynamics.

In Chapter 7, some design principles will be exposed which are mostly related to the temporal evolution in autonomous instruments. That is a complementary aspect to the timbral sound design that has been discussed in this chapter. However, it is worth stressing that, in autonomous instruments, the system's dynamics may influence the timbre more strongly than any purposive sound design strategy. Thus, one cannot just plug in any basic synthesis model in a feature-feedback system and hope that it will replace another one without fundamentally altering its behaviour. A discussion of chaotic systems and self-organisation will therefore be necessary before we look closer at autonomous instruments.

Chapter 4

Nonlinear Dynamics

The idea of autonomous instruments is related to autonomous dynamic systems. Given an initial condition, the autonomous system evolves by itself without external disturbances. Then, the problem becomes to find out how the system's control parameters and initial condition determine its behaviour. We have already studied the relationship between synthesis parameters and feature extractors, but in this chapter the tools of dynamic systems will be introduced. It has not been customary to study novel synthesis models as if they were some unknown physical system, but in the case of feature-feedback systems, this is the natural approach.

The autonomous instruments that will be introduced in later chapters are mostly deterministic systems. The theory of dynamic systems and chaos are therefore the adequate framework for understanding their dynamics, although it needs to be complemented with perceptually oriented evaluations of the autonomous instrument's output.

Composers have long been interested in the use of chaos in synthesis and note level composition. This provides a cultural context in which autonomous instruments belong. Usually, some chaotic system has been taken from the literature and applied to signal or note generation. In contrast, the present work aims at developing new systems of which we do not know in advance whether they are chaotic or not. Some nonlinear time series methods will be reviewed, which can be used to estimate the chaoticity of a system.

Feature-feedback systems may be thought of as enormously scaled-up versions of low-dimensional iterated maps. This is one of the reasons why a closer look at these simpler maps will provide some basic insight into more complex systems such as feature-feedback systems. In particular, maps can be combined with filters to provide them with memory of their recent past. Such filtered maps may be used as synthesis models, including physical models but also less conventional abstract synthesis techniques. Closely related are a class of nonlinear filters, which are generic models of feature extractors.

Chaos control and synchronisation are also relevant for a better understanding of feature-feedback systems. In chaos control, a chaotic system is tamed and forced to oscillate periodically or to settle on a fixed point. If a filter is inserted in the feedback path of an iterated map, this may be used as a chaos suppressing scheme. Related to such control schemes, in this chapter a first example of a feature-feedback system will be outlined which uses a pitch follower for the automatic control of a nonlinear oscillator.

This chapter is organised into an introduction to basic themes in dynamic systems,

including chaos, nonlinear time series analysis and stochastic signals. Then follows an exposition of uses of chaos in musical composition, sound synthesis, and its occurrence in acoustics in general. Section 4.3 deals with filtered maps, and shows how to think about feature extractors as nonlinear filters. Nonlinear oscillators are interesting systems with an obvious use for sound synthesis, although the pitch of these oscillators may be hard to control. A simple control scheme is suggested in section 4.4. Then the chapter ends with a review of synchronisation and chaos control, and a few applications to sound synthesis are mentioned.

4.1 The state space approach

For the sake of clarity, some of the basic concepts of dynamic systems will be introduced first. Then we discuss the relation between maps and flows as they are being simulated in the computer and point out how this relates to digital oscillators. Sensitive dependence on initial conditions is, in practice, the most important aspect of chaos. The Lyapunov exponent, which quantifies chaos, can be estimated by direct means if the system is sufficiently simple and its equations are known. Otherwise, there are nonlinear time series methods available, not only for the estimation of Lyapunov exponents, but for other quantities as well, some of which will probably become important in music information retrieval. Although the focus here is on deterministic systems, there are important uses for noise in sound synthesis that turn a deterministic system into a stochastic system. After some remarks about stochastic systems, this section ends with a reflexion on some more general problems in detecting and generating chaos. There are several good references on chaos and nonlinear dynamic systems (e.g. Tél and Gruiz, 2006; Strogatz, 1994; Elaydi, 2008) and, for a short introduction with particular relevance to acoustics, see the article by Lauterborn and Parlitz (1988).

4.1.1 Basic concepts

An awareness of common types of behaviour in dynamical systems is fundamental for understanding the synthesis models we will develop in later chapters. Some basic concepts of chaos theory and dynamic systems in general are recalled here for convenience. More rigorous definitions can be found in the literature, and to some extent in Chapter 6. Here, we focus on the case of iterated maps (also called *difference equations*, or just maps for short), but the terminology applies with minor modifications to ordinary differential equations as well.

Nonlinear dynamical systems are conveniently grouped into the following four classes (Frøyland, 1992): *Maps*, *ordinary differential equations*, *cellular automata*, and *partial differential equations*. We will be concerned with the first two categories. Maps have continuous state variables and a discrete time variable. Ordinary differential equations (ODEs, also called *flows*) have both continuous state variables, continuous time and a finite dimensional state space. Partial differential equations are also continuous in variables, and time, but have an infinite dimensional state space. Physical modelling of acoustic instruments is conceptually based on partial differential equations, for instance in describing the motion of every point of a drum membrane over time. Cellular automata

are composed of large numbers of cells, each taking one of a (usually small) number of states, where the state of each cell is updated by simple rules that are functions of some neighbourhood around the cell. We will briefly consider some aspects of cellular automata in the next chapter (Section 5.1.8).

The *state space* (sometimes called phase space) is the set of variables that are modelled, sometimes corresponding to spatial variables, but in mechanical dynamic systems more often including velocity and acceleration, or the first and second time derivatives describing the spatial motion of a particle. We write the first derivative with respect to time with a dot, as in $\dot{x}$, and two dots for acceleration.

A succession of values of an iterated function is called an *orbit*. There are basically the following types of orbits: fixed points, cycles, quasi-periodicity, chaos, and unbounded (globally unstable). The last category is mostly useful to know about for the sake of avoiding it. The period of a cycle is the number of iterations of a map until the sequence of points repeats; hence fixed points are also called cycles of period one.

An *attractor* is the set of points in the state space that the orbit eventually converges to (if it converges to anything). Sometimes there can be several coexisting attractors, each with its own *basin of attraction*, or set of initial conditions that eventually lead an orbit to that attractor. There is usually a transient before a trajectory reaches an attractor, unless it happens to start exactly on the attractor. *Eventually periodic* orbits are such that after approaching a cycle during a finite number of iterations, they become perfectly periodic.

If a fixed point x^* is perturbed slightly by moving it to $x^* + \epsilon$, the system will either return back to the fixed point as the map is iterated, or it will move away from it (in exceptional cases, it may even stick where it is). If a small perturbation in any direction causes the orbit to return to x^* the fixed point is stable, and if any perturbation causes the point to run away from x^* it is unstable. An intermediate case is the *saddle point*, where there is one stable direction, s, such that points perturbed in the direction $x^* + s$ will tend to return to x^*, and an unstable direction, u, such that points moved to $x^* + u$ will wander away from the fixed point. (In fact, the unstable direction must be defined as the stable direction when time is reversed.) The set of points that will approach the fixed point is called the stable manifold, and the set of points that will increase their distance to the fixed point, no matter how close to it they start off, is called the unstable manifold (see Figure 4.1).

Autonomous systems have no explicit time-dependence on the right hand side of the equation, such as in $\dot{x} = f(x)$. *Forced* or *driven* systems have a time-dependent function on the right hand side, such as $\dot{x} = f(x, t)$. However, the non-autonomous system can easily be made autonomous by substituting the forcing term with a new variable.

Bifurcations are the qualitative changes of a system's orbits as one or more of its parameters are varied. Such parameters are therefore often called bifurcation parameters. For instance, a fixed point may split up and become a stable period two cycle while there continues to exist an unstable fixed point, as in the pitchfork bifurcation.

Maps with delays are of fundamental importance for understanding feature-feedback systems, which is why they will be studied in more detail below (see Section 4.3). As an example, consider the Hénon map. This is one of the simplest chaotic maps with one delayed variable: $x_{n+1} = 1 - ax_n^2 + bx_{n-1}$. The usual way to handle maps with

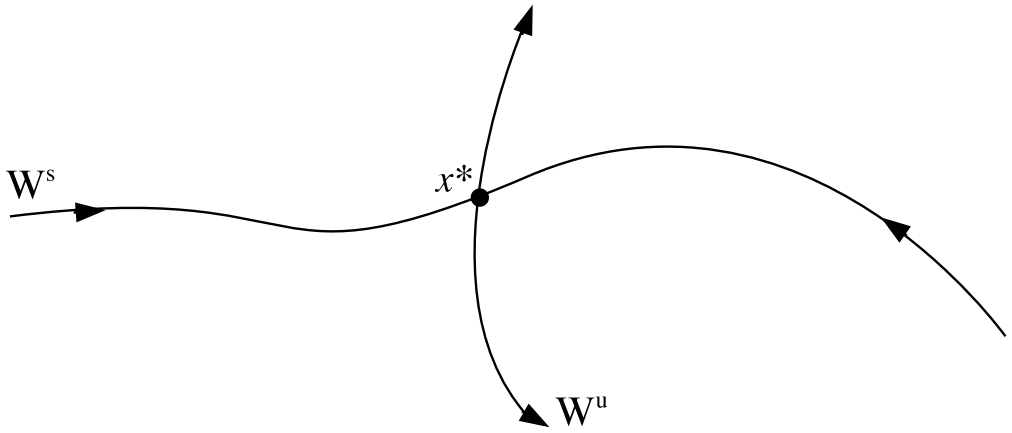

Figure 4.1: Stable (W^s) and unstable (W^u) manifolds around a saddle point x^*.

delayed variables is to introduce new variables without delay, thus increasing the system's dimension. In case of the Hénon map, we make the variable substitution $y_n \leftarrow x_{n-1}$, so the system becomes

$$\begin{aligned} x_{n+1} &= 1 - ax_n^2 + y_n \\ y_{n+1} &= bx_n \end{aligned} \tag{4.1}$$

which is indeed a two-dimensional system where the next state only depends on the current state. Longer delay lines can be handled similarly by introducing a new variable for each time step of delay. For $a = 1.4, b = 0.3$ this map is chaotic and produces a white noise type of sound if the sequence x_n is used as an audio signal.

As b in (4.1) approaches zero, the system turns into the one-dimensional quadratic map. This is another important map which is related to the *logistic* map by a simple change of variables. The logistic map

$$x_{n+1} = rx_n(1 - x_n), \quad 0 \leq r \leq 4 \tag{4.2}$$

is a standard example in virtually every introduction to chaotic systems (e.g. Elaydi, 2008), and we will occasionally refer to it in the following chapters.

4.1.2 Maps or flows?

In classical mechanics, acoustics and many other fields, differential equations are preferred over maps as models of observed phenomena. They are used as models of chemical reactions, the electrical potential in neurons and electronic circuits including analogue musical instruments, just to name a few examples. It is well known that one-dimensional maps are capable of qualitatively different phenomena than one-dimensional ordinary differential equations, most notably there is the possibility of chaos in maps. For ODEs given by smooth functions, three dimensions are necessary for chaos to exist and two dimensions are necessary for periodic motion to be possible unless the system is defined on a circle (Strogatz, 1994).

Numerical simulation necessarily turns an ODE into a map. If the first order Euler method is used, the differential equation $\dot{x} = f(x)$ is solved by iterating $x_{n+1} = x_n + h \cdot f(x_n)$, where h is a suitably small step size. If the step size is taken too large, the system is likely to blow up. As the step size h corresponds to a sampling period of the underlying continuous system, its size determines the period length if the system has a periodic solution. Therefore, it can be useful to make it a time-variable function h_n, where $0 < h_n \ll 1$. This is an easy way to control pitch if the system is used for sound synthesis. For highly accurate solutions, it has been recommended to use an algorithm with adaptive step size with the fourth order Runge-Kutta algorithm (Press et al., 2007). Later in this chapter (in Section 4.4.3), we give some examples where we either use a fixed step size or vary the step size in order to control the pitch of an oscillator.

When the frequency $\omega(t)$ of an oscillator $x(t) = \sin(\theta(t))$ is changing, this is carried out by integrating its instantaneous frequency:

$$\theta(t) \quad = \quad \int_{-\infty}^{t} \omega(\tau) d\tau$$

In digital oscillators, the phase increment is done by the Euler method. For nonlinear oscillators, the phase increment may in general be written as a function $\dot{\theta} = g(\theta, t)$ of both current phase and time. Considering a linear, digital oscillator, the corresponding function $g(\theta_n, n)$ reduces to

$$\theta_{n+1} = \theta_n + h f_n,$$

which is a function of the instantaneous frequency f_n with step size $h = 2\pi/f_s$. There is hardly any reason to use higher order integration methods for phase increments that depend only on time.

A common way to study flows is to take a Poincaré section of it. The Poincaré section of a flow in N dimensions is a subset of this space in $N - 1$ dimensions, restricted to those points where the flow cuts through a reference plane. Then, if the flow is three-dimensional, the system dynamics can be studied as a map on two dimensions instead. A remarkable feature of some two-dimensional maps is that they may be invertible, yet chaotic. In fact, the direction of time in ODEs is reversible, hence a Poincaré map should be reversible as well (Tél and Gruiz, 2006). So, from a physical standpoint flows are primary and maps are tools for studying them. In sound synthesis (and musical composition), maps may serve equally well as signal sources. However, even when chaotic, flows are characterised by a continuous motion which ties them closer than maps to acoustically plausible signals.

4.1.3 Lyapunov exponents

Sensitive dependence on initial conditions is one of the hallmarks of chaos. If the system is chaotic, then the distance between two initially very closely situated points in state space will increase exponentially for almost all choices of initial position. This exponential divergence goes on until its size is comparable to that of the whole attractor. The (greatest) Lyapunov exponent is a measure of how chaotic a system is. Specifically, it

measures the rate with which an infinitesimal perturbation of an initial condition grows. For a system in N dimensions, there exist as many Lyapunov exponents as dimensions, $\lambda_1, \lambda_2, \ldots, \lambda_N$, ordered from highest to lowest value. If $\lambda_1 > 0$ the system is chaotic, and if there are more than one positive Lyapunov exponent it is called *hyperchaotic*. Sometimes the largest exponent of a system in more than one dimension is simply referred to as "the Lyapunov exponent".

For a one-dimensional map $x_{n+1} = f(x_n)$, the Lyapunov exponent is defined as

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} \ln |f'(x_n)| \,. \tag{4.3}$$

In some simple cases such as piecewise linear maps, this expression can be calculated precisely, otherwise only approximate estimations are available.

The case of multidimensional state spaces is more interesting. If an infinitesimal region of the state space is taken as a set of initial conditions, then there will generally be a deformation of the initial region as time advances. In chaotic systems, this region will typically stretch in one direction and shrink in other directions. If the volume of this region is preserved the system is *conservative* (i.e. area or volume preserving), whereas if the volume shrinks, the system is *dissipative*. Friction is the usual cause of dissipation in mechanical systems, which for example finally brings a pendulum to halt. If they are globally stable, dissipative systems have a greater squeezing of the phase space than stretching, but there is also a folding going on which results in the system gradually approaching a strange attractor.

Chaos may appear in both dissipative and conservative systems, and it is correct in both cases to speak of sensitive dependence on initial conditions, but in very different ways. For dissipative systems, the sensitivity on initial conditions implies that there is a strange attractor towards which points in the state space move, whereas for conservative systems, two nearby initial points may have different dynamics; one may be chaotic while the other is periodic. In conservative systems, there are no attractors, but each orbit remains on a set that depends on its initial condition. This has important consequences if conservative systems should be used in sound synthesis. Then, the initial condition must be carefully selected in order to get the desired kind of trajectory.

For the computation of the Lyapunov exponents of a map or a flow in more than one dimension, its Jacobian is needed. The Jacobian is a matrix containing the partial derivatives of a system, e.g. for two dimensions:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}$$

From the Jacobian determinant one can see whether the system is dissipative or conservative. If a map has $|\det(J)| < 1$, the system is dissipative; in the limit where $|\det(J)| = 1$ for all points in phase space the system is conservative.

Experimental methods for obtaining the greatest Lyapunov exponent or the full spectrum of exponents from time series resulting from measurements have also been proposed (Wolf et al., 1985; Kantz and Schreiber, 2003). This is necessarily more difficult, since the

derivative or Jacobian of the system is unknown. In time series analysis, the embedding method is used as described below.

For the analysis of chaotic digital synthesis models the system is both known and controllable, hence the prediction time, or duration until two nearby initial conditions substantially diverge, can easily be observed. This is not to say that the measurement of Lyapunov exponents is trivial, especially if the system is complex and has many dimensions. Depending on initial conditions, including the distance and direction of initial separation, Lyapunov exponents may differ. In Chapter 6, we will explain in some detail a method for Lyapunov exponent estimation suitable for feature-feedback systems and other high-dimensional synthesis models (see Section 6.1.8).

Chaotic orbits have continuous spectra. For periodic or quasi-periodic orbits, there are separate lines in the spectrum, but a characteristic of chaos is that it is noisy. Its noisiness may however manifest itself in several ways. Low-dimensional chaotic maps usually produce white or coloured noise. In flows on the other hand, the waveform varies more smoothly over time. For instance, the Rössler attractor (eq. 4.23 below) may produce almost pitched, but rough sounds.

4.1.4 Nonlinear time series analysis

In many real-life situations where one has a signal to analyse, the system that generated the signal is unknown. Assuming that the signal comes from some deterministic dynamical system, methods from nonlinear time series analysis are adequate. One of the problems that one tries to solve with nonlinear time series analysis is to reconstruct the attractor of the dynamical system that generated the observed signal.

The basic method of most work in nonlinear time series analysis is that of *delay embedding*. A time series x_n is analysed by forming D-dimensional vectors $\mathbf{x}_n$ of delayed samples,

$$\mathbf{x}_n = \{x_n, x_{n-\tau}, x_{n-2\tau}, \ldots, x_{n-(D-1)\tau}\}$$

for some suitable delay time τ and embedding dimension D. From this representation the underlying attractor may be reconstructed if the delay and embedding dimension have been properly chosen. The method and its potential pitfalls have been described in great detail in the literature (Wolf et al., 1985; Kantz and Schreiber, 2003). A common way to determine the most suitable delay time is to inspect the autocorrelation of the time series and set τ to the lag corresponding to the first zero of the autocorrelation function. Takens embedding theorem states that, for an attractor of dimension d, an embedding dimension of $D = 2d + 1$ is sufficient to recover the original attractor, although in practice, lower embedding dimensions will often be enough. From this delay embedding representation, the Lyapunov exponent (or even the full spectrum of exponents) may be estimated, as well as the attractor's fractal dimension.

Nonlinear time series analysis using the embedding method has thus far been sparsely used for the analysis of musical signals, at least as compared to various time-frequency methods (Reiss and Sandler, 2003). However, it must be said that the applicability of delay embeddings to recordings of music, as opposed to individual tones from single instruments (Bernardi et al., 1997), is a bit questionable, since one cannot assume that music in general comes from a deterministic dynamical system. The situation is quite

different with sounds synthesised from dynamical systems, including feature-feedback systems. In that case, there is reason to believe that these methods of analysis are not only appropriate, but may contribute to a better understanding of the system. This is a field of investigation that awaits further studies, and interesting discoveries may lie ahead.

The reverse problem is to find a dynamic system that generates a signal that is identical with, or at least highly similar to, the analysed signal. Given that oscillating systems may be formulated as second order differential equations of the form $m\ddot{x} + b\dot{x} + kx = 0$, it should be possible to analyse an arbitrary sound and try to find a functional relation between its current amplitude, first, and second derivatives. This method is mentioned, though advised against by Kantz and Schreiber (2003, p. 153) because the derivative filters tend to amplify noise in the signal. If this noise amplification is bad already in the first derivative filter, it becomes even more precarious in the second derivative, which is equivalent to applying the derivative filter twice. Other problems to overcome include that the signal may not fit this model differential equation, or it may be non-stationary. For time-varying parameters, short windows could be analysed separately. Embedding methods have been proposed for solving the reverse problem and synthesise tones from various instruments (Mackenzie, 1995; Röbel, 2001), but this technique seems to be little used.

A common method often employed in nonlinear time series analysis is to observe some signal descriptor as a function of a scale size. In fact, we demonstrated such an approach in Chapter 2, where the spectral flux was plotted as a function of the analysis window size (see Figure 2.6 on page 72). The only thing that remains to do is to fit a regression line to the flux as a function of window length, and the slope would indicate how the flux scales as a function of duration over which it is measured. In particular, the *fractal dimension* can be estimated this way. There are several variants of fractal dimension, but the simplest is the *capacity dimension*, also called the *box counting dimension* after the method used to estimate it. An object such as an attractor in an M-dimensional state space is covered with a set of M-dimensional boxes of a given size ϵ, and the number N_ϵ of boxes that contain points of the attractor are counted. A regression line of N_ϵ as a function of $1/\epsilon$ is fitted on a log-log plot, and its slope gives the box counting dimension.

Novel methods of nonlinear time series analysis are still being developed, some of which are compelling either by their computational simplicity or the insights they may provide. Let us mention the 0-1 test for chaos (Gottwald and Melbourne, 2004), which is a simple tool designed to test for the presence or absence of chaos in deterministic time series, although it does not provide the nuanced quantification of the greatest Lyapunov exponent. Another interesting new development is the method of so called *visibility graphs* (Lacasa et al., 2008), in which the samples of a time series are translated into nodes of a graph. The nodes are connected if the corresponding samples are visible from each other, in the sense that no higher sample in-between blocks the view to another sample. This approach makes all methods of graph theory available to time series analysis.

Permutation entropy (Bandt and Pompe, 2002) is computed by sorting groups of adjacent observations $x_n, x_{n+1}, \ldots, x_{n+T}$ from a time series into different classes of contours. For example, $x_n < x_{n+2} < x_{n+1}$ is a different contour than $x_{n+2} < x_{n+1} < x_n$. The entropy is calculated from the number of occurrences of these distinct contours or

permutations. The permutation entropy increases with the Lyapunov exponent, which is why the authors proposed it as a complexity measure.

There is reason to believe that visibility graphs and permutation entropy may prove illuminating also in the analysis of musical audio signals, and probably even more so in the analysis of melodic pitch contours, since both of these measures deal very directly with the contour of a time series. Indeed, the permutation entropy in conjunction with a measure related to statistical complexity have recently been proposed to be of potential use for distinguishing musical genres from the analysis of audio samples (Ribeiro et al., 2012).

Time series analysis is applied in situations when the equations governing the system that generated the signal are unknown, otherwise more direct approaches become possible such as numerically solving the system from different initial conditions. However, some high-dimensional dynamical systems are already so complicated that the computational analysis of Lyapunov exponents from the system equations is no longer feasible, making time series analysis a better option.

4.1.5 Notes on noise

In deterministic systems, if the system equations and the initial condition are known, then the system's evolution is given once and for all. This should not be confused with being able to predict exactly what will happen in the future of a deterministic system. One cannot, if it is chaotic. Pseudo-random number generators are also deterministic and produce the same sequence of numbers if they are given the same seed. Hence, if such a noise source is used for stochastic composition, and provided it is not seeded with new values and everything else remains the same, two runs of the stochastic composition programme will generate identical output.

White noise differs from chaos, in that (low dimensional) chaos has a finite positive Lyapunov exponent and finite positive prediction time. Loosely speaking, the prediction time is the time it takes two orbits starting from infinitesimally separated initial conditions to diverge, say, to the size of the whole attractor. This happens quickly, though not immediately in deterministic chaotic systems. Two realisations of an uncorrelated stochastic process, however, would not show such a gradual exponential divergence, but begin to differ immediately.

The presence of noise is unavoidable in any physical system, as opposed to computer simulations. But even in numerical computation round-off errors enter. However, an orbit starting from an unstable fixed point may always stick to that point in a computer simulation, whereas this could never happen in a real-world situation. This observation can be related to electroacoustic feedback, of which several examples will be given in the next chapter, Section 5.1. In such compositions as Di Scipio's Background Noise Study (Di Scipio, 2003, 2011), background ambient noise is amplified, digitally processed and fed into the speakers to cause feedback. A computer simulation of the same setup, but excluding any ambient noise, would be likely to remain silent for all time, no matter how unstable this silent state is. The same could be said of computer simulations of any other pieces involving acoustic feedback.

The fact that noise may increase detectability of weak periodic signals or enforce

periodic behaviour is known as *stochastic resonance*. If the signal-to-noise ratio of a system exhibiting stochastic resonance is plotted as a function of noise level, the curve typically has a peak at some moderate noise level and falls off on either side. At first, it was put forward as an explanation of the periodicity of ice ages, but many other systems have been suggested as likely candidates of stochastic resonance, including the auditory system (Wiesenfeld and Jaramillo, 1998). More specifically, it is the Brownian motion of the inner hair cells which is a suggested explanation for the detectability of weak auditory stimuli. A novel synthesis method, or perhaps rather audio effect, using stochastic resonance on the amplitude spectrum was introduced by Cadiz and Cuadra (2010). A threshold level is set proportional to the amplitude in each bin, and if a stochastic signal exceeds that threshold, the amplitude of that bin is changed by the supplied noise, otherwise the amplitude is set to zero.

In dithering of audio signals, a small amount of noise is added to the signal before quantisation, and in some schemes including feedback from the quantiser's output to its input. In fact, some forms of stochastic resonance can be understood in terms of dithering (Wannamaker et al., 2000). According to Gammaitoni (1995), it was electronic engineers who started using dither in the 1950s, whereas the history of stochastic resonance research only goes back to the 1980s. As Gammaitoni also points out, the "resonance" of stochastic resonance is in many cases a misleading term; "noise induced threshold crossings" would be more appropriate. Usually, resonance refers to the frequency at which an oscillating system reaches its highest amplitude, but in stochastic resonance, the parameter is noise strength rather than frequency.

Noise shaping is applied in requantisation from high resolution digital signals to signals with lower bit resolution. The dithering noise is applied before the quantisation step, and the output is fed back through a noise shaping filter with the purpose of shaping the noise so as to be as little audible as possible (Lipshitz et al., 1991). This is accomplished by using a filter that results in a noise spectrum which in effect follows the shape of the hearing threshold.

Stochastic differential equations are often formulated as systems of the form $\dot{x} = f(x) + \xi_t$, where ξ is some stochastic process. Such processes are common in physics and finance, but they are also conceptually useful in sound synthesis using noise and feedback. An example will be given in Chapter 6 (see Section 6.4), albeit in discrete time. Different ways to formulate Brownian motion in stochastic differential equations have been proposed by Einstein, Ornstein and Uhlenbeck, and Langevin (Lemons, 2002). In general, such solutions involve a description of the time varying probability distribution as a partial differential equation.

4.1.6 Detecting and generating chaos

Before chaos theory became an established discipline, it must have been hard for researchers who were observing chaos to make sense of the phenomena. Deterministic equations were not supposed to produce very complicated behaviour. A few researchers were working in isolation on these problems, such as Lorenz on his atmosphere model in 1963 and Ueda on nonlinear oscillators around 1961 (Abraham and Ueda, 2000). What later became known as chaos was variously described as stochastic motion, "determinis-

tic nonperiodic flow" (Lorenz, 1963) or simply transients. The prehistory of chaos surely goes as far back as to Poincaré's studies of the three-body problem, but this is not the place to trace that fascinating development.

As a student, Ueda was working on frequency entrainment under the guidance of Chihiro Hayashi. He was requested to assist with doing calculations and drawing a diagram for a difficult experiment. The experiment did not turn out quite as expected, and Ueda recalls Hayashi's reactions:

> "Oh, it's probably taking time to settle down to the subharmonic oscillations. Even in an actual series resonance circuit, such a transient state lingers for a long time." [...]
>
> I may have been worried at that time that, had Prof. Hayashi seen this data, he would have told me to repeat the analog computer experiments until the transient state settled to a more acceptable result (Ueda, 2000, p. 34).

What Ueda was observing was chaos, but without a place for it in the conceptual apparatus at that time, the phenomenon could not be recognised for what it was. In some cases, as we will also discuss in Chapter 6, it is not evident whether a phenomenon is just an extremely long transient which will eventually converge to regular (periodic) motion, or if it is persistent chaos. Some feature-feedback systems can have very long transients, going on for a longer time than we care to observe; yet it may be likely that they slowly converge to some cycle or fixed point.

When using digital computers for simulations, it is common practice to spell mathematical statements as if they were taking place in, say, the real numbers, and not some subset of the rationals. Perhaps it may seem pedantic to remark that computer calculations of an iterated real valued function $f : \mathbb{R} \to \mathbb{R}$ is actually a different, rational valued function $f : E \to E$ where E is the computer's floating point approximation of the real numbers. But it is true, and there is a difference, although we will usually not notice it.

In chaotic maps, the theory assumes that the number system is the real numbers (or the complex numbers, as the case may be). The real numbers is a set with remarkable properties, which are intimately linked to certain aspects of chaotic maps. Georg Cantor introduced several of these properties, such as the fractal set that carries his name. The simplest possible chaotic maps depend on being defined on a *real* interval. There are elegant proofs that a map such as $f(x) = 2x (\mathrm{mod}\, 1)$ is chaotic (Elaydi, 2008); all that is needed is an irrational number to start from. When irrational numbers are not available (and they are not in finite precision representations), then chaos is out of the question.

This is an unsettling conclusion, because we will deal very much with chaos in digital synthesis techniques. Are those synthesis techniques really not chaotic at all? We will have to adopt a pragmatic view, and assume that we are using the real numbers as long as the resolution of the floating point numbers is sufficiently high. For all practical purposes, what appears in the sequence of digital samples as a chaotic signal will be counted as such. Already Lorenz (1963) pointed out that, if a differential equation is solved in the computer with a limited numerical precision, then after some (perhaps very large) number of iterations, the solution must return to a previously visited point in the state space so that, strictly speaking, only periodic orbits are possible. However, the period

may be extremely long in comparison to the length of an orbit that one is interested in observing.

Usually, the computer output is regarded as a simulation, and the mathematical notations as an ideal description; then there is nature, which is being modelled. However, as we have argued in Chapter 1, the synthesis models that we develop should not be regarded as models of some existing natural phenomena, at least not in any immediate way. Rather, they must be seen as real enough themselves, and the dynamics they expose is the phenomenon that we study.

4.2 Chaotic systems in music

Since chaos and fractals began to be popularised around the 1980s, they have found many applications as resources in musical composition and sound synthesis. Chaotic systems, inasmuch they are admitted to run undisturbed, are prime examples of autonomous instruments when used for sound synthesis. There are many conceivable ways in which a chaotic system can be employed in music, depending on the chosen mapping. Attempts to use chaotic systems for musical material have been twofold: either they have been applied on the note level, or directly as a means of generating an audio signal. Further distinctions can be made in the latter case. Many chaotic systems are capable of producing a rich and varied range of signals if the orbit is taken as sample values. This is especially true for ordinary differential equations, whereas chaotic maps often simply produce noisy sounds with occasional periodic sounds with periods that subdivide the sampling frequency, depending on parameter value. Chaotic systems may also be used on an intermediate level such as granular synthesis, or to generate control signals.

Sometimes chaotic systems, when used for sound synthesis, are grouped with the nonstandard class. Although this is often well motivated, many acoustic instruments are also capable of chaotic vibrations, some cases of which are mentioned below in Section 4.2.3.

If the spectrum of a chaotic map is identical with that of white noise, why not then use a random number generator instead? Potential perceptually relevant differences between deterministic chaos and noise, when applied on the note level, will be addressed after a review of some of the note-level usages of chaotic systems.

4.2.1 Chaotic systems in note level composition

One of the earliest studies of chaotic maps as generators of musical material was done by Jeff Pressing (1988). He used several different maps in one to four dimensions. The numerical values were translated into musical parameters such as pitch, inter-onset time, envelope attack time, dynamics, and tempo. Unless some quantisation is used, the pitch parameter will take on a continuous range of real values which cannot be notated in standard musical notation. Pressing, however, used a Csound instrument for the generation of musical output, where the continuous parameter ranges cause no problem. We will refer to similar mappings from discrete events to either standard musical notation or Csound scores and similar formats as *note level* composition.

For applications of chaos in composition at the note level, some of which are discussed by Pressing (1988) and Bidlack (1992), choices can always be made about how the output of the chaotic system is treated, and how these number sequences translate to musical parameters. As far as the artistic end result is concerned, these composition strategies may well be much more important than the choice of a specific chaotic system. Nevertheless, if one decides to compose using a certain chaotic system to generate the material, certain compositional strategies may bring out the character of the material better than others. As a general observation, note-based composition aimed at human performers, being discretised in time by a more or less fine-grained grid of potential note-onset times, conceptually corresponds better with maps than with flows. However, taking a Poincaré section of a flow reduces its dimension by 1, as well as making it a discrete time map, which is more suitable for note oriented composition (Bidlack, 1992).

Another method suggested by Gogins (1991) is to use iterated functions systems as a means to produce scores. Iterated functions systems are collections of maps. As the system is iterated, one of the functions is applied to the previous value at each iteration. The function to apply at each iteration may be picked either randomly or in turn. Fractal images can easily be constructed this way, usually by applying a set of affine transformations a number of times to a set of randomly distributed initial points (Elaydi, 2008). As the system is iterated the set of points gradually approach an attractor. Gogins' idea was to map this attractor to pitch and time, thus producing a musical score. This is fundamentally different from the way we are interested in using iterated maps. Using iterated functions systems, the score only becomes available at the end of a process of iterations as an approximation to the attractor, while in all applications we will develop, the iterations themselves correspond to the flow of time.

Chaotic systems lend themselves to the generation of time series suitable for use in musical composition, but it is less obvious how they could be used as variation techniques. Such an application was suggested by Dabby (1996), who used the Lorenz attractor to create variations on note sequences. First, a pitch sequence is mapped to the x-axis of the Lorenz attractor as it unfolds from a particular initial condition. Then a second initial condition is chosen, typically close to the first, and a new trajectory is generated. Finally, the pitch sequence is retrieved from the altered positions of the points on the x-axis. What this all amounts to is just a permutation of elements in temporal succession. Using a chaotic attractor for this purpose may seem overly complicated for such a simple operation, but on the other hand, it appears unlikely that exactly the same results could easily be attained by any simpler method.

Coca et al. (2010) generated tonal melodies from chaotic flows and maps. They used several three-dimensional ODEs and mapped the three coordinate to pitch, duration and amplitude. The pitches were quantised to major or minor scales; durations and amplitudes were similarly quantised to a small number of values. Not surprisingly, they found that the smooth flows when mapped to melodies were dominated by step-wise motion. However, with their mapping, the ratio of small steps to large leaps is easily controlled by setting the sampling period at which the continuous-time flow is translated to note parameters, but this is something they apparently did not consider. Their study is nevertheless interesting by their use of a large set of melodic features from which they obtain objective descriptions of the chaotic melodies. Then, they compare these features

with those similarly obtained from chaotic maps and classical music. In conclusion, melodies generated by flows are characterised by their smooth melodic curves, whereas maps have more skips, and classical melodies are somewhere between these extremes.

Note level composition is a bit removed from the immediate concerns of sound synthesis, but in self-organising algorithmic composition with autonomous instruments there is a higher level than that of the waveform that must be somehow taken into account. The next question is: What is so special about chaos? Is it really different from stochastic time series?

4.2.2 Markov chains and coarse graining

Given a chaotic map, it will reveal utterly different properties if its trajectory is used as a sequence of audio samples, or if it is mapped to the pitches of a sequence of notes. This much is obvious for physical reasons alone; it implies comparing a rapid variation of sound pressure with a much slower variation of fundamental frequencies. A fairer example is the comparison of a chaotic map used for audio samples versus the same map used on a slower timescale to control an amplitude envelope with interpolation between adjacent points. Nevertheless, it can be revealing to compare the sample sequence and a pitch sequence obtained from the same map. For the sample sequence, its amplitude spectrum is what matters most perceptually. In contrast, a pitch sequence acquires its characteristics by factors such as the statistical distribution of pitches and pitch transitions. This idea was already explored by Olsen and Belar in their construction of a "composing machine" in the beginning of the 1950s (Manning, 1993; Holmes, 2008), where melodies were randomly generated from Markov chains found by analysing popular Steven Foster songs.

Chaotic maps may produce white noise, but may have a probability distribution of the signal's amplitude that is far from uniform, Gaussian, or any other familiar shape. Consider the case of deterministic chaos in a one-dimensional map defined on an interval. Its current value $x_{n+1} = f(x_n)$ depends only on the previous value. If this map were formulated as a Markov probability matrix, each entry x_n would be followed with probability 1 by the entry x_{n+1}, with the caveat that the current value must be known to infinite precision. In practice, the interval would be partitioned into small regions, from which the transition matrix can be constructed.

Another way to state this comparison is that apart from some so called pathological functions, a map on a finite interval has a finite number N of preimages; in other words, $f^{-1}(x)$ has at most N different solutions (as illustrated in Figure 4.2, where most points have three distinct preimages). As an example, continuous functions with one local maximum such as the logistic map has $N = 2$. For a high quality pseudo-random number generator, the number N needs to be as high as possible. Hence, a map on the interval has the property that for a given value x_n, it has at most N possible values x_{n-1} and just a single possible value x_{n+1}, whereas a random sequence has, theoretically, an unlimited number of possible previous and succeeding values. In the pitch sequence example, this translates to the number of possible notes preceding and succeeding a given note, and from the argument it should be clear why deterministic chaos may be used to produce pitch sequences with some recognisable traits that differ from a random sequence.

At least one study has been made of the ability of listeners to identify one item from a

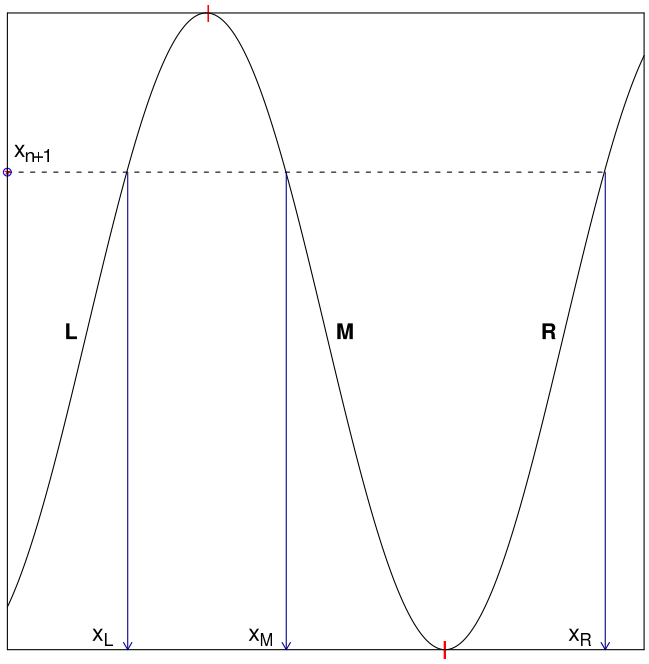

Figure 4.2: Preimages of a map. If the next point is x_{n+1}, the current point may be one of x_L, x_M, and x_R. The three monotone branches of the curve marked L, M, and R form the partition for a symbolic dynamics.

pair of pitch sequences as random, the other one being generated by deterministic chaos (Gregson and Harvey, 1992). Note sequences from several maps with different Lyapunov exponents and attractor dimensions were used. Although the study is not quite conclusive as to what strategies the listeners were employing to recognise the chaotic sequences as nonrandom, the authors indicate that they may have used different cues for different maps. The test subjects also improved their ability to distinguish the sequences as they were exposed to more examples. If psychological studies of listeners' abilities to distinguish chaos from random noise are sparse, there is a rich literature on time series analysis techniques that may be used for such discriminations (e.g. Atay et al., 2009).

The idea to compare deterministic maps with Markov chains arises naturally from coarse graining of the orbit. Coarse graining is a concept introduced in symbolic dynamics (Hao and Zheng, 1998), where one studies the dynamics of a map after having partitioned its phase space into suitable subsets, each given its own symbol. One-dimensional maps are typically partitioned into subsets consisting of monotonous segments, divided by local extrema (compare Figure 4.2 again). For a map $f(x)$ with one maximum, the part to the left of it may be labeled L, and the part to the right R. Then the symbolic dynamics is just the sequence of those two letters as it is generated by an orbit $x_n = f^n(x_0)$. Apparently, much information is lost in the coarse graining since real numbers are reduced to a small set of symbols. However, if a symbolic sequence is known to come from some chaotic one-dimensional map, then the original times series can be reconstructed within some error tolerance (Hubler, 2012). This means that the symbolic sequence provides a highly efficient compression of the original data.

Coarse graining is not without interest in the completely different context of the

analysis of notated music. In fact a kind of coarse graining is used in contour analysis, where the exact notes of a pitch sequence are not considered, but only their relative pitch height and their contour (Friedmann, 1985). Several representations of pitch contour have been introduced. The *contour adjacency series* labels rising intervals with "+" and descending intervals with "-" signs. The *contour class* represents the relative positions by sorting the N pitches of a motif and writing them as a list where 0 is the lowest note and $N-1$ the highest pitch. Contour analysis is most usefully applied to relatively short note sequences, allowing the analyst to discover similarities between motifs that are not direct transpositions.

4.2.3 Acoustics and chaos

"Nonlinear dynamics also appears in musical instruments, where oscillations are produced to generate sound. These systems are therefore *a priori* apt to generate chaotic sound waves. But there are not many investigations in this area." This remark by Lauterborn and Holzfuss (1991, p. 14) still holds today; there is not very much literature in this field, but there is some. An often cited example is the paper by McIntyre, Schumacher and Woodhouse, which has been influential for physical modelling of acoustic instruments, but they also hint at parallels with iterated chaotic maps, noting that such cases are more related to "some of the sounds made by novice instrumentalists, than to those normally made by skilled musicians" (McIntyre et al., 1983, p. 1326). Other acoustic systems have been more exploited by chaos physics. For a review of the methods of nonlinear dynamics with particular relevance to acoustics, see Lauterborn and Parlitz (1988).

Bubble oscillators have been a popular subject for experiments with acoustic chaos (Holzfuss and Lauterborn, 1989; Lauterborn and Holzfuss, 1991). When high intensity ultrasonic sound waves are projected into water, there appears clouds of tiny oscillating bubbles known as cavitation noise. Despite the fact that the ultrasonic driving is of a single frequency, a broadband noise has been measured as the response from the bubbles in the water, as well as period doubling bifurcations.

There are various possible bifurcation scenarios, but one that has been found in acoustic systems is the *subharmonic* route to chaos (Lauterborn and Cramer, 1981). In the spectrum of systems showing subharmonic routes to chaos, other subharmonics than those of period doubling (i.e. octaves) may be found as well, often in combination with harmonic overtones on these subharmonics. When excited with a sinusoid of sufficient amplitude, gongs and cymbals may also produce subharmonic frequencies. There is also a more prominent upwards spread in the spectrum; as the gong or cymbal is struck, first the sound builds up on low frequency modes, then higher modes are gradually excited (Legge and Fletcher, 1989). Subharmonic tones can also be produced in bowed string instruments, and they occur on the octave below the fundamental in starting transients. By using a heavy bow pressure, aperiodic oscillations with a longer period than the string's natural period can be obtained. This phenomenon, familiar from the contemporary string instrument performance practice (usually called *crush tones*), could also be generated by the model proposed by McIntyre et al. (1983).

Another experiment that might as well have been devised by an experimentally minded composer is that of Kitano et al. (1983). They had an experimental setup con-

sisting of a loudspeaker, a microphone connected to a full-wave rectifier and amplifier, connected in turn to the speaker, thus causing feedback. Period doubling bifurcations leading to chaos were observed as the amplifier gain was increased. They also derived a theoretical model that produced comparable results.

Fractals are closely related to chaos in several ways, for instance, a strange attractor is a fractal. Strict self-similarity over several scales (similar to the mathematically constructed Koch curve) is untypical of acoustical signals, but a statistical self-similarity may be found in certain sounds. Digital synthesis of fractal curves that are self-similar over a large range of scales is certainly possible. In fact, Manfred Schroeder (1986) introduced that idea by using a construction similar to the Shepard tones, but instead proceeding from the Weierstrass nowhere differentiable function. Essentially, this is an infinite sum of sinusoids at a fixed frequency ratio. Schroeder showed that one can construct such an audio waveform that will sound as if it were transposed down by a semitone although it was actually played at twice the speed. Although this is a self-similar fractal and fractals are for good reason often associated with chaos, it should be noted that its generating mechanism has nothing to do with chaos.

Motivated by the non-stationarity of musical signals, Bernardi et al. (1997) introduced a local fractal dimension, which is measured over short time windows and reveals changes in fractal dimension over time. They also analysed woodwind multiphonics with the embedding method and found the typical signs of low-dimensional chaos, such as an attractor that occupies a limited volume of the reconstructed phase space.

Chaos is to be expected in certain nonlinear systems including acoustic musical instruments. Therefore it seems questionable to relegate all examples of chaotic sound synthesis to the nonstandard category. There seems to be a rich potential of modelling instrumental sounds by flows. Some attempts in that direction will be mentioned next.

4.2.4 Uses of chaos in sound synthesis

As Slater (1998, p. 16) notes: "Probably all modular analog music synthesizers are capable of chaotic FM synthesis." It is a likely result if the patch cords have been wired so that there is feedback or a cross coupling of oscillators. Hence, it is reasonable to suspect that chaotic sound synthesis was experimented with already in the early days of modular synthesisers in the 1960s, unbeknownst to the musicians themselves. Of course, the notion of chaos was not developed yet, and those experimentators who actually studied it had other names for it.

Since acoustic instruments had been shown to be capable of chaotic oscillation, one would expect an interest for this in physical modelling. Indeed, Rodet (1993) studied musical applications of Chua's circuit early on. Chua's circuit is a strikingly simple three dimensional ODE, which has also been implemented in electronic circuits, and one of the most deeply studied chaotic systems (Zeraoulia and Sprott, 2010). The most crucial part of Rodet's work is to modify the original Chua's circuit by introducing a delay line in the feedback path.

Among the earliest proposals for uses of chaotic maps in sound synthesis are those of Truax (1990). He suggested using e.g. the logistic map to control the instantaneous frequency of an oscillator. Here the rate at which the frequency is changed is very

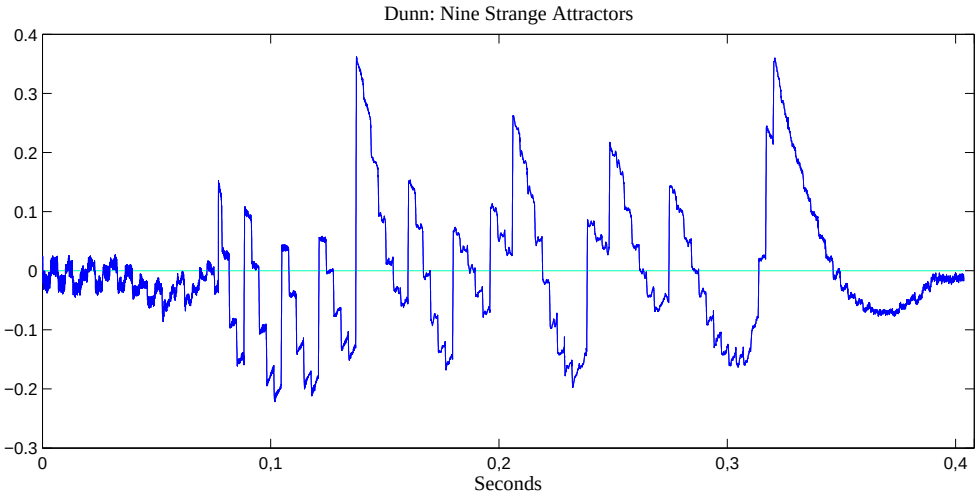

Figure 4.3: An example of a typical "staircase" waveform. From David Dunn's Nine Strange Attractors, at 16'18.5 seconds, left channel.

important; it may happen at the scale of typical period lengths. Another application suggested by Truax is to use granular synthesis and map the chaotic time series to the grain's frequency. These techniques may surely be more worthwhile strategies than using low-dimensional maps directly as streams of samples. At the same time, Di Scipio was applying chaotic maps to both instrumental composition and sound synthesis, for instance by sound granulation where the value of the chaotic map is taken as a time index into the sound file (Di Scipio, 1990).

David Dunn used several chaotic systems in his piece *Nine Strange Attractors* (2006). Both at the sample level and on a higher level, the parameters of these systems were also controlled by chaotic systems. From the composer's programme notes (Dunn, 2007, liner notes), we learn that: "Each attractor's unique spatial morphology is driven at various computational rates in real-time, ranging from those associated with its traditional high-resolution visual display to extremely low rates that reveal greater structural detail through sound."

Little information apart from this and a listing of the attractors used—including Duffing, Lorenz, Rössler, van der Pol and a few lesser known ones—is available on the construction of this piece, but one observation can immediately be made: Such chaotic systems as are cited as sources for this piece have a distinct timbre when their time series are used as audio signals, and smooth parameter changes are also likely to produce smooth changes of the waveform (and spectrum) everywhere except at bifurcation points. The systems used are differential equations, which should produce relatively smooth curves (as compared to iterated maps). In Dunn's piece, however, there are frequently occurring jagged pitch profiles with sudden changes, which is untypical of smoothly varied parameters of flows. The waveform is full of what would be regarded as artefacts in acoustic recordings and removed in a mastering session, notably sharp discontinuities that cause audible clicks, and DC offsets. A gritty sound quality as of quantisation noise is pervasive through much of the piece, as can also be seen from the waveform with its

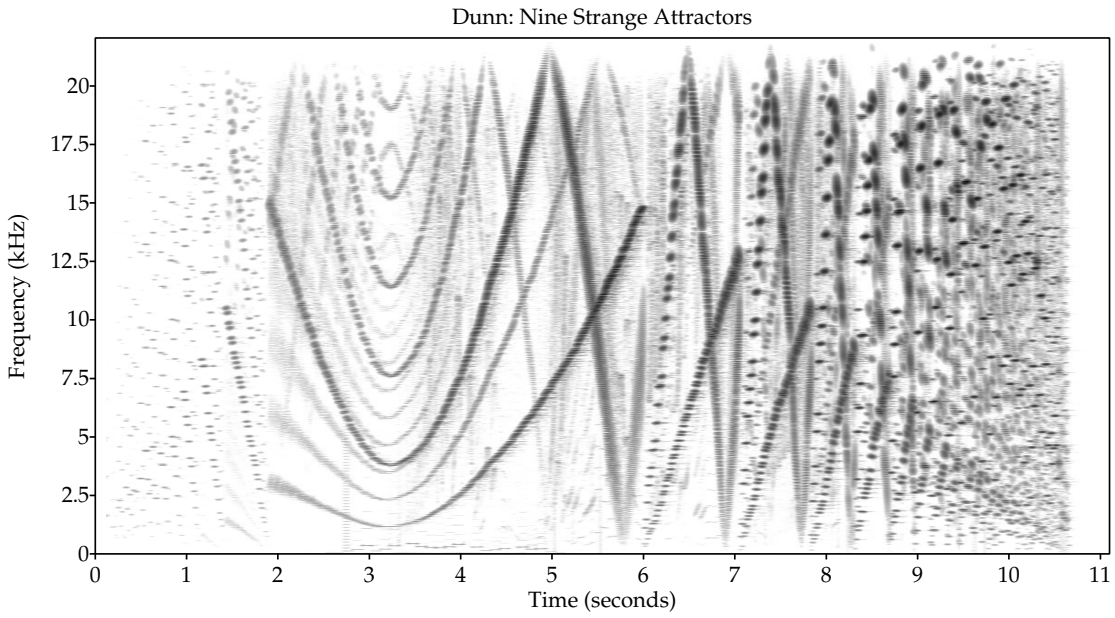

Figure 4.4: Sweeping glissandi in Dunn's *Nine Strange Attractors*, sonogram from 9'52.5, right channel.

staircase appearance (Figure 4.3). Glissandi and granular textures are also typical; they occur in sweeping gestures, as seen in Figure 4.4. It appears as though Dunn made little, if any, attempts at interfering with the algorithmic process that generated the piece; hence, it probably qualifies as an example of music realised with an autonomous instrument.

Di Scipio (2001) describes a synthesis method called *iterated nonlinear functions*, which is actually a regular iterated map, only used in a slightly different way than one would most likely think of. Given a function $x' = f(x, r)$, with a parameter r and an initial value x_0, the map is iterated a (fixed) number of times, n, to produce the output $f^n(x_0, r)$. Instead of using the successive iterations of the map, either the initial value or the parameter is varied, and for each such new condition an output sample is produced. What this procedure amounts to then, is a kind of scanning of the phase space, or the set of initial conditions, which might belong to different basins of attraction. But there is a difference, since Di Scipio advises that the number of iterations be kept low, otherwise the result is most likely broadband noise. On the contrary, when plotting basins of attraction or bifurcation diagrams, the number of iterations is set very high to avoid the effect of transients.

Ikeshiro (2010) used the Lorenz system to generate the piece *Construction in Self*. The Lorenz system is used on three levels, from signal level, through control signals to a meta level on which different synthesis techniques (all based on the Lorenz system) are chosen and the length of form sections are determined. An original technique used by Ikeshiro is to generate two orbits from slightly different initial conditions, and to use their difference as the audio signal. For parameter values corresponding to a fixed point, there will be an initial transient which depends on the initial condition; this is efficiently used

for a variety of percussive sounds. *Construction in Self* is described as a deterministic system, capable of generating the same output each time, although Ikeshiro sets different initial conditions for each performance, thus resulting in a different piece each time. This is a good illustration of how chaotic systems may function as autonomous instruments.

The list of composers and musicians who have used chaos for notated music or sound synthesis could go on, and a few more examples will be mentioned later. Some advice on sonification may however be in order, before we end this overview of chaos in sound synthesis.

Sonification is an alternative way to obtain an intuitive understanding of maps instead of looking at bifurcation plots. Functions that map from an interval into itself are easy to use; the variable is simply scaled to a suitable amplitude range and taken as successive sound samples. Period doubling bifurcations and chaotic regimes are typical for maps, and sweeping the parameter through its range produces typical and easily recognised sounds. As a two-cycle corresponds to an oscillation on the Nyquist frequency, a lower than usual sampling frequency (e.g. 16 kHz) will ensure its audibility. Correspondences between a bifurcation plot and a sonification of a map are obvious: fixed points result in silence, periodic cycles typically produce high pitched sounds, and chaotic regimes result in white or coloured noise. These phenomena arise in maps of very different appearance.

4.3 Maps with filters

Iterated maps easily produce very irregular orbits. Linear filters, specifically lowpass filters, can be used to smear out any irregularities in a signal. If flows have the advantage for sound synthesis that they produce smooth orbits, it should be interesting to combine chaotic maps with smoothing filters in their feedback path. Some examples of this method are presented in this section. Actually, such filtered maps may result in anything from waveguide physical models to nonstandard synthesis models that do not even remotely resemble any acoustic instrument.

First, the theory of nonlinear filters and maps with filters is presented, and we show that feature extractors are nonlinear filters. Some simple, low-dimensional filtered maps are then introduced. The filtered maps are in a sense simplified models of feature-feedback systems, which means that they may exhibit some similar dynamics, at least in qualitative terms. Thus, nonlinear filters and filtered maps provide an essential theoretical background for feature-feedback systems.

4.3.1 Nonlinear filters

Nonlinear processing of audio signals includes the modelling of distortion effects, a problem that continues to attract much attention. A simple technique is to lowpass filter the input signal and apply waveshaping with polynomial functions (Schattschneider and Zölzer, 1999; Zölzer, 2002). The lowpass filters are used to eliminate potential aliasing problems. Such models of distortion effects are examples of nonlinear filters.

While the theory of linear filters is well known, there is hardly a single coherent theory of nonlinear filters. Any operation on signals that fails to respect the linearity criteria qualifies as a nonlinear filter. Thus, when any of the two criteria

$$\begin{array}{rcl} \mathscr{L}(\alpha x) & = & \alpha\mathscr{L}(x) \\ \mathscr{L}(x+y) & = & \mathscr{L}(x)+\mathscr{L}(y) \end{array} \tag{4.4}$$

are violated for signals x and y and a scalar α, the system $\mathscr{L}$ is nonlinear. Memoryless nonlinearities may be considered as trivial filters, but in general we shall consider filters with delayed variables. Then, a causal time-invariant recursive nonlinear filter is described by

$$y_n = f(x_n,\, x_{n-1},\, \ldots,\, x_{n-M},\, y_{n-1},\, y_{n-2},\, \ldots,\, y_{n-N}) \tag{4.5}$$

for some nonlinear function f. Non-recursive filters are obtained by skipping all feedback terms. Extensions to non-causal and time-varying filters are straightforward, but not necessary in this context. With a sufficiently broad conception of what a function is, even median filters can be understood as cases of (4.5). In the case of median filters the function consists of a sorting procedure. If the nonlinear function in (4.5) is a suitable polynomial or some clipping function such as tanh, some interesting distortion effects may be designed (Holopainen, 2007).

Now we turn to feature extractors and show in what sense they are are nonlinear filters. The Fourier transform in all its variants is a linear operator, whereas the calculation of the spectral centroid is not. If it were, changing the amplitude of the signal would influence the centroid position, according to the first linearity criterion. Such robustness against irrelevant influences is of course highly desirable for feature extractors.

The RMS amplitude is important on its own, but it also occurs frequently as a component of other feature extractors. It is easy to see that it is also a nonlinear filter. After squaring the input signal (nonlinear operation), it is passed through a moving average filter, which is just a (linear) lowpass filter, then the square root is taken (second nonlinearity).

The discussion of feature extraction could be carried on further, but the above examples should make it clear that the kind of filters actually used are non-recursive and time-invariant. As always, the requirements of live processing dictates that the filters be causal, although in offline analysis non-causal filters are an option. In short, feature extractors of length L have the general form

$$y_n = f(x_n, \ldots, x_{n-L+1})$$

since there is usually no reason to feed back the extracted feature into this filter. All feature extractors that we have use for reduce the information content in the original signal in some way. In particular, the output of most feature extractors usually have a lower bandwidth than the analysed signal. Onset detectors that output a spike whenever an attack is detected and zero otherwise are an exception to this rule, but for other feature extractors, their output bandwidth will be proportional to f_s/L.

There is no requirement that the parameters of the feature extractor be constant over time; it is more of a practical matter that they do not change. For example, an implementation of RMS analysis with a first-order recursive filter makes it easy to vary the integration time.

Dobson and ffitch (1996) introduced a nonlinear filter intended for use as an audio effect:

$$y_n = x_n + ay_{n-1} + by_{n-2} + dy_{n-L}^2 - C \tag{4.6}$$

Here, the delay of L samples acts as a kind of comb filter combined with waveshaping by squaring, whereas the other two delayed terms taken on their own can be understood as forming a second order IIR filter with coefficients a and b (that is, it becomes a regular linear IIR filter provided $d = C = 0$). In order to study this filter as a dynamical system, the input signal could first be restricted to a unit impulse. If the delay L is long, rewriting as a delayless system is possible, but looks a bit redundant. The stability of this filter is a great problem for the user, as one cannot immediately see which parameter values will cause instability. A simple and crude solution is to wrap the right hand side of (4.6) in a limiting function such as tanh, but this does alter the sound of the filter. Alternatively, an envelope follower may be applied to y_n, and a variable gain inversely related to the amplitude applied to either some or all the filter terms (Holopainen, 2007).

The nonlinear filter of Dobson and ffitch shows how closely related such systems are to filtered maps. The distinction reduces to the question of whether there is an input signal or not.

4.3.2 Smoothed maps

There are dynamical systems that could be thought of as maps with delay, or as maps combined with causal linear time invariant filters. It is highly interesting to note that at least a limited class of these systems has been studied as a prototype for physical modelling (Rodet and Vergez, 1999b,a), which will be discussed in greater detail below. But to begin with, we will discuss some of the simplest possible systems, and introduce some theoretic notions along the way.

If a filter is inserted in the feedback loop of an iterated map, the dynamics may change profoundly. When the filter has a lowpass character, these systems will be referred to as *smoothed maps*.

Smoothed maps seem to be particularly useful for acquiring a better understanding of feature-feedback systems. The reason for this is that a typical feature extractor acts as a smoothing filter. An iterated map $x_{n+1} = f(x_n)$ can be turned into a filtered map by substituting the argument to the map with the output y_n of a filter as follows:

$$\begin{aligned} x_{n+1} &= f(y_n) \\ y_n &= \sum_{k=0}^{M} a_k x_{n-k} - \sum_{k=1}^{N} b_k y_{n-k} \end{aligned} \tag{4.7}$$

By changing filter characteristics from lowpass to bandpass, highpass or other types, the dynamics may be completely different.

Maps with time delayed variables are usually transformed into delayless maps with new variables for the delayed variables. Consider as an example the logistic equation with a first order FIR filter, studied in detail elsewhere (Holopainen, 2011):

$$\begin{aligned} x_{n+1} &= r(y_n - y_n^2) \\ y_n &= ax_n + (1-a)x_{n-1} \end{aligned} \tag{4.8}$$

For $a \in (0, 1)$ the filter is lowpass, and most strongly so when $a = 1/2$. Notice that the logistic map is recovered when $a = 1$, and $a = 0$ corresponds to a system with pure delay. The ordinary logistic map is unstable for $r > 4$, but for a range of filter coefficient values around $a = 1/2$, the parameter r may be increased significantly above 4 without losing stability (See Figure 4.5). For some parameter values, such as $a = 0, r = 4$, the map is hyperchaotic.

If a recursive, second-order lowpass filter is substituted for the two-point moving average filter in eq. 4.8, the dynamics of the system collapses to a fixed point for very large parts of the parameter space, and the regions where any oscillations occur are smaller and restricted to higher values of r. Similar results are obtained with a one-pole lowpass filter; fixed points are reached over broad areas, although not to the same extent as with the second-order filter. In that case, the system becomes

$$\begin{aligned} x_{n+1} &= f(y_n) \\ y_n &= x_n + by_{n-1}. \end{aligned}$$

A one-pole filter is a better approximation than (4.8) to the smoothing that typically goes on in feature extractors. For a fair comparison with eq. 4.8, the one-pole filter should take the form $y_n = bx_n + (1-b)y_{n-1}$. This system shares some traits with the first-order FIR system, such as increasing the range of stable fixed point solutions.

4.3.3 Plotting bifurcation diagrams

A useful tool for investigating dynamical systems is the bifurcation plot or diagram. In one dimension, the orbit is plotted against some control parameter. If there are two control parameters that can vary simultaneously, the period length can be colour coded and plotted. Higher dimensional spaces of control parameters of course pose problems for direct visualisation, but two-dimensional slices can still be plotted by fixing values of the remaining parameters. An algorithm for calculation of the period length is also needed. Here we use the average magnitude difference function (Ross et al., 1974), defined as

$$D_\tau = \frac{1}{L} \sum_{n=1}^{L} |x_n - x_{n-\tau}| \,. \tag{4.9}$$

After the transient has died out, a periodic orbit will be easy to detect this way; the sign to look for is the first minimum of D_τ, which should be very close to zero. Candidates for chaotic orbits show up as distributions of D_τ with no deep minima. Such a signature may also indicate quasi-periodicity, so a second test is needed to see whether the orbit has a positive Lyapunov exponent or not, i.e. whether two infinitesimally separated initial conditions diverge exponentially over time.

Figure 4.5, showing a bifurcation diagram of eq. 4.8, is plotted using this method. Apart from period doublings and chaotic regions, the region of stability may be of great interest. For this map, however, its stability depends on the initial condition.

Another approach to characterising orbits is the box counting algorithm. First, the amplitude range is divided into a finite number of bins. As the signal's amplitude falls within one of these bins, a counter is incremented. Finally, the number of occupied bins is counted. This too gives a rough indication of the periodicity, at least for low periods. Note, however, that closely spaced points may erroneously fall into the same bin, hence an underestimation of the period length is likely. A worthwhile computational improvement is to observe the orbits amplitude range first, and then adjust the bin spacing so that it exactly covers this interval. This is the first step of an implementation of the box counting algorithm, by which a fractal dimension may be estimated (see Section 4.1.4). Further, the statistical amplitude distribution of an orbit, also known as the natural distribution (Tél and Gruiz, 2006), can be obtained by this technique. This is not so informative if the map is applied for sample level sound synthesis, but if used on a higher level, for instance to determine a sequence of pitches, the distribution has clear perceptual implications. At least the study by Gregson and Harvey (1992) indicates that pitch sequences with flat probability distributions can be distinguished from chaotic sequences by some listeners.

Period length can be calculated directly from the map in the following way. Fixed points x^* (i.e. orbits of period one) satisfy $x_n - x_{n-1} = 0$, that is, $f(x^*) = x^*$. Similarly, for period p the orbits are given by

$$f^p(x^*) = x^* \tag{4.10}$$

which is the p-fold application of the map to a point. As can be seen, longer periods, $q > p$, also satisfy eq. 4.10 if p divides q, so the period length is the lowest p for which eq. 4.10 holds. In practice, this is too complicated to calculate directly, especially so for maps of any degree of complication, which is why we resort to numerical methods.

There is a well-known theorem due to Li and Yorke (1975), stating that "period three implies chaos". This was proved in a more general way a decade earlier by Sharkovskii (for the proof of the theorem, see Elaydi (2008)). If a continuous map on the interval has a period 3 orbit, then this implies that it has orbits of all integer periods *at the same parameter value*. These periods are unstable, however, and will not generally be observed when iterating the map on a computer. Hence this theorem is mostly of theoretical value.

Bifurcation plots can be generalised in ways that make them even more useful for investigations of unknown synthesis models. Instead of just plotting the orbit's period, feature extractors may be used to chart the parameter space in perceptually grounded terms. This strategy will be much used in Chapter 6.

4.3.4 Feedback FM with filters

As a second example of a dynamic system with filters we consider feedback FM with a single oscillator. A benefit of feedback FM over ordinary FM is that the modulation index β is better correlated with spectral brilliance. Again we use the simple first order FIR filter, and end up with the system

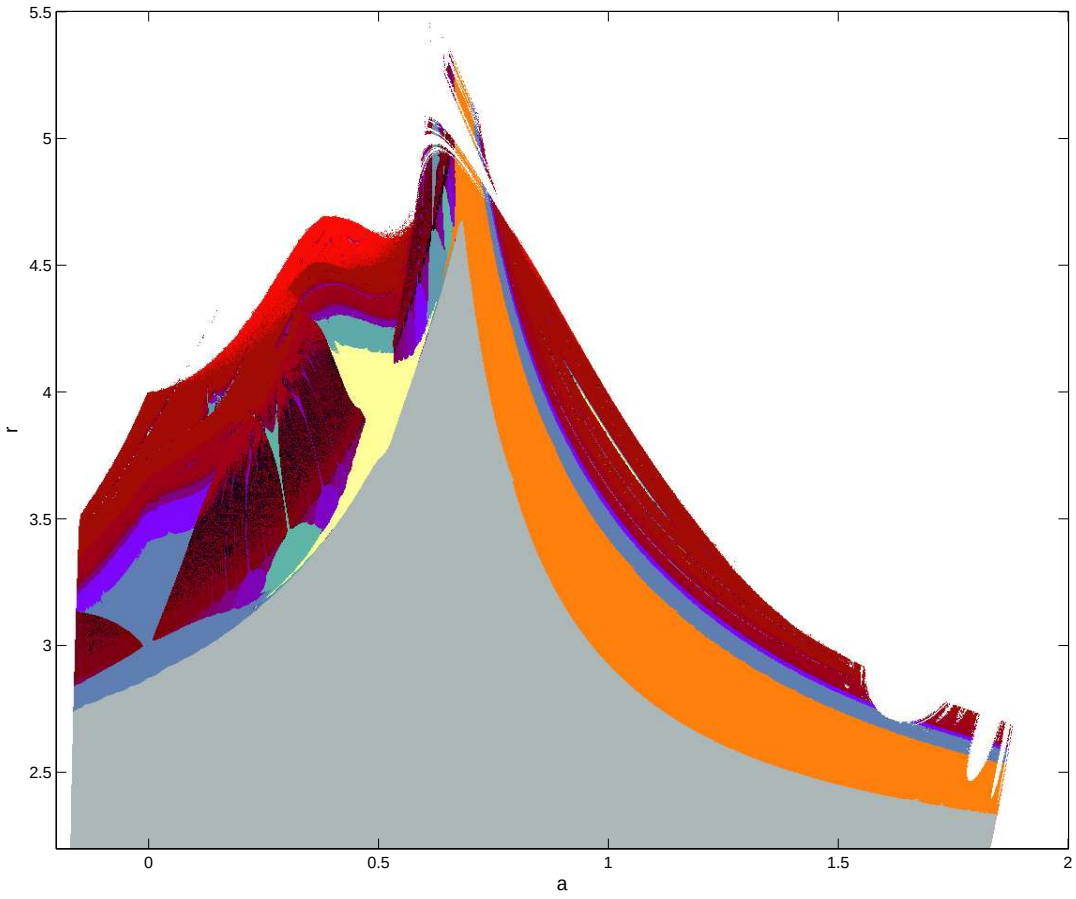

Figure 4.5: Bifurcation plot of the filtered logistic map. Period 1 is plotted in gray, period two in orange, period 3 in light yellow and period four in light blue. Higher periods are shown in various shades of blue and green; black indicates quasi-periodicity. Red and brown indicates (various degrees of) chaos, and white is unstable.

$$\begin{aligned} x_{n+1} &= \sin(\theta_n + \beta y_n) \\ \theta_{n+1} &= \theta_n + \omega_c \\ y_n &= a x_n + (1-a) x_{n-1}. \end{aligned} \tag{4.11}$$

For $a = 1$ the system reduces to standard feedback FM. With modulation index $\beta < 2$ a pitched sound is produced, while for higher modulation indices a noisy quality becomes increasingly prominent. If the filter coefficient is now decreased towards $a \approx 0.5$, the smooth pitched sound is recovered even for a higher modulation index. Thus, the filter has a stabilising function. As simple as this model is, it could be a useful building block for an instrument.

Although this system is quite different from the filtered logistic map, there are some surprising similarities of its bifurcation diagram in the particular case of no driving, i.e.

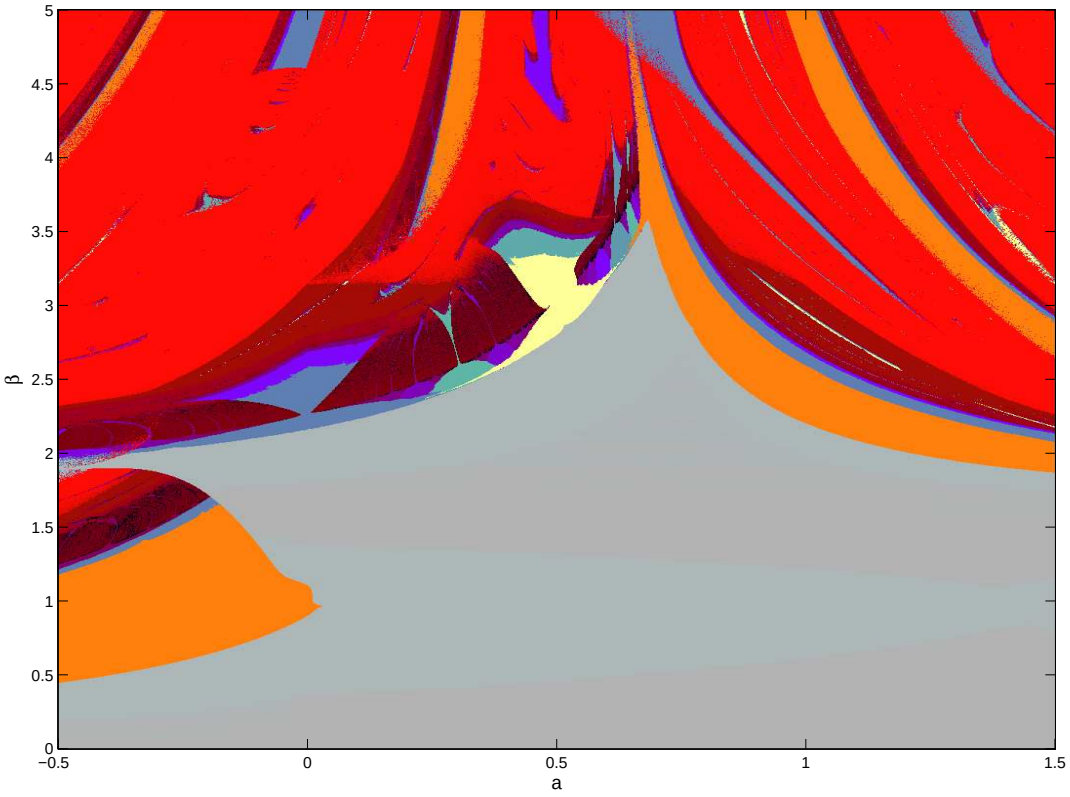

Figure 4.6: Feedback FM with first order FIR filter and no driving. Notice the similarities with the filtered logistic map, particularly in the middle of the plot.

$\omega_c = 0$ (see Figure 4.6). For oscillations to start up at all, the initial condition has to be chosen so that $\theta_0 + \beta y_0 \,(\mathrm{mod} 2\pi) \neq 0$. If these oscillations at zero driving frequency are not what one would expect to hear in feedback FM, it may be because the initial conditions that would allow such oscillations are not used. When non-zero driving is introduced, the structure of the bifurcation diagram changes. It does not produce as neat pictures (the plot roughly divides into a quasi-periodic lower region and a chaotic upper region), but instead it has the musically useful property of giving the pitch one asks for. The similarities that can be seen by comparing the bifurcation plots of the filtered logistic map and the above system are particularly clear at $a = 0$ and $a = 1$. But in contrast to the logistic map, the feedback FM system cannot become unstable. This is good news for sound synthesis, where possible instability has to be taken care of.

The filter in (4.11) corresponds to a two point moving average when $a = 0.5$. If successively longer moving average filters are used, the noise suppressing effect becomes more prominent as the length increases, although there is also a side effect, namely a ringing sound.

As longer moving average filters are used, the effect of sweeping the modulation index becomes clearer: there is a region of high modulation index corresponding to sounds that appear as comb filtered noise, and a region of low modulation index where it sounds more

like ordinary feedback FM. While the sounds of the extremes of the parameter ranges could perhaps be produced by simpler models, the *transition* between the two regions is what makes this particular model stand out. In particular, there are occurrences of subharmonics of the carrier frequency at the border between the two regimes. From a musician's perspective, it would be useful to be able to vary both the filter length and modulation index continuously, but then a time-variable delay line with a good interpolation scheme will be needed. Other filters may however be substituted in this model with interesting results.

Example 4.1. A biquad lowpass filter extends the range of the modulation index for which the filtered feedback FM system does not become noisy. It also extends the timbral range in an interesting way and allows continuous adjustments of the filter's cutoff frequency. Here, the cutoff frequency is periodically modulated in the range $200 - 1200$ Hz. A cascade of subharmonic oscillations occurs as the modulation index simultaneously increases linearly in the range $1 \leq \beta \leq 25$ over the sound's duration.

4.3.5 Feedback FM with pure delay

Delay differential equations are known to be capable of very complex and unstable behaviour. These are systems of the form $\dot{x} = f(x(t), x(t-\tau))$. Delay differential equations have most notably been used to model biological systems, such as the production of white blood cells (Mackey and Glass, 1977). Several simple delay differential equations with chaotic solutions have later been found (Sprott, 2010). In essence, what happens is that a finite delay time τ turns the system into an infinite dimensional system, because to specify the initial state, a continuous line segment of length τ with an infinite number of points needs to be provided. Similar observations can be made concerning delayed maps, except that the system's dimension remains finite.

If we insert a delay of D samples into a feedback FM oscillator, we have

$$\begin{aligned} x_n &= \sin(\phi_n + \beta x_{n-D}) \\ \phi_n &= \phi_{n-1} + \frac{2\pi}{f_s} f_c. \end{aligned} \tag{4.12}$$

Here the D initial values of x may be assumed to be zero. When the delay time is sufficiently long, say, on the order of tens of milliseconds, the modulation enters in a staggering way, much like the delay builds up in a feedback comb filter. This effect is more pronounced at a high modulation index; it is also more noticeable when the index is gradually decreasing rather than increasing. In regular, non-delay feedback FM (in fact, delayed by one sample), the carrier may have a time variable frequency without this causing any conspicuous effects. On the contrary, in delayed feedback FM, if the carrier frequency changes substantially during the delay interval, the result is inharmonic modulation and a very sudden increase in densely spaced sidebands. So, allowing the carrier frequency to change, the likelihood of obtaining noisy sounds increases rapidly as the delay time is lengthened.

Example 4.2. The staggering entry of the delayed feedback is first heard as resonant frequencies, then as an echo as the delay is increased in six stages from one sample to 0.2 seconds. For each delay length, the carrier is swept over one octave. The modulation index $\beta = 1$ has been used throughout.

Even if the carrier is kept constant, there is a risk of a build-up of inharmonic frequency components due to aliasing. This may happen whenever the carrier is not a divisor of the sampling frequency.

Although this example of FM with delayed feedback is prone to produce noise (and hence chaos), delayed feedback has also been used as a mechanism to suppress chaos (see Section 4.5.2), although in that case, both the system and the feedback are of another form. Pure delay in the feedback path is analogous to an IIR comb filter, and different from the smoothed maps where the feedback is a time averaged version of the previous output. This intuitively explains some of the qualitative differences between such systems.

4.3.6 A link to physical modelling

Rodet and Vergez (1999b) described a general framework that can be used for physical models of musical instruments. This model corresponds to a special form of filtered maps (or flows), where the filter includes a delay. The model is

$$x(t) = h * g(x(t - \tau)) \tag{4.13}$$

with delay time τ, a nonlinear function g, and convolution with the filter impulse response h. This system is not restricted to one-dimensional variables. As an example of nonlinear function, Rodet and Vergez considered Chua's circuit.

For musical uses, it will be important to have control over time-variable parameters. The nonlinear function can be parameterised, but even more important is the use of time variable filtering. Varying the delay time by large amounts is hard to do without introducing artefacts such as clicks. Nevertheless, the other part of the linear filter may be put together of first- and second-order IIR links, to form arbitrarily complex frequency responses that are easily varied over time. Now, it turns out that by introducing variable resonances in a system of this type, one can produce sounds clearly reminiscent of electroacoustic feedback, such as the squealing tones of feedback from an electric guitar and amplifier. As the resonance frequencies shift, the pitch also changes.

Let us now consider a model along the lines of (4.13), which is capable of producing clarinet-like timbres, or at least something reminiscent of woodwinds, yet also capable of sounding distinctly unlike any acoustic instrument. Inasmuch as the delay length corresponds to a fingering producing a certain effective tube length in a woodwind instrument, it is desirable to make the delay time variable. Sudden changes of delay length in just one delay line are impractical because this will tend to introduce clicks and other artefacts. Instead, several parallel delay lines may be used. In an efficient implementation, only one delay line would be active at any time and need to be computed. Instead of that, we illustrate the principles with a computationally inefficient scheme, which however is less complicated. This leads to a somewhat inflexible model, excluding the possibility of smooth transitions between delay lengths, but with a set of discrete delay lengths at one's

disposal. Switching is carried out by processing all delay lines in parallel, and multiplying them with an amplitude vector a_k, such that only the chosen delay line is multiplied with 1, while the others have their amplitude set to 0. To obviate discontinuities at the switching instants the amplitude function can be smoothed, e.g. with a moving average filter L_k.

With N separate delays and bandpass filters B_k tuned to frequencies that are inverses of the corresponding delay lengths, and amplitude functions A, the complete system becomes

$$x[n] = f(y[n]) \tag{4.14}$$

$$y[n] = \sum_{k=1}^{N} A_k[n] \cdot B_k * x[n - D_k] \tag{4.15}$$

where the amplitudes A_k are determined by activating delay K,

$$a_k[n] = \begin{cases} 1, & \text{if } k = K \\ 0, & \text{else} \end{cases} \tag{4.16}$$

and further smoothing

$$A_k = L_k * a_k[n]$$

with the filters L. For the nonlinearity, we use the one-parameter function

$$f(x, \mu) = \mu(1 - 2x^2) + (1 - \mu)x \tag{4.17}$$

which appears to be most useful in the range $0.9 < \mu < 1$. Further processing will be needed, such as removing the DC level and additional filtering to shape the timbre in suitable ways. Interesting transitions occur when one delay line is opened and the others closed. Further, various balances between the delay lines could be exploited. While sonically promising, this model is, as mentioned, not very efficient if a great number of delay lines should be needed.

Example 4.3. By adding noise to eq. 4.17 in amounts inversely proportional to μ, the instrument can be made to produce whispering sounds which are still pitched but having an "aspirated" quality. As μ diminishes during the course of this tune, the effect is similar to reducing blowing pressure in a wind instrument.

Overblowing wind instruments causes them to produce a higher harmonic; in clarinets only odd harmonics are possible. The same phenomenon arises in this model, though it is not evident how to control it.

4.4 Nonlinear oscillators by maps and flows

Since almost periodic oscillation is typical for many musical instruments, ODEs that exhibit similar behaviour are good candidates for synthesis models. There are many

two-dimensional ODEs that have bounded periodic solutions, but their period may have a nonlinear dependence on the control parameters. There is an easy solution which takes advantage of a pitch follower and a simple adaptive scheme, as will be described below. Circle maps and the standard map, being discrete time counterparts of nonlinear oscillators, can also be used for sound synthesis. However, the sharp transitions in their parameter spaces make them harder to navigate.

Nick Collins (2008b) proposed a few nonlinear oscillators and other exotic techniques under the heading of "errant sound synthesis", which are implemented as unit generators for SuperCollider. Apart from being capable of chaos if forced by a periodic oscillation, nonlinear oscillators have some other interesting features such as variable waveform depending on some bifurcation parameter. Chua's circuit has been extensively studied, and also proposed as a musical instrument by Mayer-Kress et al. (1993). They found it capable of producing a bassoon-like timbre, displaying "period adding" or subharmonic bifurcations. Percussive sounds may be obtained by setting the parameters to values that yield a fixed point, and using the transient from some initial condition. Many chaotic systems, apart from Chua's circuit, can be realised as electronic circuits (Sprott, 2010). Such circuits may sound-wise be similar to what can be done with analogue synthesisers, which is why the idea of using ODEs for sound synthesis is not so far-fetched.

4.4.1 The circle map

Circle maps are iterated mappings defined on an interval with its endpoints connected, topologically equivalent to a circle. In their most general form, circle maps are given by equations of the form $x_{n+1} = x_n + \Omega - Kf(x_n)$, where $\Omega \in [0, 1]$ is a normalised frequency variable, f is an arbitrary periodic function $f(x) = f(x + 1)$ and K is the degree of nonlinear perturbation. To be specific, we will only consider the sine circle map

$$x_{n+1} = x_n + \Omega - \frac{K}{2\pi} \sin(2\pi x_n) \quad (\mathrm{mod} 1). \tag{4.18}$$

The variable of this map may suitably be interpreted as the phase of an oscillator, such as $y_n = \sin(2\pi x_n)$.

Direct sample level synthesis with the circle map was considered by Essl (2006a,b), including variations of the nonlinear perturbation which here is taken to be the sine function. More specifically, Essl used a low sampling rate (22050 Hz), and the signal was further downsampled by a factor of ten. This is significant, since iterated maps often produce oscillations of short period, or spectra with much high frequency content. It will be useful to lower the frequency range somehow into a range that is more suitable for sound synthesis.

Figure 4.4.1 shows a bifurcation plot of the circle map for $\Omega = 0.04$ and $-8 \leq K \leq 8$. The rectangle in the upper half and in the middle (around $K = 0$) corresponds to a region of pitched sounds. As the map is iterated, trajectories rapidly approach some attractor. This can clearly be seen for orbits of periods one, two and four, where the transients are plotted in green.

The two-dimensional parameter space $\{\Omega, K\}$ is not so easy to navigate if we use it as a sound synthesis algorithm. This space is full of fixed point attractors that result

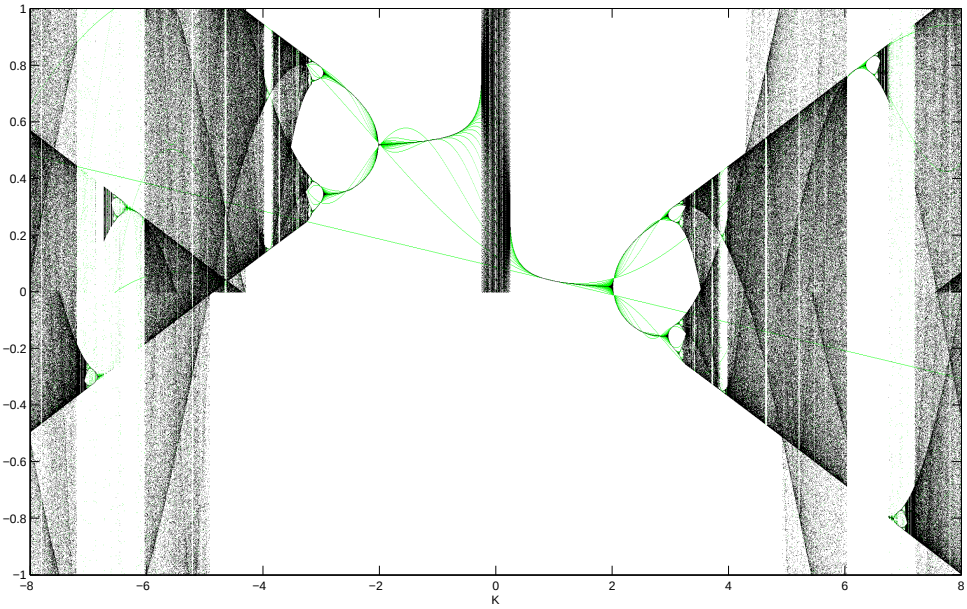

Figure 4.7: Circle map with $\Omega = 0.04$ and $-8 \leq K \leq 8$. Transients (the first 40 iterations) are plotted in green.

in silence interspersed with period doubling bifurcations and chaos. Moreover, there are sudden transitions from fixed points to chaos as the parameters are slightly varied which means that smooth changes of the parameters will sometimes result in stark contrasts. The transitions from a fixed point to chaos is an example of a *crisis* phenomenon which can be seen at $K = \pm 6$ in the figure; another abrupt change happens near $K = 0$ where a low pitched oscillation suddenly turns into a fixed point.

Circle maps are described in more detail in the literature (e.g. Frøyland, 1992; Glazier and Libchaber, 1988; Hao and Zheng, 1998); here we briefly summarise some of the most important ideas. The *lift* of a circle map is defined on the real line; it is obtained by taking the circle map without the modulo operation. *Winding numbers* (or rotation numbers) is a way to quantify the average rotation of a map, defined as

$$w = \lim_{n \to \infty} x_n / n \tag{4.19}$$

provided the limit exists, and where x_n is found by iterating the lift of (4.18). For $K = 0$, the driving frequency Ω coincides exactly with the winding number, but for increasing nonlinear coupling, there are intervals of the driving frequency that have the same winding number. This is known as frequency locking (or mode locking). As can be seen in Figure 4.8, there are quite broad intervals of frequency locking near $\Omega = 0$ and $\Omega = 1$, with a slightly narrower interval existing around $\Omega = 0.5$. In fact, it is well known that the interval lengths of frequency locking depend on the degree of "rationality" of K. The structure of frequency locking is in fact a fractal, known as the Devil's staircase. Similar patterns of frequency locking occurs in many systems; there is even evidence of

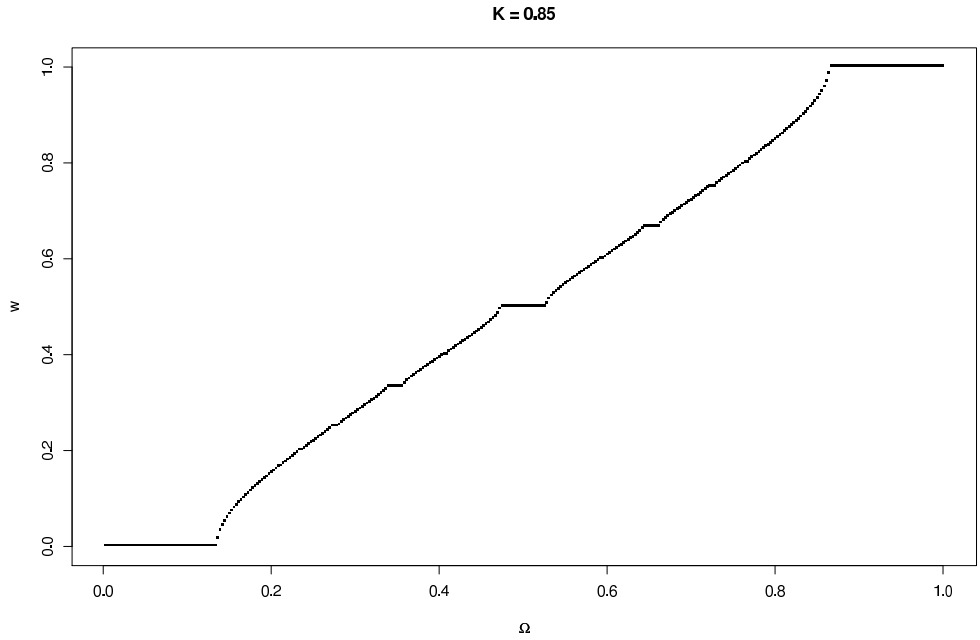

Figure 4.8: Winding numbers of the circle map at $K = 0.85$.

its presence in one of the feature-feedback systems.

Supposing the circle map is not chaotic, then if the winding number is a rational number p/q, the map will be q-periodic, whereas if it is irrational, the map will be quasi-periodic. In general, the circle map (4.18) is not monotonic on the interval $[0, 1]$, but its lift *is* monotonic for $0 \leq K \leq 1$. Above $K = 1$ the map has non-monotonic lift, and some qualitatively different behaviours can be observed. Irrational winding numbers exist for $0 < K < 1$, but only restricted to a line of the parameter space. As K increases beyond 1, these lines widen into wedge-shaped regions called Arnol'd tongues. Thus, the increased nonlinear coupling makes quasi-periodic orbits more robust.

Frequency locking is by no means limited to the circle map. In particular, it can occur in some synthesis models such as variants of feedback FM. Obviously, frequency locking is undesirable whenever a simple scheme of pitch control is desired; it means that the produced pitch increases in a highly irregular manner as a function of K until the system becomes chaotic and thus unpitched.

4.4.2 Chirikov's standard map

A two-dimensional map that models a rotating point is particularly apt for direct use in sound synthesis. The standard map is also called a kicked rotator, with spatial variable θ giving the angular position on a circle, and velocity variable v, representing the instantaneous speed of the point:

$$\begin{aligned} v_{n+1} &= v_n + K \sin \theta_n \\ \theta_{n+1} &= \theta_n + v_{n+1} \end{aligned} \tag{4.20}$$

Here, both variables are taken modulo 2π. In this form it is a conservative system, i.e., the area of an arbitrary small region of its phase space does not change under the mapping.

A more technical way to say the same thing is that the determinant of the Jacobian of a conservative system has absolute value 1 everywhere in phase space. If a parameter were to be introduced, such as $\theta_{n+1} = r\theta_n + v_{n+1}$ in the second line of (4.20), then any value $|r| < 1$ would turn this into a dissipative system. In that sense, conservative systems are not robust to small parameter perturbations.

Orbiting planets, or the rings of Saturn, are examples of physical systems that have been modelled as conservative systems (Frøyland, 1992). Conservative systems (as opposed to dissipative systems) are characterised by the absence of friction, and there is no attractor nor repellor in the phase space, and motion forwards and backwards in time is equivalent. In fact, this can be demonstrated to some degree by numerical simulation. Take a set of points in phase space, say a small square. If this set of points is mapped forward in time a limited number of steps, and then the same set of points is mapped backwards in time the same number of steps, the original arrangement of points is recovered. However, this only works for a limited number of iterations; doing the same experiment with a higher number of iterations will cause the original small square to be dissolved over the entire phase space (Tél and Gruiz, 2006). Thus, any mention of time reversibility of maps (where this makes sense) is more of a mathematical idealisation than a practical and observable phenomenon.

Conservative systems like the standard map have no attractors, but they have a particular form of sensitivity on the initial condition. Depending on the initial condition, the orbit may be either chaotic or regular (periodic). This is of great significance if conservative systems are to be used for sound synthesis or algorithmic composition. Not only useful parameter values have to be known, but suitable initial conditions must also be found.

The Chirikov standard map is interesting for use in sound synthesis. Since both variables are periodic, either of them may be used as the phase variable in an oscillator. The standard map is easily extended to a driven system, which makes it similar to feedback FM. Filters may also be inserted on both variables, to construct smoothed maps. Bandpass filters are particularly valuable for tuning the system, in a literal sense, to make it produce sounds of precisely controllable pitch. Such compounded systems—the standard map with at least two linear filters and an optional driving (modulating) oscillator expose a sufficiently rich variety of sound to grant a detailed study as a feature-feedback system, as will be done in Chapter 6.

4.4.3 Nonlinear oscillators with pitch correction

Two-dimensional flows are a wellspring of under-explored sound generators. Although it is common wisdom that three dimensions are required for chaos in flows, there are two-dimensional systems with singularities that exhibit some chaos-like irregular dynamics and are sensitive to small amounts of noise including numerical round-off error (Sprott, 2010). The two-dimensional flows we will consider are however familiar smooth systems that can only become chaotic by introducing a driving term or by other means increasing their dimension. Nonlinear oscillators are a rich source of timbrally varied sounds, although it comes at the cost of increased difficulty in controlling the pitch. Complicated mathematical techniques exist for approximations of the period of nonlinear oscillators

(Strogatz, 1994), but we resort to a much simpler approach.

Polar coordinates are useful for designing oscillators: the radius stands for amplitude, and the phase is the argument of a sinus function (or used to index a wavetable). We propose the following scheme for pitch correction: Given a system

$$\begin{aligned} \dot{r} &= f_\mu(r,\theta) \\ \dot{\theta} &= g_\mu(r,\theta) \end{aligned}$$

we generate the audio signal $x(t) = r\sin\theta$. The fundamental frequency of this signal is assumed to depend on the parameter μ. (There will also be harmonic overtones, unless the system is linear.) In this model, there will be only one pitch present, and it will be harmonic. Thus, we can apply a pitch follower to the output signal to obtain an estimated fundamental frequency $\hat{f}$. Assuming the system is reasonably well-behaved under changes of the control parameter μ, we could either increase or decrease its value until $\hat{f}$ matches some specified frequency. On the other hand, it may be desired to be able to regulate timbral nuances with the control parameter. In that case, it is no longer available for pitch adjustment, but we may resort to changing the time step of integration instead. Too long step sizes have to be avoided for the sake of keeping the system stable. This pitch correction scheme does not require polar coordinates—fortunately, since several potentially useful oscillators are better expressed in Cartesian coordinates.

The control scheme is outlined in Figure 4.9, where the user has a pitch control but no direct access to the bifurcation parameter μ. It should be noted that this is our first example of a feature-feedback system. The mapping M takes the estimated fundamental frequency $\hat{f}_o$ and the user specified frequency F_o and regulates the bifurcation parameter μ and the step size h until $\hat{f}_o \approx F_o$ within some error tolerance. The exact nature of the mapping function will depend on the particular nonlinear oscillator used. A more detailed description of a pitch correction scheme along similar principles will be deferred to Section 6.4, where it will be used in the context of noise-driven oscillators.

Now, let us consider a few examples. The Van der Pol oscillator

$$\ddot{x} + x + \mu(x^2 - 1)\dot{x} = 0 \tag{4.21}$$

generates a pure sinusoid for $\mu = 0$. As the parameter μ increases, the waveshape approaches a square wave, while simultaneously the pitch decreases. For a fixed step size h, there is a limit as to how high values the parameter can take before the system becomes unstable, which is about $\mu = 8$ for $h = 0.1$ and using the fourth order Runge-Kutta method. Conversely, with μ fixed, there are limits on the step size before instability sets in. Now, both μ and h can control the pitch, with increases in h making it higher and increases in μ lowering the pitch. The waveshape and hence the timbre depends only on μ. Thus, there is a certain room for independent control of pitch and timbre.

When solving the system (4.21), it has to be recast into two first-order equations, one representing the position x, and the other its velocity $\dot{x}$. There is nothing in the way for using the velocity instead of the position as the audio signal. Since the first derivative $\dot{x}$ of a signal can be obtained by highpass filtering $x(t)$ (see Section 2.3.4), it will have a brighter timbre.

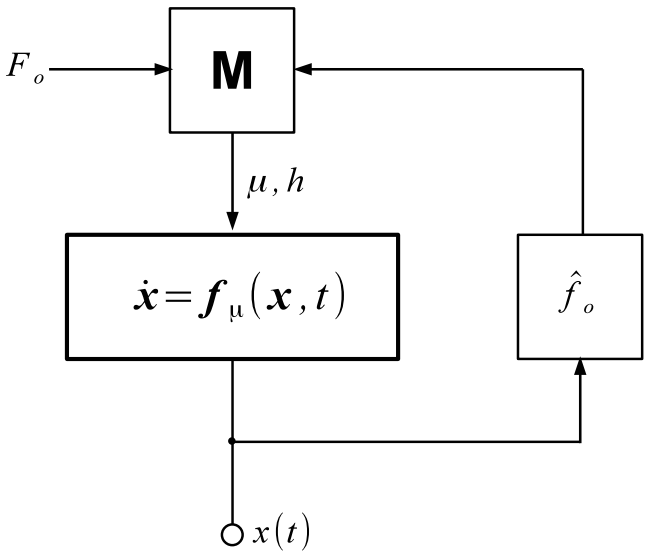

Figure 4.9: The pitch correction scheme. The pitch estimate $\hat{f}_o$ is used to obtain control parameters μ and a step size h that will yield the desired pitch F_o.

As a second example, consider the Toda oscillator

$$\ddot{x} + d\dot{x} + e^x - 1 = F\cos(\omega t). \tag{4.22}$$

This system was studied by Kurz and Lauterborn (1988) with fixed damping $d = 0.05$ and variable forcing F and driving frequency ω. Period doubling sequences can be found, as well as chaos. At some parameter values, the oscillator's amplitude is markedly greater; in other words, it has resonances at those locations of the parameter space. Such a sudden amplitude increase is clearly visible in Figure 4.10. This large dynamic range may be problematic in sound synthesis, but is easily handled with automatic gain control (a compressor) or even a limiter. For high frequency driving, the system oscillates at the driving frequency. Chaos appears only for low frequencies. Pitch control in the Toda oscillator is almost as simple as setting the driving frequency to the desired pitch. The problem, however, is that the efficacy of pitch control depends on the forcing as well as on the damping, so the interplay between all three parameters needs to be taken into account. If the frequency is swept linearly across some range while the damping and forcing are constant, there may be period doublings causing octave jumps in pitch or chaotic oscillations at some driving frequencies.

Like the Toda oscillator, the Van der Pol oscillator may also have a forcing term. For sound synthesis, this is not restricted to sinusoids, but could be any signal. Indeed, the most varied dynamics of all of these three oscillators is to be found when they are driven by an external signal. According to Sprott, a forced relaxation oscillator studied by Van der Pol and Van der Mark in 1927 was one of the first examples of a chaotic circuit; subharmonic frequencies were observed as well as an "irregular noise", which prompts Sprott (2010, p. 235) to remark: "Had they pursued that innocent observation, they would have discovered chaos half a century before it was otherwise widely known, and

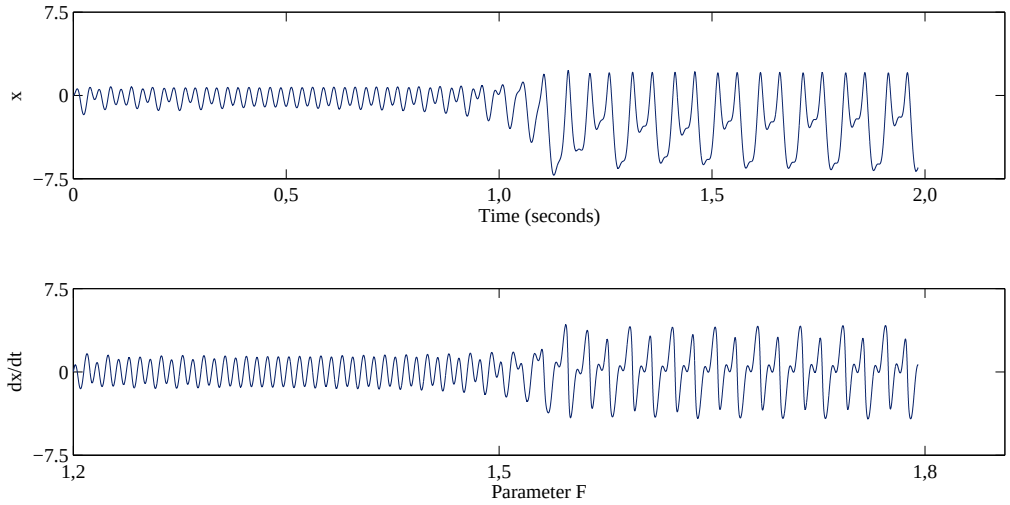

Figure 4.10: The Toda oscillator with $d = 0.05, \omega = 1.57$. The forcing increases linearly from 1.2 to 1.8 over two seconds.

history might have been different."

The Rössler system is also interesting to use as a nonlinear oscillator. With its single nonlinear term, it is one of the simplest chaotic flows.

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + xz - cz \end{aligned} \tag{4.23}$$

Typical parameter values known to produce chaos are $a = b = 0.2,\ c = 5.7$. The Rössler system produces pitched sounds for lower values of c. This range can be interesting to explore for sound synthesis, again with a pitch-correcting scheme. The Rössler attractor is almost shaped like a disk in the xy-plane, and these coordinates are most useful for audio output, whereas the z coordinate is usually close to zero, apart from irregular peaks.

Example 4.4. The subharmonic cascade leading to chaos can clearly be heard in the Rössler system as c in (4.23) is swept from 2 to 6, in this case using $a = b = 0.3$. The amplitude also increases with c. Even when the system is chaotic, there is a predominant frequency in the spectrum.

4.5 Synchronisation and chaos control

The idea of chaos control is to apply some form of adjustments to parameter values, to system variables, and perhaps to insert new components in the system so that its trajectories can be made to reach a fixed point or some desired periodicity. Synchronisation

of chaotic systems is also possible. If two identical systems (say, two Lorenz systems, with three variables each) are coupled with just one variable from one of them driving the other system, perfect synchronisation of these two systems has been demonstrated. As the number of coupled systems rises, it appears to be harder to achieve full synchronisation of all systems, although perhaps some subset may synchronise (Pecora et al., 1997). The topics of chaos control and synchronisation are interesting in the context of sound synthesis because similar coupled systems may be built as synthesis models. In particular, cross-coupled synthesis models should be understood against the backdrop of synchronisation.

Large ensembles of coupled oscillators have one thing in common with filtered maps. A group of coupled oscillators may be seen as a spatially extended system, whereas filtered maps have a memory of their past. However, if the filtered map is written in state space form, it too can be regarded as a spatially extended system. The similarity, then, can be seen in long onset transients both in coupled oscillators and in filtered maps.

4.5.1 Sync in music and acoustics

Synchronisation is a widespread phenomenon. In musical settings it is of fundamental importance, as witnessed when several musicians are engaged in spontaneous improvisation; they easily end up on a common pulse unless their strategy is specifically to avoid it. Applauses begin disordered, until at some point a synchronised clapping occurs. The clapping may speed up, and there may even be a period doubling when it reaches a certain tempo. Néda et al. (2000) show that these phenomena can be simulated by the Kuramoto model, which we return to in Section 4.5.4. In music, this kind of synchronisation in listeners is often referred to as entrainment. Large and Kolen (1994) introduced a dynamic system model for this phenomenon, loosely based on the circle map. Their system adjusts both its internal phase and its frequency to an incoming sequence of spikes that represent onsets of notes.

Synchronisation is so common (especially in music), that one may ask whether it is not the cases of persistent desynchronisation that most need to be explained. In Ligeti's piece *Poème Mechanique*, 100 mechanical metronomes are started simultaneously, each set to its own tempo (actually there are less than 100 tempo markings on standard metronomes, so a few metronomes may share the same tempo). Here, any synchronisation is out of the question, but instead there are moments when all metronomes or large groups of them happen to beat simultaneously, and sweeping gestures as ensembles of metronomes phase out. As a matter of fact, two metronomes put side by side on a freely moving board over two aluminium cans synchronise. The synchronisation is in-phase (the pendulums beat in the same direction), and occurs for small frequency differences after a few tens of seconds (Pantaleone, 2002).

If slightly detuned and positioned close together, organ pipes may synchronise and oscillate at the same frequency. The synchronisation at several harmonics of two organ pipes was studied by Abel and Bergweiler (2007) as a function of the distance and detuning of the pipes. In one experiment, frequency locking occurred for detuning by as much as ± 2 Hz; for greater detunings the pipes suddenly produced beats. As the distance of the pipes was varied, but keeping their detuning fixed at 0.7 Hz, the effect was to vary the

strength of the coupling. For a distance of 10 mm, the pipes were synchronised, but at a 25 mm separation they already became uncoupled. Evidently, Abel's and Bergweiler's findings have practical value for organ builders who would like to avoid registers that produce beats, although the same principles could be explored in novel acoustic instruments that would allow various amounts of detuning and coupling between oscillators. In digital synthesis, such effects are straightforward.

4.5.2 Chaos control

Chaos control is important to know about regardless of whether chaos is the desired behaviour or not. It may be important to know which actions to avoid, lest wanted chaos settle into regular dynamics. Feature-feedback systems are precisely such a case, where complicated dynamics get filtered out or absorbed, probably mostly due to the stabilising effects of feature extractors which involve much smoothing.

Since a ground-breaking paper by Ott, Grebogi and Yorke (1990), the control of chaos has been an important research topic. The Ott-Grebogi-Yorke (OGY) approach takes advantage of the fact that chaotic orbits coexist with unstable periodic orbits. If a system parameter can be perturbed by a small amount, then it may be possible to restrict the motion to any desired periodicity. The OGY approach works even on systems which are not explicitly known, as long as there is an experimental time series available. It should be noted that it is exactly the presence of chaos that makes the system amenable to control; if the motion is already periodic, then the orbit can only be slightly changed by small parameter perturbations.

Promising as the method sounds, its implementation does not seem to be immediately obvious, and has yet to be applied for sound synthesis. Other methods exist, however. According to Ogorzałek (1993), the simplest way to suppress chaos is to change the system parameters. Bifurcation diagrams are handy for exploitations of periodicities as functions of a parameter. By finding out in advance which parameter values correspond to which periodicities, the system can be controlled. For use in sound synthesis, this technique might be worth trying. Periodic windows of arbitrary length may exist as they do in unimodal 1-D maps, scattered across the parameter range. Producing a continuous glissando is not exactly straightforward in this scenario. Further, the waveshape is not directly controllable. Chaos may also be quenched by "shock absorbers" as Ogorzałek explains. This mechanism is similar to the shock absorbers that smooth out unpredictable vibrations in a car driving on an uneven surface. In this case, suppression of chaos is achieved by adding a subsystem, so there is no need for control signals to introduce perturbations as in the OGY scheme.

Chaos control by delayed feedback was already mentioned above in the discussion of delayed feedback FM (Section 4.3.5). This may take the form

$$x_{n+1} = (1 - K)f(x_n) - Kx_{n-D},$$

for some delay length D and coupling strength K (Buchner and Żebrowski, 2000). It is clear that the system's dimension increases by introducing this delay, but in contrast to the example of delayed feedback FM, this does not necessarily imply greater probability of obtaining chaos.

A host of other control methods have been proposed, see Chen and Dong (1993) for an early comprehensive review. The synchronisation of chaotic systems is a closely related topic that has also been much studied, see Pecora et al. (1997) for a review. The importance of this for feature-feedback systems is perhaps mostly the knowledge that coupled chaotic systems in fact may synchronise, even in the strong sense that the difference between two corresponding variables belonging to different subsystems will eventually tend to zero.

4.5.3 Coupled oscillators

In a software synthesiser, or in a digital instrument built in a sound synthesis language, there may be any number of interconnected oscillators. Dahlstedt et al. (2004) studied a special case involving a large number of oscillators (from 10 to 200) with cross-coupled FM, but avoiding self-modulation. The frequencies f_i of the carriers were randomly and uniformly distributed on a logarithmic frequency axis spanning over four octaves. Their synthesiser model was

$$y_i[n] = \cos\left(\theta_i + \sum_{j=1}^{N} m_{ij} y_j[n-1]\right) \tag{4.24}$$

with phases

$$\theta_i[n] = \theta_i[n-1] + \frac{2\pi}{f_s} f_i, \quad i = 1, \ldots, N$$

and a connexion matrix m with all diagonal ($i = j$) elements set to zero. Connexions were made randomly between oscillators with probability p, and the coupling strength was randomly drawn from a Gaussian distribution with zero mean and standard deviation s. Then the maximum Lyapunov exponent was calculated as a function of the number N of oscillators, the connexion probability p and strength s. In short, the greatest Lyapunov exponent was found to increase as a function of both p and s, and this increase took place more rapidly (for lower values of p and s) as the number of oscillators grew. The parameter space is broadly divided into a regular and a chaotic regime, the latter occurring whenever the combined values of the three parameters are sufficiently large.

No detailed descriptions of the resulting sounds were given in this study, but its purpose should be seen in the context of mapping out non-chaotic regions of a parameter space, which would be suitable to explore further with evolutionary computation techniques. To the contrary, sounds that happen to be chaotic may well deserve being considered for sound synthesis; whether they are musically useful or not depends both on the actual system and the musical needs. However, as Dahlstedt and his colleagues observe, this model tends to produce very harsh sounds when it reaches the chaotic region. In effect, it may not be the most versatile system for chaotic sound synthesis.

By introducing a time-variable global modulation index in (4.24), it becomes easy to explore the effects of varying amounts of coupling, so we replace the coupling matrix m with a scalar β. For $\beta \approx 0$ the sound is smooth, but at some point when the index is increased, there is a rhythmic phenomenon where the sound intermittently becomes very

noisy and rich in high frequency content. This rhythmic modulation is more prominent if self-modulation is allowed. At an even higher modulation index, the noisy character pervades the sound. As with more familiar types of FM, the harmonic ratios between the carrier frequencies are also very important for the resulting timbre. Setting all carriers to the same frequency or to simple harmonic ratios will tend to produce simpler sounds, but already slight deviations from simple ratios will greatly increase the noisiness.

There are many high-dimensional systems of circularly coupled ODEs that display spatiotemporal chaos (Sprott, 2010). Such systems may however also exhibit chaos that is synchronised between the nodes, that is, purely temporal chaos, although some of them need to be of a rather high order before admitting chaotic solutions. In contrast to the above FM model where couplings may occur between all oscillators with some probability, other coupling topologies are global coupling where all oscillators are connected, and local couplings to the two nearest neighbours on a ring. Many of these so called *circulant* systems could be used for sound synthesis; in particular, if multi-channel playback is available, it could be interesting to spatialise the oscillators each to its own output channel. Some experiments indicate that it may be worthwhile to use certain circulant systems for the generation of control signals, each of which may then control the frequency and perhaps amplitude of its respective oscillator. The independent pitch contours of each oscillator are easier to follow than a mix of each of the coupled nodes if used directly to generate the output signal.

So-called labyrinth chaos is a remarkable example of a conservative chaotic system that results in a random walk (Sprott, 2010). The circulant system is described by the equation $\dot{x}_i = \sin x_{i+1}$, but we add an extra damping term to the right hand side, so the system becomes $\dot{x}_i = \sin(x_{i+1}) - px_i$. There are N such circularly coupled equations, so that $x_{N+1} = x_1$.

Example 4.5. Labyrinth chaos does not result in bounded orbits, hence the variables $y_i = \sin x_i$ are used for control signals that map to the amplitude and frequency of $N = 71$ oscillators tuned to the harmonic overtones of a fundamental at 55 Hz. Here $p = 0.32$ is used, but for lower values, the patterns become more irregular. It is not known whether this example is actually chaotic or not.

4.5.4 The Kuramoto model

Synchronisation is a very important topic in the study of networks. Often the nodes of the network are modelled as oscillators, and the degree to which they synchronise is modelled as a function of the coupling strength. Kuramoto's model is perhaps the most investigated one, where there are N oscillators that are globally coupled with coupling strength K:

$$\dot{\theta}_i = \omega_i - \frac{K}{N}\sum_{k=1}^{N}\sin(\theta_i - \theta_k), \quad i = 1, \ldots, N \tag{4.25}$$

The Kuramoto model shows up in a wide range of contexts that deal with synchronisation. For instance, several metronomes standing on a common, freely moving base can be described by the Kuramoto model (Pantaleone, 2002). The dynamics of the Kuramoto

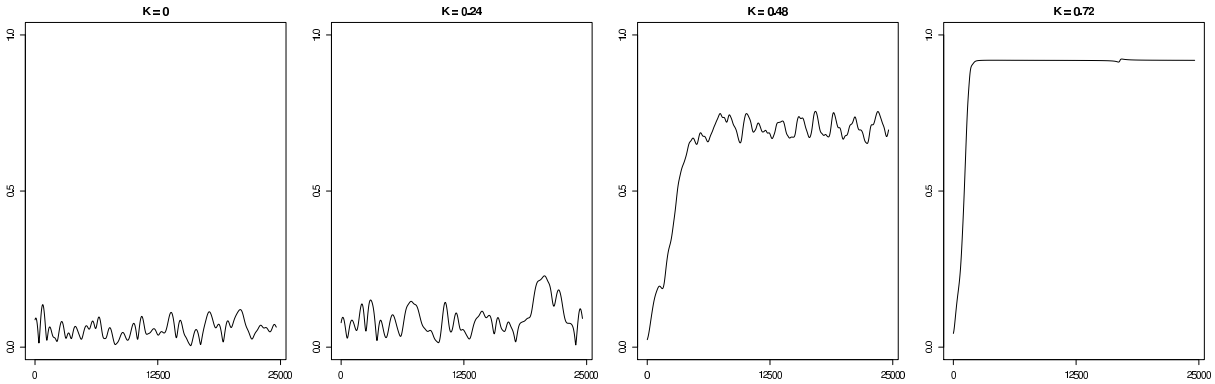

Figure 4.11: Time evolution of the order parameter in a Kuramoto model with 225 oscillators over 25000 time steps with $h = 0.01$. The onset of synchronisation happens rapidly if the coupling K is sufficiently strong. From left to right, the coupling increases from $K = 0$ to $K = 0.72$.

model is quite complicated, but for a brilliant account of some of the most important contributions to understanding it, see Strogatz (2000). There are several generalisations of the basic Kuramoto model, but similar synchronisation phenomena can be observed in most of them. If noise is added to (4.25), then noise induced synchronisation can take place in a manner that is similar to stochastic resonance (Sakaguchi, 2008). The dynamics of the Kuramoto model depends on the coupling strength, but it also depends on the probability distribution of the frequencies ω_i, which are usually taken to have a unimodal distribution. Greater spread in the distribution of frequencies will counteract the system's tendency to synchronise. For the state of the art in understanding the Kuramoto model and many related globally coupled systems of oscillators, see Ott and Antonsen (2008). Essentially, they have proposed that the global dynamics, in the limit of an infinite number of oscillators, can be described by some low-dimensional ODE.

For a quantification of the overall degree of synchronisation in a set of coupled oscillators with phases θ_i, the order parameter

$$re^{i\psi} = \frac{1}{N}\sum_{k=1}^{N} e^{i\theta_k} \tag{4.26}$$

is introduced, characterising the mean direction ψ of the phases and their coherence r. The order parameter depends on the coupling strength. Below a certain critical coupling strength $K < K_c$, r tends to zero. When the coupling is stronger than K_c, the order parameter increases as a function of K. The time evolution of the order parameter is shown in Figure 4.11, where it is seen that, for $K > K_c$, there is a rapid rise in the order parameter to its final level. The transient duration, or the time it takes before the order parameter stabilises on its final value, is a function of the size of the system. The more oscillators, the longer the initial transient becomes, as is shown in Figure 4.12. This can be intuitively understood by considering the time it takes to travel across all the nodes in the network; synchronisation somehow depends on all nodes being responsive to all other nodes. Although the Kuramoto model is distinctly different from filtered maps,

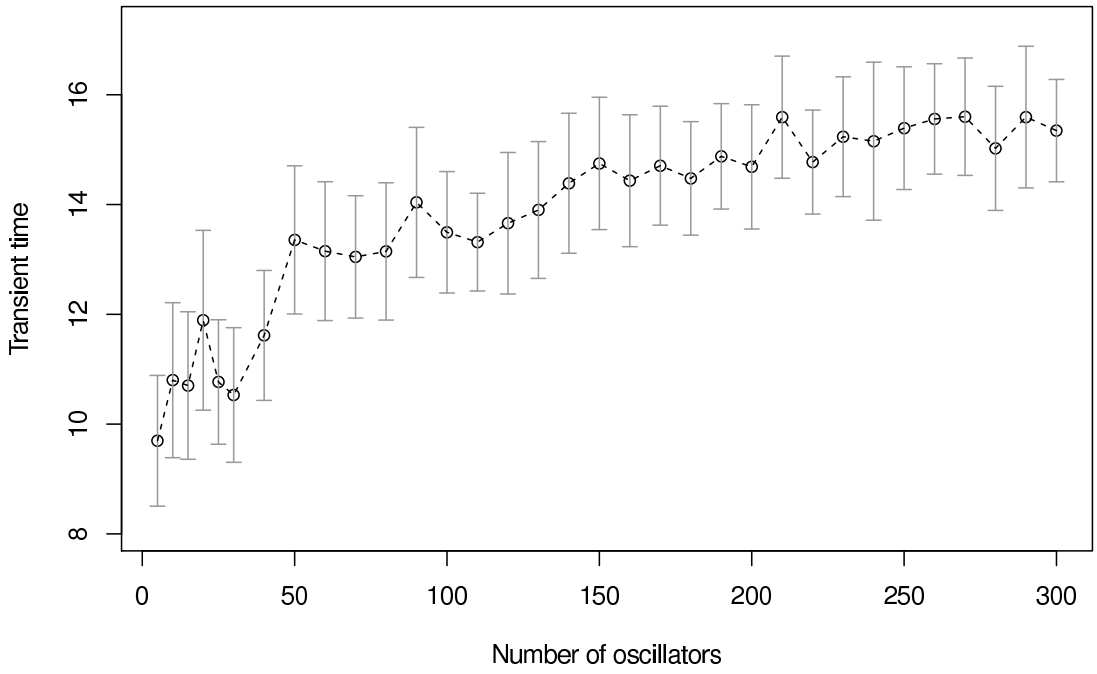

Figure 4.12: Transient times until synchronisation is reached increase as a function of the number of oscillators in the Kuramoto model. Here, $K = 1$, and for each number of oscillators, the circles show average durations taken over 100 runs, with standard deviations indicated by the error bars.

they share the same size-dependent initial transients.

Taking θ_i as the phase in a sinusoidal oscillator, a mix of the N oscillators may be used for sound synthesis. It should be noted, however, that this model is $\mathcal{O}(N^2)$ in complexity, so it is preferable to keep the number of oscillators small. Fortunately though, the most sonically promising results appear to occur precisely when there are a moderate number of oscillators (perhaps no more than five). Additionally, the Kuramoto model is not prone to become unstable as some other differential equations are, so it is sufficient to use the Euler method for integration.

It may be revealing to compare this model to coupled FM (4.24). If written in continuous time, the FM model becomes

$$\begin{aligned} \dot{\theta}_i &= \omega_i + \sum_{j=1}^{N} m_{ij}\dot{y}_j \\ y_i &= \cos\theta_i, \quad i = 1, \ldots, N \end{aligned}$$

Writing out the derivative of y explicitly, we have

$$\dot{\theta}_i = \omega_i - \sum_{j=1}^{N} m_{ij} \sin\theta_j$$

which, in contrast to the Kuramoto model does not tend to reduce phase differences, and hence induce synchronisation between the phases. Although in the Kuramoto model all oscillators are coupled, self-modulation does not occur since the phase cancels, similarly to the restriction that self-modulation is avoided in the coupled FM model.

Example 4.6. An interesting application of the Kuramoto model in sound synthesis is to use it for additive synthesis when the coupling is weak. Then, by gradually increasing the coupling from zero, the partials will begin to interact more and more. Eventually, all partials may synchonise if they are not too distant to begin with. Here, five oscillators are used, and the coupling increases from none to $K = 10$. At the end a low pitch is heard, which means that the driving frequencies have found compromise frequencies at harmonics of that low fundamental.

With a wider spread of frequencies, other effects will be heard such as a gritty sound when the entire system is unable to synchronise despite very strong coupling. There are many possible variations of the basic structure of the Kuramoto model. Instead of its full connectivity, a ring topology can be used where the oscillators are connected only to two neighbours each, thereby forming a circle. Hybrids with different couplings, some using FM and some using the phase difference, are promising for further study.

Some variants of the Kuramoto model exhibit a wide range of phenomena, such as *chimera states* (Kuramoto and Battogtokh, 2002; Abrams and Strogatz, 2004). In this state, the oscillators split into two populations, one synchronised and another desynchronised. Chimera states arise in systems of identical oscillators with the same internal frequency, where the coupling is neither global (all-to-all) nor circular, but such that there are some non-local couplings. On the other hand, in the Kuramoto model (4.25), it is commonly assumed that there is a stochastic distribution of oscillator frequencies, which makes the partial synchronisation of oscillators with close frequencies less of a mystery.

It should be added that the model giving rise to chimera states is derived from a partial differential equation of oscillators on a ring, where coupling is some decreasing function of the distance between two points on the ring. This model is then spatially discretised by solving a large number of coupled systems, typically several hundreds of oscillators. But as previously noted, for sound synthesis the low dimensional cases are the most interesting to investigate further.

4.6 Conclusion

The state space approach of dynamic systems provides solid ground for further investigations of deterministic synthesis models. Whereas the focus of the previous chapter was more on the relation between perceived sound and synthesis parameters, here we have discussed the Lyapunov exponents, the order parameter and bifurcation plots as tools for describing the dynamics of maps and flows. The idea is to attain a global description of the possible dynamics of a system over a broad range of parameter values. This is very different from the more direct approach of setting the parameters, generating a signal and listening to the resulting sound. Such close studies of synthesis models are of course necessary when they are to be used in musical contexts, but it can be very useful to

search through broader parameter ranges for promising as well as useless regions in the parameter space, particularly when the synthesis model is not already well known. The application of these ideas to feature-feedback systems will be further elaborated upon in Chapter 6.

The filtered maps were introduced partly as a simplified model of a feature-feedback system, in which the feature extractor plays the role of a smoothing filter. As we saw in the smoothed map (4.8), the parameter r of the logistic map could be taken to values that would normally be unstable. There appears to be a rule that can be formulated in general and slightly vague terms as follows: *If a lowpass filter is inserted in the feedback path of a chaotic map, the likelihood of obtaining chaos will be reduced, unless its nonlinearity is simultaneously increased.* This trade-off is deliberately used in some strategies of chaos control. The usefulness of filtered maps for sound synthesis in general, and physical modelling in particular, only adds to the fascination of such systems.

Large systems comprised of simple building blocks such as the coupled ODEs are the favoured objects for study in complexity science, although cellular automata have also been very popular. The use of large ensembles of coupled oscillators in sound synthesis has scarcely been explored, but seems promising. We used the Kuramoto model to show how the length of an initial transient can be dependent upon system size. Similar phenomena will be encountered again when feature extractors of variable analysis window length are used inside feature-feedback systems.

The nonlinear oscillator with pitch control that was sketched in this chapter was our first example of a feature-feedback system. This seems to be a very useful application of feedback control, but the mapping needs to be tailored to the specific nonlinear oscillator involved, which we have not yet attempted to do. However, our goal is quite different when developing autonomous instruments; in particular, they should provide more original behaviour, in some sense, than just tuning an oscillator to a specified pitch. This is where self-organisation becomes a key issue.

The occurrence of chaos in acoustic instruments as well as its deliberate use in musical composition and sound synthesis was discussed in Section 4.2. As long as a deterministic system is used for sound synthesis, the best guarantee for some temporal variety in its output is that the system be chaotic. Quasi-periodicity may also to a certain extent be helpful for the same purpose. Randomness is an easy solution, albeit one that will only be briefly considered as we return to the problem of pitch control in nonlinear oscillators (see Section 6.4). Therefore, in the deterministic feature-feedback systems that later will be introduced, chaotic dynamics is the only hope for achieving any interesting behaviour. However, that is not to say that chaos is enough. Next, among other things, we will consider complexity as an aesthetic criterion for music in general and its suitability for qualifying feature-feedback systems in particular.

Chapter 5

Cybernetic Topics and Complexity

If an autonomous instrument generates sounds that the composer neither specified nor expected, can it then be said to generate self-organised sound? In the fields of experimental music, algorithmic composition, generative music, nonstandard synthesis, and in music-making with semi-autonomous instruments (which were discussed in Chapter 1), one often meets a rhetoric of self-organisation and emergence as well as frequent references to complex systems and cybernetics. The composer or performer plays the role of an engineer who assembles the circuitry or programs the algorithm, just to let it run on its own. If the composer did not explicitly specify the output of the programme, then, one may argue, it must have organised itself.

Concepts such as emergence and self-organisation are vague and may not be possible to define in an uncontroversial way. In particular, there is scope for interpretation regarding the notion of organisation. One possibility is to define organisation in terms of complexity, for instance as related to the number of states that a system may be in. Unfortunately, complexity is another concept with many competing definitions, as we shall see in this chapter.

Norbert Wiener initiated the interdisciplinary cybernetics movement in the 1940s, taking its name from the Greek word for steermanship (Wiener, 1961). Cybernetics brought intriguing problems to the attention of the scientific community, including questions about self-regulating systems and feedback. As it happens, a variety of feedback systems are found in most musical examples where the topics of emergence or self-organisation are mentioned. By definition, feature-feedback systems of course include feedback. After a broad classification of feedback systems in Section 5.1, we give a few examples of compositions where the use of acoustic or other kinds of feedback is an important generative element.

What criteria should one use to evaluate the output of an autonomous instrument? In the end, there are no objective criteria since the choice of criteria is itself informed by subjective opinions and differing aesthetics. Nevertheless, as we shall argue, there are good reasons for using some form of complexity criterion in such evaluations. Various definitions of complexity have been proposed, some of which will be reviewed in Section 5.2.3. It is no exaggeration to say that the field of complex systems studies is itself a very complex subject. Hence, the discussion of complexity is more of an inventory of ideas that may be useful in the study of feature-feedback systems, than a definite proposal of

criteria for evaluation.

Complexity is closely related to emergence and self-organisation. Since the 1980's, there have been some attempts to scientifically study these topics. Nevertheless, hand-waving references to self-organisation and emergence are still common, not only in the music literature, but also in many other fields. Given some recent attempts at building a systematic theory of self-organisation, we may speculate that a more rigorous theory of self-organised sound or music may someday become an actuality. However, there is much to do before such a theory is in place; meanwhile, we will have to be content with a more intuitive understanding of self-organisation.

With the typical complex systems researcher having a background in the natural sciences, the examples of complex or self-organising systems often come from disciplines such as statistical physics or biology. Nevertheless, many of these ideas should be applicable to music in general, and to autonomous instruments in particular. In fact, some researchers have already studied music and other arts with the methods of dynamic systems and complexity theory. To a musician, some of those studies may appear to be ill-informed or slightly naïve because of their simplifying assumptions. In order to counterbalance that neglect of aesthetics and general informedness about current musical practice, we will also discuss the way complexity has been treated in music, particularly in the *new complexity* movement. The tools are at hand for the automated evaluation of certain quantifiable aspects of art. There are some situations where the automated evaluation of autonomous instruments would be helpful, but in this chapter we will also expose some of the pitfalls of such automated evaluations using various complexity measures.

This chapter is divided into three parts, dealing with feedback systems (Section 5.1), complexity (Section 5.2), self-organisation and emergence (Section 5.3) both from an aesthetic and a scientific perspective. Thus we take a closer look on some aesthetic issues related to autonomous and semi-autonomous instruments and related practices that were discussed in Chapter 1, in particular including the classical question of beauty as it relates (or not) to complexity and simplicity. The aesthetic thread will then continue in Chapter 8.

5.1 Feedback systems

Feature-feedback systems involve a particular form of feedback, but there are many other uses of feedback that have been seen mostly in experimental musical settings. In this section, we consider some examples of feedback in an electroacoustic setting, as well as in digital sound synthesis.

Acoustic feedback, also known as the Larsen effect, named after Absalon Larsen (Augoyard and Torgue, 1995), occurs when an acoustic amplified signal is reinjected into the amplifier, so that the amplitude grows until it reaches a point of saturation. Loudspeakers and amplifiers have nonlinear transfer functions at high gain settings, for instance in the shape of a sigmoid function, which limits the growth of amplitude. If a sinusoid was the original waveshape, it may come out as something more like a square wave if the system is driven into its nonlinear regime.

Feedback is a much broader concept than the Larsen effect; indeed it is such a ubiq-

uitous phenomenon in musical performance that it often passes unnoticed unless it is absent. Without audible or haptic feedback from the instrument, musical performance is virtually inconceivable. The Theremin is controlled by moving one's hands closer or farther away from its two antennas, controlling amplitude and frequency, respectively. Thus, the musician is deprived of a haptic response, so that all adjustments of the hand's position have to be based exclusively on what is heard. In this sense, feedback is intimately connected with control. Likewise, the absence of immediate acoustic response from offline sound processing and offline modes of programming in synthesis languages may partly explain why many users prefer realtime interactive software.

Acoustic feedback systems have had their use in music. Computer models of some such systems have been implemented (see Section 5.1.5). Although computer models may capture some essential traits of the dynamics of the original acoustic system, there are inevitable fundamental differences. Acoustic systems are open, in the sense that the presence of an audience or subtle ambient noise may have a huge impact on the dynamics.

5.1.1 Examples of feedback systems

Just to give an impression of how wide-spread feedback systems are, here are a few examples of applications, drawn from music mostly:

- Electroacoustic feedback: The Larsen effect; prepared loudspeakers
- Analogue: No-input mixers; cross-coupled oscillators in modular synthesisers
- Acoustic: Musician with haptic and sonic feedback from instrument
- Optical: Video feedback
- Digital systems: Recursive filters (including delay lines), feature-feedback systems; iterated maps
- Note-level symbolic composition: Process-oriented music

Prepared loudspeakers are speakers that have been modified by inserting objects into their membranes, or fastening the speaker element to various other objects as in David Tudor's *Rainforest*. If a contact microphone is placed on the speaker membrane that it is connected to, it will rattle by the sounds the speaker makes, thus producing an oscillator. An example of this approach is an electroacoustic feedback network by Jon Pigott (2011), including a prepared speaker that induces vibrations in a spring, which are amplified and sent back to the speaker.

As mentioned in Chapter 1, no-input mixers use feedback by routing the output of the mixer back to its input, often employing some effects processing in the signal chain. Analogue modular synthesisers can be set up in feedback configurations, e.g. by routing the output of one oscillator to a control input of another, and the output of the second oscillator back to the first.

The need for both haptic and sonic feedback when playing a musical instrument has already been mentioned. Haptic feedback in new digital instruments is obviously useful, and is an active field of research.

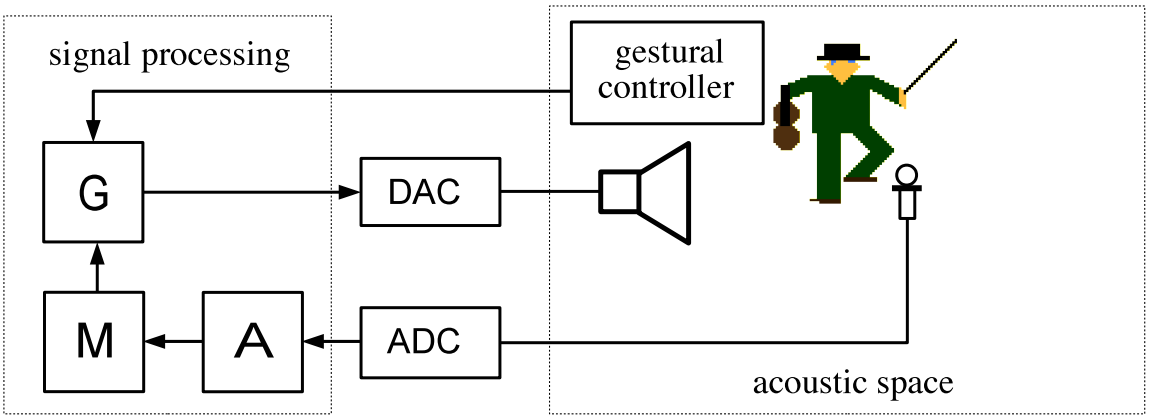

Figure 5.1: Electroacoustic feedback system with both audio and gestural input from a musician. G stands for signal generator, A for analysis or feature extractor, and M is the mapping. The output from the signal generator is sent via a digital to analogue converter (DAC) to a loudspeaker in an acoustic space. A microphone picks up sound from the loudspeaker and from the musician, which is sent back to the computer via an analogue to digital converter (ADC). The gestural controller may be used for a more immediate influence on the synthesis algorithm.

Video feedback occurs when a video camera is pointed at the monitor it is coupled to. Symmetric patterns, spiraling waves, and other complex shapes result depending on a number of control parameters such as the relative angle between the camera and the monitor, their distance and zoom factor, the focus, the brightness of the monitor and the ambient light level (Crutchfield, 1984). Video artists have been known to exploit such effects, perhaps not as the sole source for an entire work, but at least as special effects. Crutchfield (1984) proposed to use video feedback as an analogue computer for the efficient simulation of spatiotemporal systems similar to two-dimensional cellular automata. Apparently, video feedback never caught on as a simulation tool, perhaps because digital computers were meanwhile becoming more powerful.

Simple feedback systems are in common use in computer music. Recursive filters are indispensable components in synthesis languages where biquad filters are often available as unit generators; they are needed in numerous digital audio effects, as well as in several synthesis models. However, these recursive filters are in a sense atomic units from which more complicated instruments can be built, hence we shall not make too much of the fact that they are feedback systems. A good programming interface to a filter hides its internals from the user, so there is no way to insert other things in the feedback path. Such systems only appear as a black box with an input and an output.

Note-level composition of process-oriented music is also listed as an instance of feedback systems. Here, what we have in mind is a compositional technique where one takes a short musical object (phrase, or motif) and transforms it into a new musical object which is placed after the first one in the score; then this object is likewise transformed, and so on. We will outline this idea in Chapter 7, and compare it to signal level feature-feedback systems.

For a musician engaging with a semi-autonomous instrument, one of the most important aspects may be whether the feedback takes an acoustic path or not. A generic semi-autonomous instrument with acoustic feedback is shown in Figure 5.1. The musician in this case has two different means of influencing the system; either by making sounds that are captured by the microphone, or by using a gestural controller mapped to the synthesis parameters of the signal generator. There are two distinct ways that the feedback loop may be closed in this system. The most common is that the musician receives a response from the system by the sounds it makes, and reacts to these sounds by playing something that is captured by sensors or by the microphone. However, a stronger (or more immediate) kind of feedback results if the acoustic output of the loudspeaker in the room is also fed directly back into the microphone. The latter scenario forms the backdrop for Di Scipio's *Audible Ecosystems* in all its versions. Acoustic and haptic feedback from an acoustic instrument to a performer is such a commonplace situation that we will henceforth simply take it for granted and not discuss it further.

Acoustic feedback systems may be called *open systems*, since they are indeed open to any accidental input from their sonic environment. Semi-autonomous instruments are necessarily open systems. Correspondingly, feedback systems that exist only in the computer will be referred to as closed systems if they take no input. The next two chapters will deal exclusively with closed systems.

5.1.2 Feedback topologies

Let us consider a few examples of feedback topologies that may be found in analogue or digital synthesis, which are particularly relevant to feature-feedback systems. Suppose there is a signal generator and a mapping from its output. The mapping may or may not be preceded by a feature extractor. Three cases are shown in Figure 5.2. A basic feedback system takes the output signal and maps it to its synthesis parameters. Depending on what is meant by mapping, one may include examples such as Xenakis' GENDYN programme and the Karplus-Strong algorithm (Karplus and Strong, 1983). The Karplus-Strong algorithm fills a delay line with noise, and then takes the two-point average of the samples and writes them into the delay line, leading to decay in high spectral energy similar to a plucked string. The parameters are stochastically perturbed in the case of GENDYN, leading to a random walk, whereas for Karplus-Strong, the sample buffer corresponds to the generator unit, and the averaging process to the mapping. In these two cases the mapping does not really influence the synthesis parameters; nevertheless, there is a feedback from the output to the internal state of the signal generator.

Cross-coupled systems have two or more signal generators that influence each other. As was mentioned in Chapter 4, large ensembles of coupled oscillators can exhibit various different phenomena depending on their coupling topology. Three types of network topologies are commonly studied, namely the circular topology, all-to-all or global coupling, and the so-called *small-world* networks which are used to model social networks, for instance (Mitchell, 2009). Small-world networks hold an intermediate position between global and circular coupling with respect to the number of edges that connect the nodes.

Any effects processing or analysis and resynthesis scheme may be used iteratively in

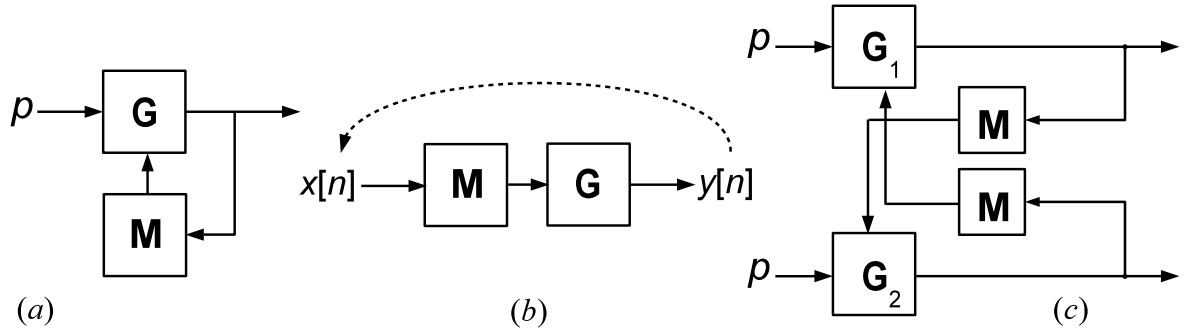

Figure 5.2: Some variants of feedback: (a) basic feedback system; (b) a transformation of an input may be used in offline feedback loops if the output y is used as input x in the next stage of processing; (c) cross-coupled feedback. G is a signal generator and M is a mapping. User defined parameters p are also shown.

an implicit feedback loop. A sound file is processed and the output is again processed with the same algorithm, and so on. This process can be repeated indefinitely and the successive stages of the process can be concatenated into a single sound file. As an example, Morris (2007) mentions the process of starting with white noise and iteratively applying noise reduction. Other examples of such a process, including Alvin Lucier's *I am sitting in a room*, will be discussed below.

A striking example of the importance of choosing an appropriate network topology in feedback-based art is provided by the electronic sculpture *crash and bloom* by Douglas Irving Repetto (2004). This biologically inspired installation consists of a number of custom-made electronic boxes which are able to flash a light and play a short tone. There are different types of boxes, some of which have an input and an output, and others that can split a signal into two outputs, or join two inputs into a single output. Messages called *pings* are sent between the boxes. Three simple rules apply to each box:

1. When the box receives a ping, it flashes its light and plays its tone during its *active* phase. When it has finished playing, it passes on the ping to its output.

2. Each time a box has received a ping, its active duration is reduced until it reaches a minimum duration. If it is at the minimum duration, next time the box receives a ping, it is reset to full duration.

3. If a box receives two pings while active, it stops without passing the ping along.

A further type of box is the ping injector, which is used to get the system going. A circular topology results in the ping traveling around the loop, gaining speed until it resets to the slowest pace according to the second rule. More complicated behaviour becomes possible with a figure-eight topology. Then, there is a splitter that doubles the number of pings traveling around the network, which leads to an exponential increase in pings (the "bloom" of the title) until a point of saturation (the "crash") where the boxes are reset. After a crash, there may not be any surviving pings left to set the system going

again. Quite understandably, Repetto used a more complicated network structure with several nested loops for exhibitions of *crash and bloom*. This allowed for more varied, and not least, persistent behaviour.

5.1.3 Electroacoustic feedback

Before describing some deliberate uses of acoustic feedback, let us note that it is a serious problem in amplified live music; however, there are some strategies to avoid it. Most obvious are the positioning of microphones in relation to loudspeakers and the adjustment of the sound level from the PA system, but signal processing with equalisers and modulators may also be used.

If the signal is modulated periodically by delay modulation, phase modulation, FM, or frequency shift, this has the effect of increasing the amount of gain that can be applied before instability. Gain increases of up to 12 dB have been obtained this way. Some of the theory behind this is exposed in Nielsen and Svensson (1999), along with experimental comparisons of various feedback suppression schemes. The modulators are linear time-varying systems, whose effect in the frequency domain is to add shifted copies of the input spectrum X, weighted by a set of modulator transfer functions G,

$$Y(f) = \sum_{k=-\infty}^{\infty} G_k(f - kf_m)X(f - kf_m) \tag{5.1}$$

assuming periodic modulation at some frequency f_m. Acoustic feedback in reverberant rooms is complicated to model because of resonant frequencies. Intuitively, the feedback instability would be expected to happen at one of the room's resonances. Hence, the goal is to smooth out irregularities of the room's transfer function. Modulators do this by shifting frequency components. Frequency shifting can easily be seen to work, since for each round trip of the feedback loop, the frequencies are shifted in the same direction. It may be less obvious that the other modulation schemes should also work since they have both positive and negative sidebands; a frequency component that gets shifted up will be shifted back down in the next round. What matters, however, is the carrier suppression and the spread of spectral components. Feedback control can be modeled in the computer too, and is a useful technique in sound synthesis using feedback networks where unlimited gain increase may cause problems.

While being a nuisance in most concert settings, feedback has also been exploited as a sound source in its own right in some musical compositions and improvisations. In particular, feedback is at the core of some types of live-electronic music such as the *Audible Ecosystems* of Agostino Di Scipio. In the previous chapter, it was mentioned that acoustic chaos can be produced by an acoustic feedback system with a full-wave rectifier in the loop, as demonstrated in a study by Kitano et al. (1983). Loosely similar systems have been used in musical settings, where the instabilities of acoustic feedback are exploited, either alone, or more commonly with additional signal processing in the loop. It is not surprising then, that acoustic feedback gets mentioned several times in Nicolas Collins overview chapter on live electronic music:

> In the tightly proscribed world of pop it became the bad-boy way to insert the irresponsible and unpredictable. Cheap, loud and only somewhat controllable, it possessed a seemingly willful independence that, in the 1960's, echoed the spirit of the times. But it wasn't just noise, it had content — feedback traced in sound the movement of a microphone or speaker, and it revealed the resonant frequencies of rooms, musical instruments, mouths, culverts and barrels (Collins, 2007a, p. 41).

Robert Ashley's *The Wolfman* (1964) is a veritable sonic assault, notorious for its use of extreme amplification and feedback shaped by the mouth of the performer. David Behrman's *Wave Train* (1966) uses pickups placed directly on piano strings, connected to guitar amps underneath the piano. When the feedback sets in, the pickups vibrate and rattle on the strings, producing complex changing timbres (Collins, 2007a). David Tudor's compositions are among the most musically successful uses of feedback systems. The compositions are documented as circuit diagrams, but apparently details were adjusted for each performance. According to Ron Kuivila, the piece *Untitled* (*Homage to Toshi Ichiyanagi*) from 1972 builds on another piece called *Pepscillator*, which was "an autonomous electronic system with 'no input"' (Kuivila, 2004, p. 21). *Untitled*, however, uses input from three tape recorders, although an intricate feedback system is maybe the most important part of it. Tudor's compositions have not been widely studied from a technical point of view, but digital simulations should be possible from existing circuit diagrams.

The frequency shifting and pitch shifting techniques as described above have been used in a musical instrument by Morris (2007), where the purpose is not to avoid audible feedback but to impose pitch contours on the sound. Morris also experimented with many other techniques such as interrupting the feedback path by manually damping the loudspeaker, or by using a *shutter* that gates the signal temporarily. Further examples of musical works that use acoustic or other feedback paths are discussed at greater length below.

5.1.4 The tape-loop is related to Karplus-Strong

Numerous composers in electroacoustic music have employed the tape loop to create anything from short echos to repetitions after a delay of several seconds. Examples are to be found in Pauline Oliveros' early studio works such as *I of IV* (1966), *Bye Bye Butterfly* (1965) and several others, in Steve Reich's *It's gonna rain* (1965), *Come Out* (1966) and much of Brian Eno's ambient music, e.g. *Discreet music* (Holmes, 2008). Tape loops were even used live as a delay system. Reich's use of loops involved cutting and splicing together tape segments with the same recorded fragment on each and letting two or more such loops gradually phase out against each other (Reich, 2002).

Delay lines simply repeat whatever has been introduced into them. As a primary compositional technique, this might easily result in laid-back compositions running on autopilot. Pauline Oliveros somehow manages to eschew this trap, as her works with tape loops appear to be the product of very attentive listening. If this is correct, the explanation may perhaps be found in the rigidity of the process: "The delay feedback retained and extended every adjustment made to the oscillators, and having to put up

with the consequences of her actions for several minutes put Oliveros in a contemplative state — her tape delay pieces are characterised by small changes over long periods of time" (Collins, 2007a, p. 44).

Alvin Lucier's *I am sitting in a room* derives from a similar process. A text is read and recorded onto tape. The tape is played back in a room, and recorded again. This recording is played back and recorded, and so on. The entire process is repeated until the resonances of the room have completely overtaken the original recorded signal. For the piece, a montage of each stage put in succession is used.

I am sitting in a room has become an influential work, and there are probably several reasons for this. As it is realised in its most known form, it consists of Lucier himself reading a text describing the process of making the piece. This is already an elegant concept—the work contains its own construction manual. It also carries an expressive quality due to the slow-paced, well-balanced rhythm of Lucier's reading, including some stuttering; the reason for submitting the recording to the process being to "smooth out any irregularities my speech might have", as the text goes.

Apart from the conceptual and poetic interest this work has, it provides a good example of a feedback process. In technical, but not in perceptual terms, this process is identical whether applied to tape stretches of several minutes or to the briefest possible snippets of tape.

The microphone, the tape recorder, the amplifier and the loudspeaker each act as filters, and also add some distortion and noise to the signal. The intention of this work (as in many other of Lucier's works) is to bring forth the acoustics of the room. In the first few iterations, its resonant frequencies are at first not too apparent, but gradually emerge as these frequencies become reinforced. Several realisations have been made, and the score actually admits some freedom as to what text is read and who reads it (Broening, 2006). Depending on the acoustics of the room, the process of blurring the speech takes a larger number of iterations if done in a space with dry acoustics than it does in a room with a long reverberation time. If Lucier's tape loop process is transferred to digital processing, some surprising connexions are revealed.

If we make the optimistic assumption that all electronic components in the setup yield perfect reproduction, all we need to model is the room impulse response and a delay length corresponding to the duration of Lucier's speech. Then, if the original message of length T samples is x_n, $n = 0, \ldots, T-1$, the subsequent copies are

$$x_n = h_n * x_{n-T}, \quad n \geq T \tag{5.2}$$

using the room impulse response h for convolution. Note what happens if, first, the input is white noise, second, the delay length is rather short—say, less than 50 ms, and third, the impulse response is taken to be a two point average: This is the formulation of the Karplus-Strong algorithm! Again it must be emphasised that any resemblance is purely technical, not perceptual. Nevertheless, it is striking that two so different sounding processes as the simulation of a plucked string and Lucier's *I am sitting in a room* can be modeled as two extreme possibilities of the same system.

More realistic modelling should take into account the nonlinear distortion, filtering, and noise introduced by the equipment. If all filtering can be lumped together with the room's impulse response, and ignoring noise, the model becomes $x_n = h_n * f(x_{n-T})$ for

some nonlinear function f. Here, if the delay time corresponds to a pitch period instead of several seconds, and the filter h and the function f are suitably chosen, this might as well be a physical model for woodwind instruments, as we saw in the previous chapter (Section 4.3.6).

We have committed another important simplification that is easy to overlook. In passing, Broening (2006, p. 94) notes: "Other than adjusting the volume from iteration to iteration to avoid tape saturation, Lucier made no changes to the resulting sound." One may speculate what the work would have sounded like without those adjustments. Either the gain would have been too low, and the signal would gradually decay while tape hiss would take over, or if the gain were too high, distortion from tape saturation would accumulate. What Lucier did manually is an example of adaptive gain control. The concept is readily extensible to the automatic control of any variant of iterated sample buffer synthesis.

5.1.5 Pendulum Music

In his influential collection of aphorisms *Music as a gradual process*, Steve Reich proposed slowly unfolding musical processes where one could follow the changes as they happened. This remarkable trust in the sufficiency of the process and whatever it leads to is characteristic of algorithmic composition. "Although I may have the pleasure of discovering musical processes and composing the musical material to run through them, once the process is set up and loaded it runs by itself" (Reich, 2002, p. 34). As Broening (2006) remarks, Lucier and Reich had some points in common regarding the conceptual basis for their music.

Pendulum Music (1968) is probably the most radical process piece by Reich. It has the appearance of a slightly bizarre physics classroom demonstration. Four amplifiers are used, each with a microphone plugged in and suspended above it. The gain is adjusted to produce feedback when the microphone is in its resting position above the loudspeaker. The piece begins by releasing the microphones so that they swing like pendulums, and as they cross the feedback zone a tweeting sound is produced. As the microphones lose momentum the orbits narrow down, and the proportional duration of feedback increases until it becomes continuous. When all motion has ceased, the piece ends by unplugging the microphones.

A computer simulation of *Pendulum Music* has been attempted by Coco (2006). A variable delay line modelled the distance between the microphone and amplifier, a variable gain synchronised to the distance was used, a resonant filter modelled the room's impulse response, and a signal limiter simulated the amplifier's clipping of the waveform. Considered as a system, *Pendulum Music* has some interesting properties. The motion of the microphone should introduce a doppler shift, shifting the frequency upwards as it approaches the speaker, although the feedback frequency also depends on distance.

Pendulum Music is a good example of a feedback system in music, not least for its conceptual clarity. It is immediately evident what is going on, in contrast to some more recent examples of feedback systems that run the signal through layers of signal processing. In those cases, one cannot know for sure what strange quirks the music will end up taking. Reich's piece, on the other hand, has a predetermined development, and

there can be no surprise about how it will end. There are several recordings of *Pendulum Music* that display its potential. Ensemble Avantgarde included four different takes of it on the same CD, all of the same ilk though durations vary—their sound is always smooth and polished. Sonic Youth's rendering of it is of a decidedly different character; it sounds more like a rock band, as befits them.

5.1.6 Burns' nonstandard waveguide feedback network

Waveguides used in physical modeling is the point of departure for the feedback system of Christopher Burns (2003), although there are some notable differences. Whereas waveguides are built from linear filters in recursive structures with carefully controlled gain, the feedback networks of Burns use limiting functions that guarantee that the system's amplitude cannot grow without bounds. However, the signal may become saturated and end up containing just a DC component. The entire network consists of several similar sections, each taking an input which is amplified, submitted to limiting (waveshaping) and delayed with a continuously variable delay time (see Figure 5.3). The output is connected bidirectionally to other such sections in a circle, and each section is also connected to a loudspeaker. Two microphones are used for input to the delay lines, but a sound file may also be used to excite the system. The network reportedly offered quite some surprise when used in a realisation of Cage's piece *Electronic Music for Piano*, which includes the words "feedback" and "for David Tudor", but otherwise leaves many choices to the interpreter:

> The network performs in unpredictable ways, sometimes imitating onsets and pitches played at the piano very precisely, sometimes remaining quiet during busy passages, sometimes bursting into noise in the middle of a long silence. [...] Complex, swooping pitch contours with continuous micro-alterations of timbre are typical, while the continuously varying delay lengths produce shifting, inharmonic pitch relations. Depending on the gain settings, punctuating noisy explosions may also be frequent (Burns, 2003, p. 269).

If anything, this would be a good example of a system with emergent sonic behaviour, as Burns himself suggests.

The overall architecture, or topology, of this system is quite interesting as it combines two different kinds of feedback: one that runs directly in the digital delay lines, and one that uses the electroacoustic system of microphones and loudspeakers. Even without the acoustic feedback path, this would be a rather complex system. Its similarity with coupled systems of oscillators such as the Kuramoto and FM models that were discussed in Chapter 4 (Section 4.5.4) should be noted. Here, however, the components are not oscillators, and there is actually nothing internal to the system driving it. In the absence of any input it would remain silent, unless the thermal noise in the system would manage to get sufficiently amplified.

It is also interesting to note the use of modulated delay lengths. As discussed in Section 5.1.3 above, delay modulation is one of the strategies for feedback suppression, so it probably also serves the function of increasing the gain level that can be applied before feedback occurs in Burns' network.

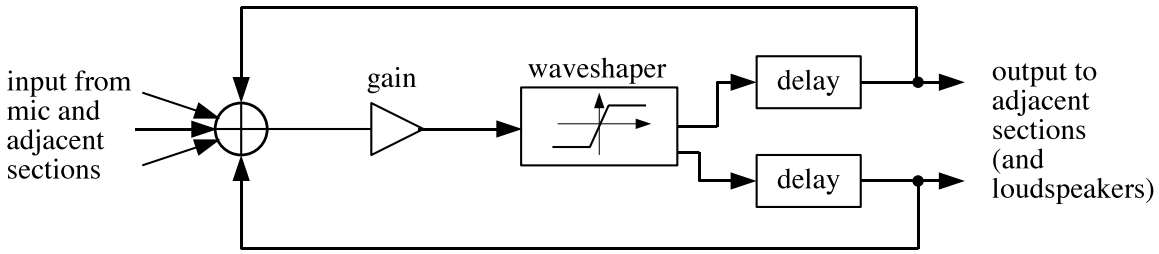

Figure 5.3: One section of Burns' feedback network. Eight such sections are connected in a bidirectional delay line. The delay times are modulated. Adapted from Burns (2003).

5.1.7 Xenakis' GENDYN algorithm

The GENDYN programme (also often called GENDY) by Xenakis has reached fame, with several re-implementations and extensions. The synthesis technique had already been used as one of the sonic elements in *La Légende d'Eer*, but it was the sole sound source for *S.709* and the lesser known piece *Gendy3* (Serra, 1993). The original GENDYN programme, written in Basic (Xenakis, 1992, ch. XIV), is full of idiosyncrasies. In terms of software engineering, it is anything but safe (Hoffmann, 2000). Of course, there is no point in painstaking software engineering of a programme that is intended only for the generation of a single composition. In fact, GENDYN is a good example of a composed instrument, and its unstructured code is not untypical.

Basically the idea of GENDYN is to generate a waveform by choosing points in the time-amplitude plane, which are linearly interpolated. Before the waveform is iterated, it is modified. Each point undergoes a random perturbation in both time and amplitude, given some probability distribution. A concept of elastic barriers ensures that points that would reach outside the boundaries for amplitude are reflected inside. Likewise, time-points are perturbed and kept within bounds by elastic barriers. Hence a kind of Brownian motion results. If the perturbations are strong, the sound will be noisy, whereas for mild perturbations it will be quasi-periodic.

Although the output of the GENDYN algorithm as such is dynamic with meandering streams of more or less noisy pitches, there is some timescale where one begins to hear the constancy within all variation. Thus the next step is to introduce a higher control level. A separate programme was used in the composition of *Gendy3* to generate up to sixteen simultaneous tracks, with durations of sound and silence and all other parameters drawn from various probability distributions (Serra, 1993).

Extensions and variations of the original GENDYN programme include realtime control, other forms of interpolation of the waveform than linear, and the generation of percussive sounds (Brown, 2005). A frequency domain version treats successive spectral envelopes the same way as the waveform in the original programme (Döbereiner, 2009). This is a bit ironic, because it was precisely the Fourier decomposition of sound that Xenakis criticised as an impasse (Xenakis, 1992, ch. IX). However, this criticism was partly based on the results from the early era of additive synthesis, and much of what

then appeared impractical is now easily achieved, for instance with spectral modeling synthesis including stochastic variations and transients, as discussed in Chapter 3.

GENDYN is a striking example of the elegance of generating an entire piece of music by the same algorithm, letting its global form emerge from the processual modification of its waveforms. It shares this principle with autonomous instruments—indeed it *is* an autonomous instrument. The algorithm is clearly a feedback system where a waveform segment is the unit that is processed in a feedback loop. An obvious extension is to insert a feature extractor in this loop, or alternatively apply a feature extractor on an external signal that controls the synthesis parameters.

The GENDYN programme shares some features with other nonstandard synthesis algorithms based on the waveform, such as the one elaborated by Herbert Brün and used in his SAWDUST pieces (Brün, 2004; Roads, 1985). But in Brün's case, the waveforms are deterministically produced, in a way that ultimately puts the composer in a more direct control of the end result. One-dimensional cellular automata have also been applied to sound synthesis in a way that is slightly reminiscent of the GENDYN algorithm (see below), with the main difference being that the cellular automata have a fixed length. It is precisely this flexibility of the waveform's length that gives the particular gestural flavour to Xenakis' algorithm.

5.1.8 Sound synthesis by cellular automata

Cellular automata are often taken as examples of complex systems. Both their rules and their geometry are extremely simple, consisting only of a discrete lattice of sites that may be in one of a small number of states at any moment, and the next state of each cell is determined by a function of its immediate neighbourhood. Given these simple conditions, it is all the more remarkable that they are capable of forming complex patterns. Cellular automata as well as iterated maps can also be regarded as feedback systems, something that is rarely mentioned, perhaps because it is all too obvious. Nevertheless, the current state of all the cells is the information that gets fed back by applying the chosen updating rule, and this uniquely determines the next state of each cell.

Since the patterns that form in cellular automata are not imposed from above, but arise from the iterated application of simple rules, cellular automata are often said to display emergent behaviour. Physical modelling of partial differential equations can be carried out in the framework of cellular automata (Toffoli, 1994). Since partial differential equations are defined on continuous space, in continuous variables and in continuous time, all these domains will have to be discretised before they can be computed. It may appear counter-intuitive that cellular automata are capable of simulating any physical phenomena that are usually described in partial differential equations. In the case of acoustic physical modelling, Toffoli demonstrates an example of sound wave propagation, where the wave expands as a growing circle despite the fact that the cells in the model only interact in vertical and horizontal directions.

Cellular automata have been used in various ways both for sound synthesis, and for note-level composition (Miranda, 2007). Chaosynth, Miranda's programme for granular synthesis, uses cellular automata where oscillators are associated with submatrices of the total lattice of sites. A note-level application is the CAMUS programme, which is also by

Miranda. Both these instances of musical applications of cellular automata add further structure to the automata in the form of non-trivial mappings to the musical domain.

An interesting case of sound synthesis by cellular automata was introduced by Jacques Chareyron (1990). His application, entitled LASy (for Linear Automata Synthesis), uses one-dimensional automata where the cell's states correspond to integer sample values. Instead of the customary two-state automata, Chareyron's have 2^b states, where b is the number of bits. This leads to an enormous number of possible rules, even when considering only rules that depend on the sum of three neighbours of each site.

Let p be the length of the array that stores the samples. Then LASy uses an update rule

$$y[n] = f(y[n-p-1], y[n-p], y[n-p+1]) \tag{5.3}$$

for $n > p$, whereas the first $p+1$ values of the array are initialised with any suitable waveform. A rule that yields plucked string sounds is to sum the arguments to the function f, and scaling them by a constant. This is similar to the Karplus-Strong algorithm, which instead uses the sum of two samples (Karplus and Strong, 1983). General updating rules, however, may include nonlinear combinations of the arguments, leading to nonlinear filters similar to those that were mentioned in the previous chapter (Section 4.3.1).

The feedback structure of LASy is about as simple as can be, but with the addition of a feature extractor that determines the functional form of the updating rule, this model would be a feature-feedback system. Cellular automata use integers for their states, and usually only binary numbers. Sound synthesis, however, is more conveniently handled using real numbers in floating point representation. That is one reason why we do not use cellular automata for feature-feedback systems.

5.2 Complexity

The composer or musician who makes music with an autonomous instrument will usually be interested in evaluating the resulting output according to its appropriateness. Aesthetical criteria will be important, and these are very much up to each practitioner to decide upon. Nonetheless, there is a sound argument in favour of using criteria related to some kind of complexity for the evaluation of autonomous instruments. The reason why complex results should be sought is that trivial behaviour is so easily achieved. This is the case in cellular automata (Chareyron, 1990), where most rules cause simple behaviour, whereas finding rules that produce complex behaviour is not that easy. Feature-feedback systems may be quite complicated with programmes consisting of thousands of lines of code; then, if they rapidly settle on a sinusoid that goes on indefinitely without variation, this would be a very uneconomical way of producing that result.

Concepts such as complexity and simplicity are notoriously hard to define, regardless of whether the context is restricted to musical complexity or complex systems in general. In music, some composers have engaged in fierce aesthetical debates of complexity versus simplicity as related to style, while some music psychologists have studied the perceptual complexity of various (often relatively simple) stimuli. *Complex adaptive systems* is a

heterogenous field of research that studies anything from ant hives to the internet, and which has spawned several notions of complexity. In the words of Melanie Mitchell (2009, p. 95, italics in original),

> [...] there is not yet a single science of complexity but rather several different sciences of complexity with different notions of what complexity means. Some of these notions are quite formal, and some are still very informal. If the *sciences* of complexity are to become a unified *science* of complexity, then people are going to have to figure out how these diverse notions—formal and informal—are related to one another, and how to most usefully refine the overly complex notion of *complexity*.

Both formal and informal notions of complexity will be of interest when applied to the evaluation of autonomous instruments. The formal notions are often clearly defined and, at least in principle, possible to measure, whereas the informal notions may be more intuitive and subjective. Objective measures of complexity may be useful for the automated evaluation of autonomous instruments. Therefore several candidates for such measures will be reviewed. Meanwhile, it is important not to lose contact with the musical realities and the many ways in which a piece of music may be experienced as being complex. In order not to fall into that trap, we discuss some examples of how music and other arts have been analysed from a complex systems perspective as well as the aesthetic debates around musical complexity and simplicity.

In Chapter 7, the experienced complexity of certain sound examples made by feature-feedback systems will be evaluated in a listening test. This is an example of informal and subjective complexity assessment. Among the various objective complexity measures that will be discussed below, some may be more useful than others in assisting with the evaluation of autonomous instruments. However, the appropriateness of these complexity measures remains to be evaluated. Nevertheless, some of the formal notions of complexity will be needed in the ensuing discussion of emergence and self-organisation in Section 5.3.

5.2.1 The New Complexity movement

At times, there has been much talk about musical complexity, at least in contemporary music circles. Clear definitions of musical complexity are often absent from these discussions, but still a sample of the various points of view will give an impression of what some people mean by musical complexity.

There is probably no uncontroversial definition of musical complexity. If the written score is the arbiter, then we shall agree that the music of Ferneyhough and a few other representatives of the *New Complexity* movement reach almost insurmountable levels of complexity—not least for the interpreter who has to make sense of the notation. For the listener, however, what looks complicated in the score may sound simple; and conversely, what looks simple may sound complex. Accordingly, Richard Toop suggests that "any meaningful theory of musical complexity would have an aesthetic or perceptual orientation, rather than a technical one" (Toop, 2010, p. 90).

There was a lively discussion about complex music in *Perspectives of New Music* in 1993-94, and specifically about the new complexity. James Boros (1994) defended

complex music as vehemently as he showed his disdain for the new simplicity and new romanticism. Philip Glass and John Adams were accused of having "acquiesced to the culture industry's demand for consumable objects" (Boros, 1994, p. 97). Furthermore, Boros (1994, p. 98) confessed that his "sense of horror in the face of certain minimalist music may also be attributable to its 'didactic' nature, to its habit of 'talking down' to the audience". It is as though this "simpler" music was perceived as a direct threat to more complex musical expressions—perhaps rightly so, if one thinks of market shares and media coverage. Although not providing a definition, Boros tries to narrow down the concept of complexity. First, he observes that it is not the amount of information presented that matters as much as the contextual relationships. A rapid flurry of notes is likely to fuse into a perceptually simple, homogenous object. Complex music also encourages the coexistence of multiple viewpoints, implying an ambiguity of hierarchies such as foreground and background layers. For Boros, complexity is a question of aesthetics, as formulated in the following credo:

> The aspect of "complex" music that I find most appealing is the one which others seem to find troubling, namely that much of it has ragged, tattered edges, foregoing the "hot licks" and glossy, synthetic sheens characteristic of the typical mass-produced regurgitation in favor of laying bare its imperfections, its flaws, its intrinsic awkwardness (Boros, 1994, p. 93).

Richard Toop (1993) discussed several aspects of musical complexity; its historic lineage as well as its relationship with the complicated and with difficulty, all with much of the same aversion towards the neo-simplicity, albeit stated in a somewhat milder tone than that of Boros. Difficulties can arise in the domains of performance, in composition and in listening, as noted by Toop. To some extent, this difficulty can be attributed to unfamiliarity. Indeed, it is a common experience that some music needs repeated exposure before it begins to make sense. This, then, is the rationale for making complex music, and as Toop illustrates with a few historical examples, it is not a new fashion.

> In effect, this is the standard composers' plea in favor of complexity: it insists on the right of, even the need for the composer to "over-compose"—to create something that resists immediate decoding, while seeking to offer some kind of incremental revelation through repeated listening (Toop, 1993, p. 48).

Barry Truax (1994) argued in favour of a different conception of complexity. Truax criticised the new complexity movement for its single-minded focus on musical organisation in terms of inner relationships of the musical elements (which he called *inner* complexity), rather than involving external contexts in various ways (*outer* complexity). According to Truax, the solution would be to make use of both these inner and outer complexities. Among the outer relationships that would bring some true complexity to music, Truax suggests the use of specifically chosen performance spaces, social situations or circumstances (which we nowadays would call site-specific composition), the incorporation of recognisable environmental sounds as in soundscape composition, and writing music for specific individuals rather than just any performer of the instrument in question. Clearly, the conception of outer complexity is one that suits most electroacoustic music better than that of inner complexity. So-called "real-world sounds", be it field recordings

or studio recordings of musical instruments, abound in most dialects of electroacoustic music.

The use of references, quotations, or recorded fragments taken from an external audible reality is perhaps less idiomatic in music realised with autonomous instruments. This is not to say that sampled sounds cannot be introduced in feature-feedback systems; in fact, we will briefly discuss how to do so in Chapter 7 (see Section 7.4).

According to Toop (2010), the representatives of New Complexity were reluctant to work with purely electroacoustic media, although they sometimes use live-electronic processing. He also remarks that although these composers have a wide intellectual horizon and surely know about scientific advances such as complex adaptive systems, they rarely make much out of that parallel, but rather see their artistic practice in relation to other music, visual art and literature. The situation is different with autonomous instruments, where the link to complex systems is much closer.

5.2.2 Music analysis

Let us consider some of the factors that cause problems for objective measures of musical complexity. Objective measures are here taken to be measures that have been formalised so that two pieces of music may be unambiguously ordered with respect to their complexity level, regardless of the measure's appropriateness to a listener's subjective judgments. Listeners with different background will come to the same piece of music with differing expectations and listening skills. There is also a role for learning in music, be it of one particular work or a style. Repeated exposure to the same piece will reveal more of it, provided it is rich enough. For these reasons, the perceived complexity of a piece of music cannot only be a function of the music, but depends also on the listener. Objective complexity measures do not face that problem, although their adequacy for classifying music depends on how closely they are related to the perceptual complexity. What is meant by perceptual complexity also needs to be clarified. For now, we assume it to be related to the difficulty a subject has with recalling or reproducing a musical segment.

The entropy of musical sequences is one example of objective complexity measures. Boon and Decroly (1995) measured first- and second-order entropies of note sequences taken from several compositions ranging from J. S. Bach to Elliott Carter. The second-order entropy, which measures note transition probabilities, is most relevant. Boon and Decroly found that the entropy, interpreted as the degree of complexity, did not increase over the course of music history, but varied from one piece to another. It should be noted that the underlying complexity criterion here does not rely on judgements made by listeners; rather, it is conceived on the basis of a speculation on what might constitute complexity in music, and then applied to music that fits the molds of the analysis method in question. Thus, it is an objective complexity measure whose perceptual validity remains to assess.

Apart from entropy, several other statistical measures are discussed by Beran (2004), though not specifically in relation to complexity. The common trait of these methods is that they are applied to whole corpora of compositions or recordings. After all, it is not very illuminating to quote the entropy or fractal dimension of a single composition. The utility of such methods lies in trying to make predictions about questions such as which

period, composer, or performer the analysed music comes from. Indeed, the complexity-related method proposed by Ribeiro et al. (2012) which is based on permutation entropy (see Section 4.1.4) has its intended use in musical style classification. Their use of the permutation entropy is however restricted to very small windows of the signal—they used five samples as the embedding dimension, or about 10^{-4} seconds—which should be expected to reveal differences related to the upper frequency range of the average short time spectrum rather than processes over longer time spans.

The entropy of tone sequences (Boon and Decroly, 1995) deals with one melodic line at a time. One will find that repeated notes and periodically repeating patterns, such as an ascending and descending scale has the least entropy, while randomly chosen notes have the highest entropy, with most music falling somewhere in-between. However, this complexity measure misses the effects of harmony, so one might try to incorporate a measure of unpredictability of the chord changes. This might work fine for homophonic settings, but what are we to make of "diagonal" writing, where notes almost never enter simultaneously, resulting in simultaneous interwoven melodies and ever-changing harmonies as is typical of Elliott Carter's string quartet writing, particularly in the slow movements? And what about Xenakis' string quartets, in which glissandi are about as common as stable pitches?

In John Zorn's and Naked City's cut-up pieces and John Oswald's Plunderphonics, the stylistic references undoubtedly add layers of complexity, this time of an extrinsic nature. The study of Boon and Decroly goes as far as to Carter, most likely because his music still fits the analysis paradigm. Their finding that complexity has not increased through Western music history—correct or not—should be taken with a grain of salt, since their analysis method reduces all music to individual voices represented as time series of pitches, and clearly that is not always a relevant way to regard music. So, if an objective complexity measure is to be a relevant criterion for comparison between two pieces of music, then both pieces have to fit into the same conceptual analysis scheme.

Until recent times, most music analysis has focused on note-based music in contrast to oral traditions and improvisation; it has favoured the study of western music rather than ethnic musics; and it has studied pitch (as melody and harmony) and temporal relations (rhythm), while largely ignoring timbre, texture and morphology (Godøy, 1997). Aspects that are particularly highlighted in electroacoustic music, such as the concrete referentiality of sound, its spatialisation, timbre and morphology, have passed uncommented in most music analysis devoted to the classical Western repertoire. Schaeffer and his followers naturally form an exception. In fact, even spectral music which tries to blur the boundaries between harmony and timbre (Murail, 2005) is problematic when studied with analysis techniques based on single notes, as opposed to some more global approach.

In Schaeffer's sound classification (see Section 2.1.3), there are the three overarching categories *Objets trop élémentaires, Objets équilibrés, Objets trop originaux*—too elementary objects, balanced objects, and too original objects (Schaeffer, 1966; Chion, 1983). Schaeffer had his clear preferences for objects that were neither too simple and predictable, nor too complex and hard to memorise. Sound synthesis in the early days of analogue electronic and computer music had a certain reputation for producing sterile timbres with a lack of nuances. This is less of a problem today, but when the sound production is entirely controlled by an algorithm as it is in autonomous instruments, too

simple or redundant output is not an unlikely outcome. Too complex or original sounds might also be a problem, depending on one's expectations. In Schaeffer's classification, a sound that is originally balanced may become either redundant or excentric, too unpredictable, if prolonged. Therefore, it seems motivated to take the duration into account if a perceptually grounded formal complexity measure should be developed. To date, there are not many attempts to develop quantitative measures related to perceived musical complexity that work directly on the audio signal. An exception is the work of Streich (2006), which we will return to later.

5.2.3 Notions of complexity

The sciences of complex adaptive systems have been applied to many different fields including the immune system, ant hives, the nervous system, the internet and much else (Mitchell, 2009). Even the arts, and perhaps primarily the visual arts, have received some attention from complex systems researchers. Given this diversity, it is no wonder that many different complexity measures have been proposed. Underlying questions often deal with how to describe a system, an object or a signal, but also how hard an object is to construct, and how organised it is. Next follows a resumé partly based on Mitchell (2009).

Complexity as size. We would like to think that humans are more complex than a single-celled amoeba, but its DNA "has about 225 times as many base pairs as humans do" (Mitchell, 2009, p. 96). Obviously we first have to agree on what units to count, and why. The number of different parts could also indicate the complexity level, but again one has to decide what level to look at, since a part may contain smaller parts. Moreover, this says nothing about how the parts are interconnected (Sommerer and Mignonneau, 2003).

Entropy. Also known from information theory as Shannon entropy, or information,

$$H = -\sum_{x \in X} p(x) \log p(x) \tag{5.4}$$

where X is the set of all events (or messages) x under consideration (Shannon, 1948). If there is only one possible message the entropy is zero, whereas if each one of N different messages are equally likely, the entropy is maximised and becomes $\log N$. This is one of the problems with entropy as a complexity measure; a uniformly distributed stochastic variable, while maximising entropy, is very simple to describe in statistical terms. The intermediate cases, where some messages are highly improbable while others occur more often, are generally more interesting. Furthermore, Shannon entropy is only defined for discrete variables x, although continuous variables could be partitioned somehow into discrete bins. Several measures that extend the basic entropy has been proposed, including conditional entropy (Prokopenko et al., 2008; Boon and Decroly, 1995) where the probability of an event given a previously occurring event is calculated, or the permutation entropy (Bandt and Pompe, 2002).

Algorithmic information content. At least for computer programmers, there might be some appeal in the notion that complexity could be measured as the length of the shortest computer programme that could generate a complete description of an object. This

theoretically important measure is also known as Kolmogorov complexity (Solomonoff and Chaitin are also independently credited with its discovery). Some difficulties are immediately apparent. How do we know for sure that there does not exist an even shorter algorithm for solving the problem? A consequence of the Kolmogorov notion of complexity is that a random string cannot be compressed and thus coded in shorter space than the string itself; thus random strings hold the highest complexity level. This is often seen as an unwanted effect.

Logical depth. This concept goes back to the mathematician Charles Bennett, who proposed it as measure of how difficult an object is to construct. To make this idea tractable, the object in question is a binary string, and the logical depth is the number of steps it would take for a Turing machine to yield the specified sequence as its output. Several different Turing machines might solve this problem; if so, the shortest or simplest of them should be chosen. Mitchell (2009, p. 101) states that logical depth matches our intuitions about complexity, although there is no practical way of measuring the complexity of most objects of interest since there is no obvious way of finding the smallest Turing machine that generates a given object.

Sometimes it can be quite easy to assign relative, if not absolute logical depth. A picture such as the bifurcation plot of the smoothed map in the previous chapter (Figure 4.5) does have a relatively high logical depth, since in order to determine the colour of each of its pixels, the map has to be iterated for a large number of steps. Just picking a colour at random each time would be much simpler and faster, hence having a lower logical depth.

Statistical complexity. Lucidly summarised by Mitchell (2009, p. 102), the statistical complexity "measures the minimum amount of information about the past behavior of a system that is needed to optimally predict the statistical behavior of the system in the future." The methods used for dealing with this information is that of automata and grammars. It was introduced by Crutchfield and Young (1989), who observed that high entropy (white noise) as well as simple periodic patterns are statistically simple. Midways between these extremes, they argued, we should find higher complexity. They propose a quite complicated algorithm for the computation of statistical complexity from a time series, using what they call ε-machines, which are automata capable of producing equivalent time series. Briefly stated, these automata are used for the recognition of words, or sequences of symbols. Peter Grassberger's "Effective measure complexity" is a related concept which is based on, first, a coarse graining of the time series, then studying patterns in the resulting strings by constructing automata that recognise words as belonging or not to a certain language (Grassberger, 1986). As pointed out by Crutchfield and Young, the benefit of a statistical view on complexity is that, as opposed to the Shannon entropy, a random sequence has low statistical information content, and hence a low complexity. In Section 5.3.4, the application of statistical complexity to self-organisation will be discussed.

Fractal dimension. Zooming in on a fractal object, there is always the same amount of detail. It makes sense that self-similar and fractal geometry are more complex than classical geometric objects such as lines, curves and smooth surfaces. For mathematically defined fractal objects, the fractal dimension can be computed exactly. Sometimes the box-counting method is used (see Section 4.1.4), which yields the so-called *capacity*

dimension, although other methods are often preferred (e.g. Kantz and Schreiber, 2003).

Capacity for performing computation is another criterion, with notable examples from cellular automata (Wolfram, 1983). This means that some rather specific cellular automata may be used to implement Turing machines.

Self-dissimilarity was proposed as a way to quantify complexity by Wolpert and Macready (2007). They considered how spatiotemporal patterns change depending on the scale of observation. Actually, they argue that self-similarity over several scales is a sign of simplicity, since the pattern that occurs across different scales can be encoded in a short description. Clearly, most large-scale musical works are self-dissimilar in this sense, unless, perhaps, some cases of radically minimalist works.

In the context of the complexity of concepts, Jacob Feldman (2004) discussed two related measures called *Boolean complexity* and *algebraic complexity*. The concepts in question are such that they may be described by a set of Boolean features. When applied to simple geometric figures, the features might be "circle" versus "not circle"; "filled" versus "open", and so on. Bongard problems are an ideal testbed for these complexity measures, since they are comprised of two sets of figures, where the problem is to find the common trait in the first set, the second set providing counter-examples. Boolean complexity corresponds to the number of variables in the shortest formula describing a set of objects. According to Feldman, the Boolean complexity of a concept is in accordance with the difficulty subjects have in learning it.

Yet another list of definitions and properties of complex systems is provided by Sommerer and Mignonneau (2003). From their view, if the system consists of interacting autonomous particles or agents, then, in a complex system these constituents couple to each other, they learn and adapt, mutate, evolve and replicate, react to their neighbours, and organise a hierarchy of higher-order structures. Among the key characteristic properties of complex systems, they mention the following: *Variety*, in terms of behaviour and properties of the system. *Irreducibility* is exemplified with the three-body problem of Newtonian mechanics. The analytical solution of each combination of only two bodies does not contribute to the solution of the complete problem. *Ability to surprise* is surely important in the context of generative or algorithmic music, and will be further discussed below (Section 5.3.1). "The ability to surprise is not possessed by very simple and thus well-understood systems, and consequently comes to be seen as an essential property of complex systems" (Sommerer and Mignonneau, 2003, p. 93). Of course, the amount of surprise one may experience depends on the level of knowledge one has, and is a property of the beholder as much as of the system.

A promising attempt to bring confusion to an end and collect some of the buzzwords of complexity science under a single coherent framework is the contribution by Prokopenko et al. (2008). They use information theory and variants of the Shannon entropy as this common framework, and propose quantifiable and calculable measures of complexity, self-organisation, and emergence. Their endeavour is to a large part based on the work of Crutchfield and others on statistical complexity.

5.2.4 Evaluation of complexity measures

After this survey of different notions of complexity, it remains to discuss which ones of them may be suitable measures of musical complexity. Obviously, this depends on the music to be analysed. We shall have in mind the situation where there is no score at hand. Following Schaeffer, and Tristan Murail (2005), we argue that the sound can have an inherent complexity not dependent on structures of notes, but rather on micro-fluctuations.

Some of the complexity concepts presented above are more promising than others for our purposes, so we will not evaluate all the concepts or measures mentioned in the previous section. Entropy is a very versatile concept, and there are numerous ways to apply it to an audio signal. On the lowest level one could measure the entropy of the amplitude values, although this is not very useful. Spectral entropy is related to the flatness of the amplitude spectrum, and consequently has some perceptual validity (see Section 2.3.7). The permutation entropy appears to be a useful measure at least for distinguishing among certain classes of signals, including music of different genres, as mentioned in Chapter 4. It is however difficult to say, at the moment, in what sense it relates to perceived complexity. Moreover, the entropy of various feature extractors may together yield a more complete picture of the sound. The problem of how to partition the continuous values of the feature extractors must be solved in order for this approach to be practical.

Various fractal dimensions could conceivably be found by treating the sound as a vector space spanned by time-varying vectors of features. Then, ideally, the features should be as little correlated with one another as possible. This approach seems to be particularly worthwhile for feature-feedback systems, where some feature extractors are already a part of their dynamics. The capacity dimension has the disadvantage that each cell of the phase space counts as much regardless of how frequently it is visited. For example, a short transient would contribute as much as a steady state which is reached after a small number of iterations. Therefore, it would be better to use other dimensional measures that take the probability measure of a region being visited into account, such as the correlation or information dimension (Kantz and Schreiber, 2003).

If a piece of music is algorithmically composed, an alternative might be to look at the generating algorithm. Insofar as the algorithm has been formulated as the shortest possible computer programme that carries out its task correctly, this would give the Kolmogorov complexity. Thus, autonomous instruments implemented as computer programmes should lend themselves well to being quantified by their computational complexity. Quite naturally, very short programmes that are still capable of generating interesting audio output are particularly fascinating. Some "one-liners", or single line code expressions, have been used to generate music in the spirit of nonstandard synthesis, glitch and chip music; the practitioners themselves call it "bytebeat" (Heikkilä, 2011). This class of programmes have a for loop with a time variable `t`, which is modified with some arithmetical and bit-level operators such as

$$\texttt{X} = (\texttt{t} >> 3 \,\&\, \texttt{t} \,|\, 33 * \texttt{t} \,\&\, \texttt{t} >> 11);$$

where the variable `X` is then sent as 8-bit samples to the DAC at a sample rate of 8

kHz. This particular one-liner, written in C, results in a pattern that repeats only after a minute and a few seconds. The shortcomings of the Kolmogorov complexity are evident from such programmes: by changing a single constant, the output may become radically different; its complexity as measured by other means may differ although the programme length stays the same. Moreover, there may be no relation at all between the computational and perceived complexities. Still, a worthwhile goal is to have few lines of code producing interesting output, which can be stated as a ratio of low computational complexity to high perceptual complexity (of some kind that remains to be specified). Besides, the algorithmic complexity concept would be very hard to operationalise if the music were not the result of an algorithmic procedure, but just the product of the sudden whims of the composer. Finally, whereas the shortest possible programme length may be determined in some extreme cases such as these one-liners, this is often not practically possible with longer programmes.

Statistical complexity roughly deals with prediction of future states depending on knowledge of past states. In that sense, there is a loose similarity to the perception of musical passages, where memories of past moments influence the expectations of what is to come next. Like statistical complexity, the Boolean or algebraic complexities are nontrivial and not easily implemented. It is not clear how useful it would be to think of musical fragments in terms of logical concepts, although perhaps matters of stylistic invariances could be treated that way.

Surely music is different enough from statistical physics, biology or any other field from which the complexity measures have been derived to warrant its tailor-made complexity measures. Nevertheless, one reason for dwelling on this wide assortment of complexity measures is that feature-feedback systems may be regarded as complex systems in their own right.

Until Sebastian Streich wrote his thesis on measures of musical complexity (Streich, 2006), there was little work on how to estimate perceived complexity from the audio signal without any recourse to symbolic data. We have already noted how complicated many complexity measures are, and those proposed by Streich make no exception. His approach is to divide complexity into several facets, corresponding to different musical dimensions such as tonality, rhythm, timbre and acoustics, including spatial and dynamic aspects. Streich developed several algorithms for each of these facets based on psychoacoustic modelling. This division into facets makes the perceived complexity somewhat easier to handle. As an example of this strategy, some experiments on the perception of rhythmic complexity will be discussed below in Section 5.2.7.

Most of the above mentioned complexity measures would either be quite complicated to implement, or hard to make reliable as in the case of Kolmogorov complexity. Therefore, we have not yet tried to implement any of them, although a very simple complexity measure will be introduced in Chapter 7. Nevertheless, it would be interesting for future work to study the usefulness of these complexity measures for the automated evaluation of autonomous instruments. Meanwhile, let us review some of the attempts to apply complexity measures or other related objective measures to music as well as visual arts.

5.2.5 Complexity science looking at the arts

A few studies have addressed the estimated fractal dimension of visual works of art (Taylor, 2003), music (Boon and Decroly, 1995), isolated musical instrument tones and multiphonics (Bernardi et al., 1997), and even dancing (Tatlier and Şuvak, 2008). In each of these cases, the fractal dimension (specifically the capacity dimension) is typically calculated by box counting. In the case of two-dimensional surfaces such as pictures, the capacity dimension is a reasonable choice of measure, as opposed to the well-known problems associated with its use for characterising chaotic attractors (Kantz and Schreiber, 2003). The resulting estimate of fractal dimension is not very informative on its own; it becomes useful only as a way to compare different pieces of music or pictures.

Another approach related to fractal dimensions is to look at spectral scaling properties of music. The famous discovery of $1/f$ spectral distributions in music by Voss and Clarke (1978) is not without its methodological flaws, although their idea that melodic or loudness variations in music follow a distribution such that small steps are common, whereas large leaps are infrequent, often seems to be valid. What makes the results of Voss and Clarke questionable, however, is that they used time series from several hours of radio broadcasts for their measures, thus including both speech and music, and derived a single spectral distribution from that data instead of analysing individual pieces of music. A similar method is *detrended fluctuation analysis* (Jennings et al., 2004), which finds a scaling exponent from the signal's intensity variations over different time scales. The technique has been employed for distinguishing different musical genres such as techno, Javanese gamelan, and Western classical music, apparently with some success. Detrended fluctuation analysis has also been used for a measure of "danceability" (Streich, 2006).

Interesting applications of fractal dimension analysis to visual art were suggested by Richard Taylor (2003), who analysed several of Jackson Pollock's drip paintings. He has claimed to be able to demonstrate a change in fractal dimension over time, making it possible to date Pollock's paintings, and to validate the authenticity of purported Pollock paintings of unknown origin. Methodologically, Taylor's fractal studies of painting are among the most meticulous; he even simulated drip painting in a controlled way by swinging an automated pendulum with a leaking bucket of paint over a canvas. The pendulum motion could be tuned so as to produce damped harmonic motion or chaotic orbits. Visually, the chaotic orbits clearly resemble the tangles of paint in Pollock's drip paintings. One might suspect that fractals may be found in many other paintings as well if one were to look after them. But for something to be fractal, it should exhibit a scaling over a range of magnitudes; Taylor notes that many paintings, even some drip paintings, fail to have a converging fractal dimension. The next question then is, how do viewers appreciate fractal paintings?

The dilemmas of applying complexity science to art are well expressed by John Casti (1998, p. 12):

> Is there a consistent relationship between the perceived "quality" of a piece of art and any reasonable measure of its complexity? Is a complex artwork more aesthetically satisfying than one that is "simple?" To even pose this question implies that we have some type of complexity measure that is intrinsic to the piece of art itself, and which does not depend on the person observing the [...]

> artwork [...] under consideration.

Casti discusses the potential of evaluating art by its algorithmic complexity, but dismisses the idea on the grounds that it would favour totally random objects. Taking the engraving *Sky and Water* by M. C. Escher as an example, Casti instead proposes a way to measure the connectivity of parts. The procedure may look rigorous enough, but depends critically on subjective choices in the analysis of the picture. This problem is by no means untypical; all-important assumptions of what aspects are important to look at are easily hidden behind elegant formalisms.

It is a delusion to think that there should be one single, optimal complexity measure that could be used for the comparison of a broad range of works of art. Choosing one complexity measure over of another is no innocuous affair; one could always suspect the choice to be informed by predilections for certain kinds of artworks that happen to be rich in the aspect captured by that particular complexity measure. Nevertheless, once a complexity measure is chosen, it makes sense to compare paintings or pieces of music with it and pose questions such as how it relates to the viewer's or listener's perception. Casti poses the question whether a complex artwork is aesthetically preferred to one that is simple; now, if fractal patterns versus no fractals is a rough indication of one kind of complexity, Taylor (2003) argues that our preferences for fractal drip paintings may find an explanation in our likings of natural environments—mud cracks, ferns, and forests have fractal dimensions comparable to those found in Pollock paintings.

Aks and Sprott (1996) studied the preferences for images with various fractal dimensions. These images were attractors of the general 2-D quadratic map (Zeraoulia and Sprott, 2010) with randomly chosen parameters. Aks and Sprott found that, on average, their test subjects preferred images with a fractal dimension of 1.26 and a Lyapunov exponent of 0.37 bits per iteration. It is a little bit unclear what the visual correlates of the Lyapunov exponent might be, since it is a diagnostic of the dynamical instability of an orbit, whereas the fractal dimension is obviously related to how densely the plane is occupied by the attractor. A second experiment was conducted to find out about the relations between various personal traits such as creativity and analytical thinking and preferences for images of various fractal dimensions. The personal traits were assessed both by self-reported replies to a questionnaire, and by creativity tests involving several tasks such as divergent thinking.

Aks and Sprott found that people who score high on a creativity test prefer less detailed patterns (low fractal dimension), whereas those who regard themselves as being creative prefer more detailed patterns (higher fractal dimension). People who judge themselves as skilled in science prefer more unstructured patterns with a higher Lyapunov exponent. However, with as little as 11 test participants completing all parts of the study, the results are at best tentative. Furthermore, it is hard to say to what extent their findings of preferences for attractor images of certain fractal dimensions have any bearing on the preferences for visual art in general. Nevertheless, their study raises interesting questions that could be further investigated, not least in the field of music. As mentioned in the previous chapter (Section 4.2.2), Gregson and Harvey (1992) found that some subjects managed to distinguish melodies of random tone sequences from those generated by certain chaotic maps, perhaps due to differences in fractal dimension, but

they did not ask about preferences.

The way complex systems science usually treats art is highly reductionistic. We should not expect revelatory hermeneutic analyses from complexity-related art studies; their scope is more modest and mostly limited to comparisons of mutually comparable and analysable works of art. Feature-feedback systems, however, are suitably understood in terms of dynamic systems, and the conceptual apparatus of complex systems is at least partly apt.

5.2.6 On simplicity and beauty

If complexity is a slippery concept, how about simplicity? Again, as with complexity, the same thing can be simple or not depending on what aspect one looks at. For a dynamic system, a straightforward qualification of its simplicity might be to count the number of variables, perhaps weighing linear terms, square terms, and so on, with increasing weight factors. Such a measure was in fact used by Sprott in his automated search for the simplest possible chaotic systems of various kinds (Sprott, 2010). The complexity of dynamic systems may then be measured as the dimension of the attractor.

It is often harder to arrive at a simple representation of an idea than a more complicated formulation. In this respect, simplicity is related to elegance and to finding the most essential aspect. As such, there is also a connexion to logical depth. With Kolmogorov (algorithmic) complexity as the criterion, Jürgen Schmidhuber (1997) set out to generate low-complexity art or design. There are two goals for this low-complexity art: first, the drawing should "look right", or it should "represent the depicted object's essence"; and second, the algorithmic complexity should be low, that is, the length of the shortest computer programme that generates the drawing should be as short as possible; and related to this, an informed observer should be able to discern the algorithmic simplicity of such a drawing (Schmidhuber, 1997, p. 97). The actual mechanism for producing these drawings is to use an underlying grid of intersecting circles with a different radius, which yields a kind of fractal geometry. Then, arcs between crossing circle segments are chosen as elements for the drawing. Figures with round forms or wavy lines are obviously easy to produce, while straight lines can only be approximated. Schmidhuber notes that low-complexity art is hard to create; it is easier to make acceptable cartoons with higher algorithmic complexity. Clearly, this difficulty must stem from the artistic goals he has set himself, namely a realistic depiction given formal constraints.

In fact, what Schmidhuber's low complexity art is all about is not the simplicity of the resulting picture as such, but of the encoding of it. In that respect, it is directly comparable to image compression schemes, in this case basing the compression scheme on circles. Schmidhuber also seems to equate beauty with short description length. Although he admits that beauty is fully subjective and may vary from one culture or age to another, as well as across individuals, this is perhaps not the problem here. As an example of low complexity drawings, he presents a completely symmetrical woman's face. If this figure would have been encoded with rectangular or triangular blocks rather than carefully chosen circle segments, would it have appeared equally beautiful to a test panel? If not, the encoding by circle segments or other shapes definitely has something to say.

More recently, Schmidhuber (2009) has brought in concepts of *interest* and *curiosity* in addition to beauty in his algorithmic aesthetic theory. Ultimately, the theory claims to provide a qualitative explanation of some aspects of aesthetic appreciation in human subjects, although at present, this may seem overly ambitious.

The underlying assumptions of Schmidhuber's theory are as follows: All sensory observations and actions in the environment are supposed to be stored in memory. Regularity in the data should be explored by trying to find an adaptive compression algorithm (a compressor) which makes the data storage more efficient. An intrinsic reward is introduced whenever the adaptive compression scheme makes improvements. This step is associated with "curiosity". Finally, the reward for this intrinsic curiosity is to be maximised, which happens by focusing the efforts on aspects of the world where previously unknown regularities can be found; hence the steepness of the compressor's learning curve should be as high as possible.

These assumptions, when applied to aesthetics, have wide-ranging consequences which are used by Schmidhuber to explain everything from beauty, interest, creativity, art, music, jokes, and science. Although his bold conjectures are too far-reaching to be entirely plausible, the ideas are nevertheless too interesting to be dismissed altogether. Beauty, for example, is described as being proportional to the number of bits needed to encode a new observation, given the observer's previous knowledge. This conception implies that the beauty of the same object will be different for different observers, and also for the same observer at different times. Unfortunately, the assumption that beauty depends on previous familiarity (and no other causes are explicitly mentioned), appears to be taken out of the blue, even though one could find a few examples where this might be the case. The theory predicts that the beauty of an object as experienced by an observer should diminish over time and repeated exposure. It is not hard to think of counter-examples, where one discovers more beauty by repeatedly paying close attention to the same piece of music. Such details may be possible to accommodate in Schmidhuber's conceptual framework, perhaps by replacing the assumption of perfect memory storage with some forgetfulness.

The complete subjectivity of beauty, an important premise in Schmidhuber's theory, has not been taken for granted by some notable aestheticians (e.g. Dahlhaus, 1992). In his Critique of Judgement, Kant strove to show that beauty is not simply subjective, but has a *common sense* aspect; that by having taste, our judgement of something as beautiful is followed by an expectation that everyone else should share our view. Schmidhuber also appears to confound sensory pleasure with beauty, which are carefully distinguished by Kant. Pleasure is our private experience, which we would not argue about, whereas the experience of beauty is supposedly intersubjective to some degree. Although one need not agree with Kant's views, it can be argued that the concept of beauty is richer than what can be captured by the computational scheme proposed by Schmidhuber.

It is common wisdom that the degree of unpleasantness of certain sounds is not entirely subjective. For instance, many people find the squeal of a chalk against a blackboard particularly unpleasant. In a study by Kumar et al. (2008), a common trait of unpleasant sounds was found to be fast amplitude modulation (1-16 Hz) in combination with high energy content in the 2.5-5.5 kHz area. They also found the least unpleasant sounds to be baby laughter and sounds of running, bubbling, or flowing water. It would not be too

far-fetched to seek explanations of these results in ecological psychology.

Schmidhuber (2009) further proposes an interpretation of *interestingness* as the time derivative of subjective beauty. The motivation for this is that as the agent improves its compression algorithm, what previously appeared as random data now comes to appear more regular, and hence more beautiful. As a musical example, Schmidhuber reminds us that for most listeners, Schönberg's music is less popular than certain pop tunes, but those who enjoy it often have some prior musical education, perhaps including knowledge about twelve tone composition.

> The interesting musical and other subsequences are those with previously unknown yet learnable types of regularities, because they lead to compressor improvements. The boring patterns are those that seem arbitrary or random, or whose structure seems too hard to understand (Schmidhuber, 2009, p. 25).

Today's artistic practice is not necessarily concerned with attaining beauty at all. In the essay *Kalliphobia in Contemporary Art; Or: What Ever Happened to Beauty?* Arthur C. Danto (2005) notes that the invention of art that disposes with the ideal of beauty came from the Dada movement, whose art was a protest against the absurdity of a society that sent an entire generation of young men into the trenches to slaughter each other during the First World War. But the disposal of beauty has left its marks on the ensuing art history, even in more peaceful times. The artist Diether Roth is quoted as saying that if something he is working on threatens to become beautiful, he stops working on it. One of his works consists of 24 hours of dog barking, of which Danto (2005, pp. 326–327) comments:

> It gets on our nerves. It is annoying. What we are not to imagine is that someone would say, I have learned to find beauty in the constant barking of dogs. But for just this reason, I cannot imagine Roth stopping because *he* found the barking beautiful. [...] In fact, I think on the evidence of Roth's oeuvre that he recognized the same things as beautiful that everyone else does. He just did not want them to be part of his art.

The new complexity movement in music (see Section 5.2.1) also had its counter-movement—the new simplicity. Although some proponents of high musical complexity also preferred music with rough edges and some proponents of simplicity may have had a predilection for consonant and pleasant intervals, neither the aesthetics of simplicity nor that of complexity can be equated with a search for beauty—or an avoidance of it. One may imagine very simple music (say, as judged by several listeners) that is not generally held to be beautiful at all, or conversely. In Chapter 7 (Section 7.3), we will return to the questions of simplicity, complexity and aesthetic preferences in the context of a listening test. Let us just note that terms like simplicity and complexity, when applied to music, often tend to become value-laden.

5.2.7 Perceptual complexity of rhythm

Like timbre, perceptual complexity is a multidimensional concept. In music, it could apply to rhythm, melodic contour, harmony, counterpoint, timbral diversity, and possibly

other aspects. Since complexity is such a multi-faceted concept, it is not surprising that research somehow dealing with its perceptual aspects often picks out certain more delimited fields. In music psychology, there are at least some studies on rhythmic complexity that will be discussed below. The question is how to operationalise complexity so that it becomes a testable property. To this end, one could compare test subject's ability to recognise or to reproduce various sound fragments or musical phrases. Those that are hard to recall or reproduce might then be classified as more complex. This makes sense; long sequences are necessarily difficult to memorise, unless they are redundant.

There is probably no good theory of overall perceived complexity in music, but extant studies have been more or less successful in explaining specific aspects (Streich, 2006). Most attempts at quantifying the perceptual complexity of music suffer from the shortcoming of only taking notated or "lattice-based" music into account. Wishart (1996) opposed the lattice-based music to dynamic morphologic music, such as is typical for acousmatic and some improvised music. Nevertheless, the simplifications involved in restricting music to a lattice sometimes pay off in making research tractable.

A model proposed by Povel and Essens (1985) assumes that listeners perceive temporal sequences with reference to an induced clock, if possible. Their temporal patterns consisted of tone sequences played repeatedly, where each inter-onset duration was an integer multiple of a unit duration of 200 ms. Their model takes perceived accents into account; although the tones are all identical, those followed or preceded by a long empty interval tend to be heard as accented. In experiments where subjects were asked to reproduce the temporal pattern, it was found that patterns that were easy to encode according to the model also had less deviations. Povel and Essens introduced a complexity measure for their temporal patterns, which uses the clock that fits best to the sequence. All irregular subdivisions of the beat were thought to contribute to complexity, although a later study (Essens, 1995) showed that other factors such as variations in how groups of tones cluster around beats must influence the complexity of temporal patterns. Further refinements to this complexity measure were made by Shmulevich and Povel (2000). Their model uses no less than seven parameters that account for several specific cases of how a beat is subdivided. After tuning the parameters to experimental data, it is not surprising to find a good fit between predictions and observations. Let us just note how complicated it is to account accurately for perceived complexity, even in such restricted cases as repeating quantised temporal patterns.

The clock model has obvious limitations: it does not work for "irregular" rhythms with a prime number of time intervals, it can only handle rhythms with coexisting N-tuples (e.g. eighth triplets followed by eighth notes) by increasing the subdivision, and it says nothing about the intricacies of temporal deviations found in performances of music. Non-repeating temporal patterns are better represented by means other than the clock model. For instance, a tree structure can be suitable since it has several hierarchical levels corresponding to different time-spans. An attempt to measure rhythmic complexity from such tree structures has been made by Liou et al. (2009). So far their method assumes a binary rhythmic subdivision. Another approach that works directly on the audio signal, and without any limitations on the music, is the previously mentioned danceability measure of Streich (2006). A regular, strongly marked beat would score high on danceability, thus it is probably best seen as an indication of rhythmical simplicity.

Rhythmic structure has not been one of the primary concerns in the present work on feature-feedback systems. Perceived regular pulse may or may not arise, but the discretised temporal complexity measures of Povel, Essens and Shmulevich appear to be unsuitable for the unquantised timing typical of feature-feedback systems.

Somewhat related to slow-paced rhythm is the concept of structural change. Any musical dimension, such as timbre or harmony may be differentiated and serve to articulate large scale form. Thus, structural change may occur along any of these dimensions, as well as on a range of different time scales. From these ideas, Mauch and Levy (2011) introduced a complexity-related measure of structural change on multiple time scales, which was based on feature extractors related to timbre, rhythm and chroma. Their high level structural change feature quantifies the change as it occurs on different temporal levels, from highly localised to more extended processes. The structural change seems to be one of the most promising features for use in automatic complexity measurements of autonomous instruments, but its application for that purpose has to be left to future work. Nevertheless, some simple measures of non-stationarity may be a good point of departure, as will be further discussed in Chapter 7.

5.2.8 Aesthetic preference

A study of preferences for compositions of varying complexity was conducted by Heyduk (1975). Four piano pieces were written for the study; they differed in their number of chords and use of syncopation, but otherwise followed a similar form scheme. First the test subjects rated the compositions according to how much they liked it and how complex it sounded to them. Groups of test subjects were also presented one of the four compositions, and asked to rate their preference for this piece after hearing it once and after hearing it 16 times.

There are some interesting underlying assumptions for this study. First, the complexity dealt with here is called psychological complexity, and it encompasses various stimulus attributes (such as novelty, uncertainty, arousal), and it is a *mutable* characteristic, that is, it may change as an individual encounters the same object repeatedly. Second, it is assumed that an inverted U-shape should result when plotting preferences as a function of psychological complexity (this idea goes back to Wundt). Further, Heyduk assumed that familiarity would reduce psychological complexity. Given an optimal complexity for an individual, one should be able to predict how preferences change after repeated presentations of the same piece. If the piece is more complex than what is optimal, then the preference for it should increase; if less complex, the preference for it should decrease.

Heyduk's study gives some evidence that these assumptions are realistic. The mean rated complexity of the four pieces agrees with the intended complexity, and the mean rated appreciation shows a peak at the third most complex piece. The predicted increase or decrease in the liking of pieces that were above or below optimum complexity was also found.

Boolean or algebraic complexity was mentioned above (see Section 5.2.3). The question that Feldman (2004) pursues has to do with our reaction to simple or complex patterns. When the police detective finds that all evidence points to the same suspect, or when a mathematician finds a very short proof for a theorem, there is a compelling

simplicity of explanation. With the aid of the algebraic complexity measure, a statistical distribution can be found for sets of concepts with a given number of features. Then one can ask how likely some pattern or concept would be, if it had been generated by chance. In general, patterns with a striking simplicity are unlikely to occur by random generation, and so are extremely complex patterns. However, the difference is that patterns that have a higher algebraic complexity than the average randomly generated pattern are not as psychologically significant as are the simple patterns.

Music is notoriously multifaceted, and its complexity may arise in various domains. Dahlhaus (1992) offers an interesting discussion of how the complexity of different musical dimensions has been balanced in various historical epochs. High complexity in one domain (e.g. harmony, melody, rhythm) is often counterpoised by simplicity in others, although this is not always so; the proponents of New Complexity (discussed in Section 5.2.1) are notable exceptions.

Apart from psychological tests with limited numbers of participants, a completely different approach to aesthetic preferences has become possible recently with the emergence of social websites for music sharing. People are influenced by what they think other people like, and artists who are already popular will tend to stay popular. It is possible to predict what will become a hit just from the dynamics on such social websites (Bischoff et al., 2009). Such studies cannot help us to understand what aspects of the music it is that attracts people to it in the first place, but it is a sobering reminder of the fact that marketing and social networks play an enormous role in the shaping of popular taste.

Let us now return to autonomous instruments and their evaluation. As previously argued, autonomous instruments may be evaluated according to the complexity of their output. This evaluation may be carried out either by listening to the results and making a subjective judgment, or it may be automated using an objective complexity measure. Both of these strategies have their merits and their limits. In the end, most composers would not make music without evaluating the result by listening to it before making it available to an audience. As we will see in Chapter 8, there are exceptions in the context of generative music, where all conceivable variants that may eventually be generated cannot possibly be evaluated in advance. It should be emphasised that the criterion of perceptual complexity is proposed because some experimentation with feature-feedback systems has led to the impression that it is all too easy to generate static sounds, or textures that do not evolve much over time. Furthermore, we have argued that it is a bit wasteful to have a long and complicated programme to generate highly redundant audio output, which may as well be generated with much simpler synthesis techniques. To this end, some objective complexity measure may be useful as a fitness criterion in automated search for autonomous instruments, as long as the complexity measure also yields perceptually valid qualifications. The usefulness, as well as difficulty of such automated searches, should become evident in the next chapter.

5.3 Emergence and self-organisation

Feedback systems and semi-autonomous instruments are often capable of behaviour that the creator of the instrument did not specify or even foresee. Such behaviour is often

referred to as "emergent properties" or "self-organisation". In this section, we take a closer look at these concepts.

With the cybernetics movement researchers began addressing questions about systems in general, and to look at phenomena from an interdisciplinary perspective. Feedback processes and self-regulation were studied, and some of the grounds were laid for what was later to be absorbed into complex systems theory. In particular, this includes notions of self-organisation and emergent phenomena. Composers and musicians with an interest in autonomous or semi-autonomous instruments are often found referring to cybernetics as well as to self-organisation. This link by itself is worth paying attention to, and we will do so partly in the rest of this chapter and in the final chapter. Despite this interest in self-organisation, it has not been very common to speak about *self-organised sound*. The reason may be that usually, some ready-made system is adopted and mapped to musical parameters. Then, it is not the sound as such that organises itself, whatever that would mean, although one may hear the result of some self-organising process. Indeed, this was pointed out by Davis and Rebelo (2005), who used a model of swarming, sounding particles that were spatialised according to the position in the swarm. A more convincing example of self-organised sound is probably Di Scipio's *Audible Ecosystems*, in which any small noises get amplified and structured by the complex feedback process including signal-adaptive granulation and other processing (Di Scipio, 2003).

If it is in some sense correct to say that a feature-feedback system self-organises, it does so by adjusting its synthesis parameters to its recent output; then, the audio signals that run in its feedback loops may be said to self-organise. However, the concept of self-organisation, whilst being handy for vague qualitative descriptions, says little about what actually goes on and what kind of organisation is taking place. For a clear understanding of self-organisation, one first needs to pin down the concept of organisation, and then describe how organisation in general is distinguished from something that is self-organised. According to Shalizi et al. (2004), the term self-organisation was introduced in the 1940s by W. R. Ashby. However, the concept was already mentioned by Kant in the context of teleology in the second part of Critique of Judgement (see § 65 in particular). Emergence is frequently mentioned in the same breath as self-organisation, but they are not necessarily related. An important distinction can be made regarding whether an observer has to be in place for emergent phenomena to occur or not. Relying completely on the observer's judgement leads to the unfortunate disappearance of objective criteria. If anything, the ensuing discussion in this section will show the need for heightened terminological accuracy in questions related to self-organisation and emergence, including discussions of music.

5.3.1 Emergence

In cellular automata, a small set of simple rules can generate emergent behaviour in the sense that persistent patterns may arise without being directly specified by the rules. In the same way, short computer programmes can generate an audio file or musical score with properties that cannot easily be deduced from reading the source code. A few favourable conditions for emergence were listed by John Holland (1998, pp. 225–230):

- Emergence occurs in systems composed of large numbers of small parts, preferably

interconnected.

- The whole is larger than the sum of its parts, because of nonlinear interactions that make it impossible to predict the behaviour of the whole from the knowledge of each part alone.

- Persistent patterns with changing components may form. Some examples can be found in standing waves, and in organisms that turn over their constituent matter, but remain the same.

- The function of emergent patterns is determined by the context where it occurs. Holland gives the example of three bones with different functions in different species; they provide flexible linkage in the gill of fish, a wider extension of a reptile's jaws, and are found as the ossicles in the inner ear of mammals.

- Persistent patterns may interact and add to the "competence" of a system. Thus the competence or sophistication of response of ant colonies increases with the number of individuals.

- There are often macro-level laws that describe persistent patterns in much simpler ways than the lower-level laws.

- Similar to selection in Darwinian evolution, differential persistence is a typical consequence of laws that cause emergent phenomena. Certain patterns persist after interactions with others; some patterns are modified, others dissolved.

- Enhanced persistence makes generating procedures possible on a higher level. Taking again an example from evolution, the spontaneous formation of an eye by a random assembly of molecular parts appears extremely unlikely. But considering that the building blocks are already in place (light-sensitive and lenselike compounds, neurons), it is just a question of time before a functional combination of them occurs; hence the importance of persistence.

Although there is no consensus about how to define emergence, it is often understood as a phenomenon caused by processes on some low level, which become easier to grasp on a higher level. Emergence is also frequently associated with a holistic view; studying the parts by themselves may not reveal anything significant about how they interact. Critics have dismissed emergent phenomena as mere epiphenomena; these were thought to disappear when a causal explanation was eventually found (Corning, 2002). Reductionism and holism both have their merits, and it is quite possible to study emergence with a reductionist attitude.

> Reductionism, or detailed analysis of the parts and their interactions, is essential for answering the "how" question in evolution: how does a complex living system work? But holism is equally necessary for answering the "why" question: why did a particular arrangement of parts evolve? (Corning, 2002, p. 21).

In biology, emergence can be related to synergy, as when two species live in symbiosis. As a test of synergy, Corning proposes a practical or thought experiment, "synergy minus one" as he calls it: "one can test for the presence of synergy by removing an important part and observing the consequences" (Corning, 2002, p. 23). If there is synergy, the system will cease to function properly after one or more parts have been removed. Next, Corning (2002, p. 23) proposes a definition of emergence as a "subset ... of cooperative interactions that produce synergistic effects" and adds two important points: First, that synergies do not have to be observed or perceived to qualify as emergent effects; second, that emergent effects do not have to be the result of self-organisation, although they can be. The counter-example to the latter point is functional design or purposeful organisation, such as that of assembling a car from all of its various parts.

Currently, a global scale synergy minus one test is unwittingly applied to the ecosystem with the extinction of species. Let us assert that it is safer to apply the test to an autonomous instrument. Suppose it is a feature-feedback system with the three interconnected components of signal generator, mapping unit and feature extractor; then, the generator cannot be removed without causing the audio output to stop. If either the mapping or the feature extractor is removed, feedback is blocked, so the dynamics of the instrument will be radically different. Hence, such feedback systems pass the test of synergy, though this is of course rather obvious. Pointing out that the system is synergetic would then be equivalent to saying that it is a feedback system.

Crutchfield (1994) distinguishes two kinds of emergence: pattern formation and intrinsic emergence. The former happens when an external observer discovers a pattern; in the latter case the observers are themselves parts of the system they observe. Surprisingly, Crutchfield finds the intrinsic variety to be the only well-defined notion of emergence. This theory is stated in the formalism of ϵ-machines (as discussed in section 5.2.3). Emergence is understood as novelty, a situation where the current understanding (say, of a pattern) breaks down, and the invention of a more powerful model is required for explanation (see also Prokopenko et al. (2008)).

In music, and particularly in experimental music and electroacoustic or purely digital feedback systems, the term emergence appears every now and then as a qualifier. Here, as much as in physics, biology, or complex systems studies generally, it is warranted to ask what it stands for. To state that something is emergent is not an explanation of anything, but points to something that begs for an explanation. If we accept that music is made for listening (as opposed to, say, silent reading of scores), then it makes sense to demand that emergent effects need an observer who perceives the phenomenon. A sensation of pitch is emergent in this sense when the fundamental frequency of a sound is in the range of about 20 Hz to 4 kHz.

What is important about emergence in a musical context seems to be that it offers something unexpected to the observer. Let us see where this idea leads us. Corning included purposeful organisation as a possible route to emergent phenomena. In the purposeful organisation of musical material into a composition, the end result will be more or less what the composer intended it to be, so it should therefore come as no big surprise to its creator. Hence, thoroughly-organised music holds little surprise for its composer, although a premiere audience will of course not know what to expect and must be prepared to encounter something they may never have heard before. On the

other hand, if the composition has been made by algorithmic procedures, then neither the composer nor the audience will know in advance exactly what the result will turn out to be. In Tim Perkis' words, "the composer's position is not that different from the audience. He or she is capable of being as surprised as anyone by what actually happens in the music" (Perkis, 2003, p. 76). But if you already know what will happen because it is an encore, will not the elements of surprise be bleached? If surprise were the only criterion of emergence, then anything experienced for the second time would lose its emergent quality. Thus, the objectivity of emergent effects breaks down; that is the consequence of putting so much emphasis on the effect. Still, there are procedures of music making that are likely to offer some surprise, at least on a first hearing.

Nevertheless, this analysis indicates that emergence is quite an obtuse concept which often needs to be refined. The most important emergence-related phenomenon in the context of autonomous instruments is that simple rules, when applied repeatedly, can cause complicated patterns. Another, but rather trivial point is that musical entities at different levels emerge only at a specific time scale. For a pitch to be perceived, a few periods of the waveform need to be heard, while for shorter segments one hears only a click or a thump; a melodic contour does not exist at the level of individual notes, but requires a succession of notes. This notion where the whole is "greater than the sum of its parts" is what McCormack et al. (2009) call *gestalt processes*, which they rightly distinguish from emergent properties. They argue that a chord *could* be considered emergent since harmony is a property of the relations between tones that does not exist on the level of individual tones. In contrast, the type of chords that arise in Cornelius Cardew's The Great Learning, Paragraph 7 (see Chapter 1, Section 1.3.4) by the interaction among singers, is an example of what they classify as an emergent phenomenon. "A systemic (rather than perceptual) property that distinguishes gestalt from emergent phenomena is the *active* nature of the component parts, which allows for internally generated (rather than externally imposed) organization" (McCormack et al., 2009, p. 364; italics in original). This is a view that we cannot but agree with. Now, all we need to find out is what it means that this organisation is internally generated.

5.3.2 Self-organisation: Ashby's view

In a paper about the nervous system, and the cerebral cortex in particular, Ashby (1947) suggested that if changes of neuronal organisation are possible, then an increasing number of equilibrium states should develop. In other words, the process that is responsible for self-organisation is the increase in the number of equilibria. The mathematical model Ashby sets up is interesting, although hardly adequate as seen from today's vantage point, neither from a neurological nor from a dynamic systems perspective. In short, the nervous system in Ashby's model is described as a system of differential equations,

$$\dot{x}_i = f_i(x_1, \ldots, x_n), \qquad i = 1, 2, \ldots n$$

and the changes to this system are represented by a step-function h, which is constant over time except when it suddenly switches to another state, which influences the functional form of f_i. Hence, different organisations of the system occur for

$$\dot{x}_i = f_i(x_1, \ldots, x_n, h)$$

when h takes on different values. Now, the twist is that this system, the step function included, is to be regarded as an autonomous system, so that the changes to h depend on h itself as well as all the x variables. Then, Ashby's argument goes, the number of visited attractors (or "fields" as he calls them) will increase as different states h are active, hence the probability of reaching an equilibrium increases. So far, his argument appears to be valid from a dynamic systems perspective. But with hindsight, we now take the possible occurrence of chaotic trajectories for granted in systems of this kind, unless they are linear. Apart from that, it would be more realistic to include the environment (the state of the brain is different with eyes open or closed), and perhaps the equations should have memory, thus making them delay differential equations.

Ashby advocated an approach that would be highly abstract and general, proceeding as it were from generic results of dynamic systems theory directly to applied specific instances. This approach, though at its best quite elegant, is not always ideal for making accurate or realistic models of specific phenomena.

No less contentious is Ashby's later paper on self-organisation, where automata theory (though called "machines" in Ashby's terminology) plays a prominent role (Ashby, 1962). Actually, Ashby argues that the word self-organisation is either ambiguous, or self-contradictory. A system such as a developing nervous system in an embryo starts with separate parts that over time form connexions; although it can be said to self-organise, it might as well be called "self-connecting".

The proof that, strictly speaking, self-organisation is self-contradictory goes as follows. Ashby regards organisation as synonymous with the function that updates the automaton's state. Now, self-organisation would at least require changes of the organisation to happen, that is, the updating function must change. If an automaton has the states S, and a function $f : S \to S$ that determines the next state given the current state (we are ignoring the possibility of input here), then the function f cannot itself depend on the states, since this would only turn this into another automaton. But if it does not depend on the states, then it is no more *self*-organising.

> Since no system can correctly be said to be self-organizing, and since use of the phrase "self-organizing" tends to perpetuate a fundamentally confused and inconsistent way of looking at the subject, the phrase is probably better allowed to die out (Ashby, 1962, p. 269).

But as we know, the phrase, though still causing just as much confusion, has not silently passed away. Recently, more stringent attempts at characterising self-organisation have been made.

5.3.3 Entropy reduction in cellular automata

In the Kuramoto model, there is a process that goes from a disordered, unsynchronised state to a synchronised state if the coupling between the oscillators is sufficiently strong (see Section 4.5.4). Then, one might say that the onset of synchronisation is a particular

case of self-organisation, although it is usually just referred to as synchronisation, which is more specific after all.

Processes that go from disordered states to ordered states can also be found in cellular automata and other systems. If the transition from disorder to order is a necessary criterion for self-organisation, then it may be revealing to study this process and the conditions for it to occur. This transition can be illustrated with some of the simplest, one-dimensional cellular automata with N sites $s_n(i)$ at position $i = 1, \ldots, N$ and time n. All details for this example come from Wolfram (1983), which is also a good review of cellular automata.

Let the cellular automaton have periodic boundary conditions such that the site $s(N) \equiv s(0)$ is the neighbour of $s(1)$, and consider a three-site neighbourhood and the update rule

$$s_{n+1}(i) = \sum_{j=-1}^{1} s_n(i+j) \ (\mathrm{mod} K). \tag{5.5}$$

This is just the sum of the neighbours taken modulo K, the number of possible states at each site. For $K = 2$, this is known as rule 150 according to the standard numbering scheme. (The number of the rule comes from writing the three parents of a site as binary numbers, and marking with 0 or 1 depending on the value obtained at the next state:

111	110	101	100	011	010	001	000
1	0	0	1	0	1	1	0

Translating the lower row to a decimal number yields 150.)

Self-similar triangular patterns are produced by this and several other rules, although the actual pattern depends on the initial condition. Rules that exhibit self-similarity do so in the most striking way when the initial condition consists of just one non-zero site. On the other hand, if the system is started from a random initial condition, more irregular patterns will emerge, though clearly showing more or less traces of orderliness. Figure 5.4 compares two different cellular automata, each started from an ordered and a random initial condition. Using the sum of neighbours modulo 3 rule, the initial configuration with one non-zero site results in a pattern quite similar to that of the same rule taken modulo 2. With random initial conditions, however, $K = 2$ and $K = 3$ are visibly different; the latter case is more "noisy", although small triangles appear quite frequently.

Exact measures of the increase of order (as related to the entropy or probability distribution of states) can be found in the case of random initial conditions (Wolfram, 1983). If there were no increase of order, then at each time step the total configuration of sites should have the same statistics as the initial configuration. The distribution of states should be similar, and so should the likelihood of obtaining continuous runs of sites with the same state. Starting from a uniform probability distribution of initial states, rule 150 (5.5) conserves the probability distribution. However, with rule 22, which also produces self-similar triangles similar to those of rule 150, the states rapidly converge to an unequal distribution of approximately $p(s = 0) = 0.65$ and $p(s = 1) = 0.35$. Thus, considering only the probability distribution of states, rule 22 shows the signs of organisation in a way that rule 150 does not. Some rules result in all states becoming

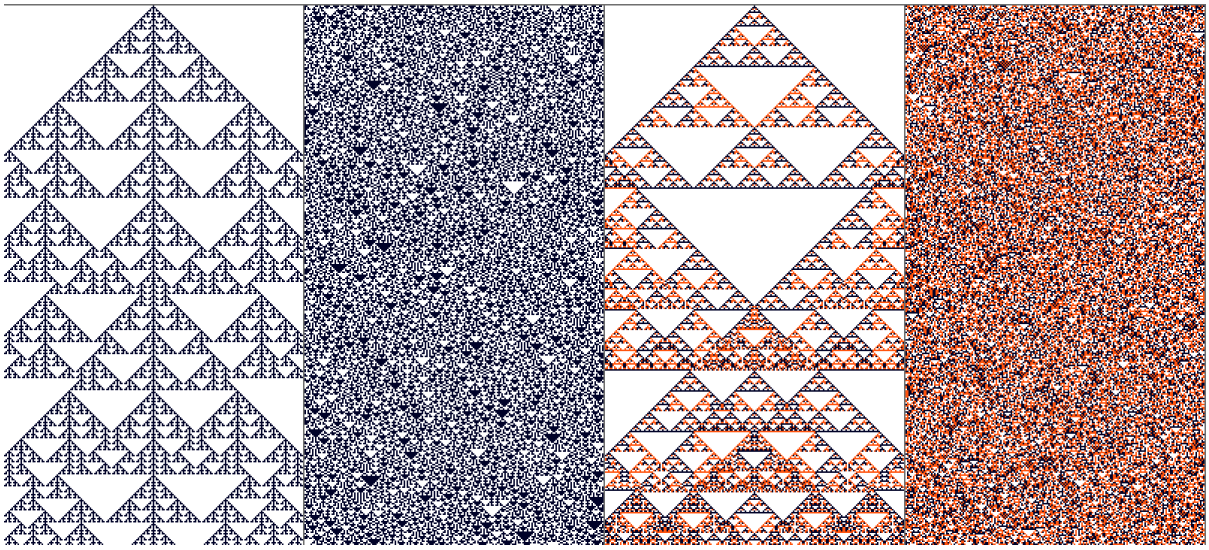

Figure 5.4: Cellular automata with the sum of three neighbours modulo K rule. Time runs from top to bottom. In the two on the left half $K = 2$, and on the right $K = 3$, where K is the number of states. The first and third panel from the left were started from a single non-zero site, the others had a random initial condition with an equal probability of each state.

zero; this configuration is of course very different from the initial random state, but it does not seem warranted to speak of self-organisation in such cases. To say that a cellular automaton is an example of self-organisation does not say very much. Organisation can potentially be defined in many ways, even in the simplest forms of cellular automata; hence there is scope for refinement of the terminology.

Cellular automata may also be compared to dynamical systems, specifically as a discrete counterpart of partial differential equations (Toffoli, 1994). In the case of one-dimensional automata, the configuration at each time step can be regarded as one number, say, a base K number $\alpha(n) \in [0, 1)$ that is mapped into a new configuration $\alpha(n + 1)$. In the limit of infinitely many cells, the configuration of all cells may be thought of as analogous to a real number, although it does not make sense to fix any site as the "most significant" or "least significant" digit. However, a metric may be defined simply as the number of sites that differ. Then, the similarities with dissipative chaotic maps can be seen. Sensitive dependence on initial conditions may be shown by starting the automaton from two nearby initial conditions, such as changing the value at a single site. Attractors in maps may be compared to the fact that, for cellular automata starting from an arbitrary initial condition, only a small subset of all possible configurations are ever visited.

Cellular automata with a finite number of sites N and states K must necessarily be periodic with a maximum period of K^N, though the actual period is often shorter. This is in stark contrast with maps and flows, where trajectories may be chaotic or quasi-periodic. At least it may appear so, though with limited resolution floating point representations, iterated maps will also have some period, albeit extremely long, at which

a nominally chaotic orbit repeats. Nevertheless, the real-valued representation of maps makes them more tractable for sound synthesis.

Now, there is something to learn from cellular automata concerning self-organisation, or rather entropy reduction. When a cellular automaton is started from a random configuration, the accessible phase space volume shrinks asymptotically to some limit. The same process can be witnessed in other dissipative dynamical systems, although the concept of "random initial condition" is not directly translatable to maps or flows. If instead a map is initiated from a set of different initial conditions, then they all converge to the same attractor unless they belong to different basins of attraction. As the system evolves, some of the initial states may become unreachable.

Actually, there is no consensus that the shrinking of the accessible phase space is a case of self-organisation. Counter-arguments will be given next.

5.3.4 Self-organisation with statistical complexity

A new proposal for a definition of self-organisation that attempts to be mathematically precise, experimentally applicable and in line with our intuitive understanding was given by Shalizi et al. (2004). First, "organisation" needs to be formalised. Thermodynamic entropy, being "proportional to the logarithm of the accessible volume in phase space [...] has no necessary connection to any kind of organization", according to Shalizi et al. (2004). Thus, they refute the idea that self-organisation should be described as a decrease of entropy, and provide a few counter-examples from physics and biology. Instead of conceiving of self-organisation as entropy reduction, they propose that organisation corresponds to an increase in complexity, and more specifically statistical complexity (Grassberger, 1986; Crutchfield and Young, 1989).

The details for the computation of statistical complexity are quite complicated, although the basic idea has to do with making maximally accurate predictions about future states of the system given its past states. One then finds equivalence classes (causal states) of past states, such that they all lead to the same future configuration.

Specifically, the *mutual information*

$$I[x;y] = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)} \tag{5.6}$$

is used as a measure of the information about the variable x contained in y (Prokopenko et al., 2008). Mutual information uses the joint probability distribution $p(x,y)$, and is symmetric in its two variables. Let x^+ be the future state, and x^- the past configuration (referred to as *light cones*), and let the equivalence classes $\epsilon(x^-)$ be such that they all lead to the same x^+; then the statistical complexity is defined as

$$C = I[\epsilon(x^-); x^-], \tag{5.7}$$

which can be estimated by clustering causal states that have similar future light cones (Shalizi et al., 2004). In other words, the statistical complexity quantifies the amount of information about the future state that is contained in the past state. All that remains to do is to observe whether C increases or decreases over time; an increase in complexity

implies that the system has organised. Further, if the rise in complexity is not due to external causes, then the system must have self-organised.

Here we should note that autonomous instruments do not take any external input, and if they can be shown to organise in the sense of increasing statistical complexity, then it must obviously be a case of self-organisation. In systems with input, it is clearly more difficult to judge whether or to what extent a system has self-organised. In principle, this measure of complexity and self-organisation should be applicable to autonomous instruments, but there are lots of details to fill in here as to exactly how it would be used. First, the past and future light cones need to be specified. This concept is familiar in physics, where it denotes the points in four-dimensional space-time that are reachable from an initial coordinate. Here, reachable means that one cannot travel faster than light, or in other contexts, faster than some maximum speed of information propagation.

A circular cellular automaton was used as an illustration of the self-organisation measure by Shalizi et al. (2004). Depending on its rules, different levels of complexity were attained, but in all cases the complexity leveled out after some time. Consequently, there is only a rise in complexity in this system during an initial transient, so self-organisation only occurs in the beginning of the process, after which the system reaches a stable complexity level. This complexity equilibrium must not be confused with fixed points of the system's dynamics; it happens for periodic oscillations as well as turbulent behaviour.

5.3.5 Swarming

Flocks of birds, schools of fish, or swarms of insects are all intuitive examples of self-organisation. Each individual only has to follow simple rules for their movement, but the swarm as a whole appears as a coherent entity. The sufficient motion rules are: move close to your neighbours, but avoid collision, and try to match your velocity with that of others. Such rules have also been used for the computer simulation of swarms. Davis and Rebelo (2005) used such a simulation and spatialised the particles of the swarm using 8 loudspeakers. Swarms are easy to grasp as they are seen in a spatial configuration, but it is unclear how we can hear the emergent patterns, they contend. The swarm's particles are represented as sound waves each spatialised to its own position; the constructive or destructive interference of these sound waves is then a concrete example of an emergent sonic phenomenon.

According to Blackwell and Young (2004), swarming behaviour shares some traits with improvised music, notably in the way musicians interact. This similarity was the conceptual basis for a swarming algorithm implemented by Blackwell and Young for interactive use in improvisation. Particle positions in the swarm are translated into musical parameters and sent to an audio or MIDI output; likewise, audio or MIDI input is sent to the swarm model.

There is no direct mapping from what the musicians play to the behaviour of the swarm, rather the input from the improvising musicians is interpreted in the swarm module as attracting points in the space of the swarm particles. This can be seen as a form of *stigmergy*, which is an indirect communication mediated by traces left in the environment. Communication between ants does not only happen by direct contact, but also through stigmergy, where the strength of pheromone trails provides clues about

where to find food. The behaviour of the swarm in the model of Blackwell and Young is then mapped to sound, thus providing response to what the musicians are doing. This leads to a flexible tripartite model of semi-autonomous instruments, or *live algorithms*, where there is an input "listening" unit, a patterning algorithm f which may be the swarm model, an evolutionary algorithm, neural network, chaotic system or other model, and an output unit which maps the dynamics of the algorithm f to sound (Blackwell, 2007).

Blackwell and Young (2004) cite four criteria for self-organisation: positive and negative feedback, amplification of fluctuations, and multiple interactions. Positive feedback occurs when the swarm approaches an "attractor" (the term is not used exactly as in dynamic systems theory, it is more like an attracting force such as gravity). Since the particles are driven apart by a repulsive force, there is also negative feedback. The same repulsions cause fluctuations which may be amplified by interacting musicians. All particles interact by being attracted to the centroid of the swarm, that is, towards the center of its mass.

The four criteria necessary for self-organisation may appear straightforward in the context of swarms, but it is not obvious how they apply to feature-feedback systems. There is feedback, but in what sense could we say that it is positive or negative? Such terminology is imprecise; the feedback takes place through some nonlinear mapping function. Amplification of fluctuations is certainly an important part of what happens in feature-feedback systems. Multiple interactions occur between particles in a swarm model, and there is more than a handful of particles in a swarm. In contrast, an autonomous instrument may be constructed with a small number of parts (three if one counts the signal generator, the analysis unit and the mapping), each of which is only connected by one input and one output. Self-organisation is obvious when it happens in a swarm: it has to do with how each particle begins to follow the average movement. Nothing like it happens in systems that are not composed of large numbers of similar parts. If several units of autonomous instruments were coupled together, then swarm-like behaviour could be designed. However, coupled oscillators are not regularly envisaged as flying around in some space and having varying distances between them, hence the spatial aspect of swarms would somehow have to be translated to oscillator configurations. Synchronisation of coupled oscillators is not a good analogy since it does not include the repulsive force between swarm particles.

Self-organising swarms used in musical improvisation gives an idea of how structures emerge from lower temporal levels up to larger levels. This leads Blackwell and Young to speculate:

> There is a tantalising possibility that interpretation could take place only at the smallest perceivable level, the micro-level, and that musical structure at every level upwards could arise through self-organisation. (Blackwell and Young, 2004, pp. 135–136).

Indeed, the generation of musical structure on high levels from decisions on low levels is an important goal for the present work on autonomous instruments, but it is hard to attain in a musically interesting way.

5.3.6 Adaptive systems in music

Adaptation figures most importantly in evolution biology, but often the concept is borrowed in disciplines that take evolution as a metaphor, such as evolutionary algorithms and artificial life. Adaptive audio effects and adaptive or feature based synthesis were discussed in Chapter 3. These are typically implemented in feedforward structures, with an input signal submitted to feature extraction and used to control synthesis or effect parameters. Adaptive filters may be mentioned here as well; the idea is to adapt filter coefficients in such a way that noise is optimally removed (Proakis and Manolakis, 2007). Another application is automatic tuning adjustments in just intonation. In a keyboard with twelve notes per octave, compromises have to me made; it can be tuned with perfect intervals in one key, but the more distant keys one tries to play in, the more out of tune it becomes. One solution is to analyse the played notes in real-time, and adaptively adjust the intervals to just intonation (Sethares, 2005).

Generally, adaptivity has to do with agents in environments. The environment imposes constraints, and the agent has to adapt to those constraints. So, in automated computer accompaniment, it is the computer that has to adapt to the human musician, although adaptation certainly goes both ways in that case.

Artificial life deals with agents in environments, often by computer simulations, but sometimes using physical agents such as robots (Langton, 1995). One hopes to find out things about biological life forms by studying highly simplified models. Although it has been accused of being a "fact-free science" (see Horgan, 1995), artificial life has the benefit of allowing researchers to test hypotheses that could never have been tested in real life, either for practical or ethical reasons, such as studying conditions for a species to go extinct. Artificial life has also found applications in the arts.

An interesting use of artificial life as a model of musical creativity (as well as a tool for composition) was discussed by Todd and Miranda (2006). Music-making is a social activity, they contend, hence they develop the ideas of collective music-making by artificial agents. This can happen at three distinct levels. First, the agents may move around in their environment, and their activities, whatever they may be, are sonified. For instance, the agents may move over the coordinates of a plane, avoiding collisions, and distinct patches of the space may be interpreted as specific notes which are played when the agent moves over it. The swarming model discussed above might also be an example of this level, particularly if run without the intervention of human musicians. At the second level, the individuals produce musical output, and the quality of it determines their fate. In practice, evolutionary computing is used. As Todd and Miranda say, it has proven difficult to obtain interesting musical output by running the evolutionary algorithm with coded fitness criteria. The alternative is to have the user listen through every generation of musical material and pick the individuals that survive to the next generation. This can be time consuming, but leads to more interesting results. The third level introduces *critics* beside the composer individuals who generate musical output. These are associated with birds; male for composers and female for critics. Here too, the population evolves by selection. The male's songs consist of a number of notes, and the female critics analyse their note transition probabilities. According to Todd and Miranda, interesting results were obtained when the critics enjoyed being surprised. Here, surprise

means to hear an unlikely note transition. But the result is not that the male birds generate random note patterns, because expectations have to be built up in the first place for surprise to occur. In other words, the entropy of the note transitions should not be too high and not too low.

Adaptivity is a concept that seems better suited to semi-autonomous instruments and other open systems than to autonomous feature-feedback systems. However, one could develop systems of coupled autonomous instruments that may react and adapt to each other. This is a path that we have not pursued in any depth yet, because first, there is much to find out about the dynamics of individual autonomous instruments.

5.4 Discussion

Complex systems science does provide a number of definitions of measurable complexity. Among the most useful notions are various entropies, algorithmic or Kolmogorov complexity, statistical complexity and algebraic complexity. These are generic complexity measures which may not be suitable for characterising music. On the other hand, researchers in music psychology have studied the effects of various levels of musical complexity, where the concept of complexity has usually not been explicitly defined. Instead, relative complexity levels have been rated by human experts (see Sections 5.2.7 and 5.2.8).

Some kind of perceptual complexity, we have argued, should be a plausible criterion for the evaluation of autonomous instruments. This may seem to be a purely aesthetically motivated choice, but there are logical grounds as well. Generating musical material that remains interesting over several minutes with an autonomous instrument is no easy affair. If perceptual complexity is a plausible success criterion, it is so because too simple behaviour is so easily obtained. What is too simple is not only a question of what may be interesting to include in a composition, but also of achieving an output from the autonomous instrument that is on a par with the instrument's sophistication. In other words, simpler and more controllable synthesis techniques can preferably be used to generate simple material.

A similar view regarding the evaluation of performances of his *Audible Ecosystemic* works is espoused by Agostino Di Scipio, who thinks that aesthetic principles as such are not the crucial point:

> A good performance of any of the *Audible Ecosystemics* works is when as many different system states as possible have been gone through, provided they are audibly revealed to the ear as different nuances of sound, as variations in texture, in timbre, in pace and density of gestures (Di Scipio, 2011, p. 103).

Perceptual complexity is an informal notion, although psychological studies may reveal some conditions that make certain stimuli susceptible of being perceived as simple or complex. Schaeffer's typological classification of sounds into categories of too simple, too complex and balanced could conceivably be applied to the output of an autonomous instrument. This is certainly a subjective evaluation, which moreover is likely to depend on the sound's duration. Whereas a short sound fragment does not have the time to become boring, any prolonged sonic texture is more likely to appear static and to be

unable to hold on the listener's interest. In effect, this informal notion of complexity has to some extent guided the evaluation of feature-feedback instruments in the following two chapters.

Apart from evaluations of the resulting music, another reason for dealing with complexity is that it shows us the way into more precise definitions of emergence and self-organisation than are usually encountered in the discourse around some forms of electronic music. Feedback systems in music are sometimes claimed to show emergent behaviour, or the systems are described as self-organised, with little reflexion about the meanings of those concepts. There are exceptions, of course, such as the writings of Di Scipio and a few others (Di Scipio, 2003, 2008, 2011; McCormack et al., 2009).

Depending on how one defines it, self-organisation could be something that almost surely happens in most feedback systems. The entropy reduction or shrinking of attainable regions of the system's state space as demonstrated in cellular automata is a case in point. On the positive side, it is a straightforward concept, which is easy to measure. Ashby's view on self-organisation—that it is best to forget the phrase altogether—is not very helpful if we want to understand the phenomenon. Shalizi, Crutchfield and others have developed the concept of statistical complexity, and demonstrated how to use it to diagnose self-organisation in some simple systems such as cellular automata. From their point of view, self-organisation is an increase in statistical complexity.

A possible source of confusion about self-organisation is that "organisation" may refer to two different things. Organisation is either seen as a temporal process in which something messy gets sorted and entropy decreases (or statistical complexity increases), or it could be seen as the final orderly state. It seems to be very common to confound these two notions, the noun and the verb sense of self-organisation. The process of organisation may just be a short-lived transient in a dynamic system, whence the final, orderly state is what one typically observes.

In the next two chapters, we introduce a few new autonomous instruments. There will be no quantitative assessments of the complexity of these instruments, with a single modest exception using a new measure related to feature extractors. There does not appear to be any shortcut that would allow making perceptually relevant objective complexity measurements in a simple way. As said in the beginning of this chapter, a rigorous theory of self-organised sound with autonomous instruments might one day be possible, but we are not there yet.

In the present chapter, we have considered several systems that qualify as open systems, in contrast to the closed feature-feedback systems that will be studied in the following two chapters. Closed systems such as autonomous instruments are easier to study than open systems, because they are easily controllable and a thorough analysis of the dynamics becomes feasible. On the other hand, the design of interesting autonomous instruments is hard because no external input, voluntary or accidental, may push the system into more promising behaviour if that should be needed. Acoustic spaces and the random inputs that influence open systems are highly complex, even in the absence of a musician, as exemplified in some of Di Scipio's works.

For the sake of clarity, it is desirable to keep synthesis models simple when they are analysed. Designing realistic autonomous instruments, if not deliberately making them crude, often results in complicated code. The feature-feedback systems that will be

presented next have not been constructed with a view to serve as full-fledged generators of musical compositions, but rather as slightly simplified examples, some of which definitely yield promising results. Still, many of these instruments are far from simple. Clearly there is a dilemma; excessive simplification does not do justice to the musical potential of these instruments, but presenting realistic and complete instruments would imply an intractable complexity.

Chapter 6

Analysis of Parameter Spaces

The synthesis and analysis of sound often stand in a chicken-and-egg relation to each other, especially so with complicated synthesis techniques that are not modelled on any known phenomena. Therefore, it is appropriate to begin the study of feature-feedback systems with some relatively simple models and analyse these systems as a function of their components and parameters. Then, we will look at progressively more complicated systems and discuss relevant ways of analysing them.

Mastering an acoustic instrument is a long and complicated process. The beginner makes all sorts of squawks and squeals in the process of learning how to control the tone production of an instrument such as the violin or the clarinet. A similar process of trial-and-error faces us in the first experimentation with autonomous instruments. Far from all ways to build such an instrument lead to any interesting sounds; some do not even produce sound at all. If we go along with an open-minded attitude, it may so happen that those first squeals and squawks manage to influence our taste a little bit, so that we find ways to make use of them as musical material. Virtually any sound can be used for musical purposes (in order to eschew radical conceptualism, it helps if it is audible); this is the consequence of the so called liberation of sound that occurred throughout the twentieth century. This point may be useful to bear in mind, even though the problems of composition will not be fully addressed until the final chapter.

With the investigations of the present chapter we hope to demonstrate some useful ways to explore autonomous instruments, to map out their sonic behaviour as a function of their control parameters. Picking up the thread from Chapter 4, the autonomous instruments will be treated as dynamic systems, so we can look for fixed points, periodic cycles and strange attractors, as well as various bifurcation scenarios. These analysis methods form the basis for the following chapter, where the goal will be to bridge the gap from experimental investigations to purposeful sound design with autonomous instruments.

This chapter begins with some theoretical considerations of dynamic systems and chaos. Then follows a number of case studies of some particular feature-feedback systems, each of which will introduce new methods for analysing parameter spaces. Many of the methods are generally applicable, but carrying out the same analysis of each of the systems introduced in this chapter would be tedious. However, some methods are best suited to specific systems. Thus, one can try to study relatively simple systems such as

the cross-modulated oscillator in section 6.2 using familiar techniques of maps, whereas the noise driven oscillator in Section 6.4 is better tackled with statistical methods, and the huge number of coefficients in the mapping of the wave terrain oscillator in Section 6.5 necessitates random sampling of the parameter space.

6.1 Theoretical issues

Before undertaking any detailed studies of feature-feedback systems, we will describe exactly how the theory of dynamic systems can be applied. The first step is to formulate generic feature-feedback systems as a set of equations. Then, a simple oscillator with feedback from a pitch estimator is introduced in order to have a concrete example to hinge some of the theory upon. The state space is an important concept in dynamic systems, and being able to identify what constitutes the state of an autonomous instrument is crucial for further investigation. Even the notion of an initial condition needs to be carefully formulated before it is possible to estimate Lyapunov exponents by the method that will be proposed in Section 6.1.8.

6.1.1 The feature-feedback system equation

Let us begin by writing the equation for a general form of feature-feedback systems. The audio signal x_n is generated by a synthesis technique $\mathcal{G}$ which is controlled by some time varying parameters π_n. A mapping $\mathcal{M}$ updates the parameters based on the output of the analysis unit $\mathcal{A}$ which uses one or more feature extractors, and which in turn is a function of the L last generated output samples. Thus, the equation takes the form

$$x_n = \mathcal{G}(\pi_n, n) \tag{6.1}$$
$$\pi_{n+1} = \mathcal{M}(\pi_n, \phi_n) \tag{6.2}$$
$$\phi_n = \mathcal{A}(x_{n-1}, x_{n-2}, \ldots, x_{n-L}) \tag{6.3}$$

with the parameters $\pi_n \in \mathbb{R}^p$ and the features $\phi_n \in \mathbb{R}^q$ being real vectors or in many cases scalars. The output signal x_n will typically be in mono, although it may have any number of channels. It will usually be assumed that the signal's amplitude range is limited to $x_n \in [-1, 1]$, so that clipping of the output signal will occur in case it should exceed that range.

The mapping (6.2) is a function $\mathcal{M} : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^p$; the analysis unit with q different feature extractors is a function $\mathcal{A} : \mathbb{R}^L \to \mathbb{R}^q$, and the generator is a function $\mathcal{G} : \mathbb{R}^p \times \mathbb{N} \to \mathbb{R}^c$ for an output of c channels. In (6.3), the input signal is assumed to be mono. Despite the simplicity, this form covers most of the systems we will deal with in this chapter. More complicated systems can be built by various extensions and modifications; such systems will be constructed in the next chapter.

Note that an explicit time dependence is introduced in (6.1). For the purist, this cannot be an autonomous system since the signal generator is a function of time. However, the signal generator is typically some kind of oscillator such as $x(t) = \sin(\omega t)$, where it makes sense to mention its dependence on the time variable. All periodically driven

systems, such as $\dot{x} = f(x) + \sin t$, may be written as an autonomous system by turning the time variable into a new dependent variable by inserting a new equation $\dot{t} = 1$; hence it is only a matter of viewpoint whether one chooses to call the system autonomous or forced. However, the time dependence may be nondeterministic, as is the case if the system is driven with noise; then the system is no longer a deterministic dynamical system in the sense that its future behaviour depends uniquely on its current state. Nevertheless, since noise is an indispensable resource in computer music, we will also investigate a noise driven system (see Section 6.4), even if it means that we have to leave behind the methods of deterministic dynamical systems.

The mapping (6.2) is written as a recursive function with a time-varying parameter ϕ. The extracted feature ϕ of course comes from the same system through a complicated feedback path; that is, in the end it depends on previous parameter values $\pi_{n-k}, k \geq 1$. Likewise, the feature extractor operates on a window of L past samples x_n, which ultimately depend on past output feature values $\phi_{n-k}, k \geq 1$. Apparently, there is no privileged position in this chain where it all starts. In other words, each line of the complete system (6.1–6.3) may be considered as primary, with the other two serving as auxiliary variables. From a practical point of view there are certainly important differences, but for an understanding of the dynamics of the system it may be as revealing to study the time series of π_n or ϕ_n as the audio output.

Further simplifications can be made to this system. The recursive form of (6.2) can be dropped, reducing it to

$$\pi_{n+1} = \mathcal{M}(\phi_n). \tag{6.4}$$

The feature extractor can be bypassed, resulting in a mapping $\pi_{n+1} = \mathcal{M}(\pi_n)$, although this is no longer a feature-feedback system. This is the case when a synthesis model is controlled by another external recursive system. If the updating rate is slow enough, then this may be considered a case of note-level algorithmic composition. Each parameter update then corresponds to the onset of a new note. An adaptation of feature-feedback systems to note-level algorithmic composition will be discussed in the following chapter (Section 7.5).

When the complete system (6.1–6.3) is implemented in one single computer programme running on a single computer, then each of the internal variables can be made accessible to every part of the programme, which is the use case we are addressing here. Then, the necessity of a feature extractor may not be obvious. For instance, a feature extractor might be used to estimate the generated pitch. However, we already have access to the oscillator's control variable for pitch, so there seems to be no good reason to go to all the trouble of analysing the pitch from the generated signal, apart from, perhaps, assessing the performance of the feature extractor. Another case is the automated pitch control of nonlinear oscillators, as discussed in Section 4.4.3. Furthermore, there are other scenarios where a direct access to synthesis parameters is barred. Imagine a setup with several interconnected computers, each of which generates an audio stream that the other computers have access to, whereas the internal operation of each computer remains hidden to the others. Such setups would be typical of computer network ensembles, including the Hub and many others that followed their practice. That would be a generalisation of the current setting where use of feature extractors becomes mandatory.

When synthesis by feature-feedback systems is implemented in a single computer with no acoustic signal path, it might appear that the use of feature extraction could be evaded, or rather simulated. The signal generator's synthesis parameters can be made available to other parts of the instrument, and a shortcut may be taken. Then, instead of analysing the approximate pitch of a signal generator, one may access its frequency variable and read off its value directly. This would certainly be much easier, although in some cases, the relationship between synthesis parameters and feature values (let alone perceptual qualities) may not be known in advance, or perhaps there is no control parameter that matches exactly with a certain signal descriptor. However, there is yet another shortcoming with this simplified approach. As mentioned in Chapter 4, a major effect of most feature extractors is that they effectively downsample the signal by smoothing out rapid variations. This lowpass filtering effect would have to be simulated, if autonomous instruments should be constructed without feature extractors, yet closely mimicking the behaviour of the corresponding feature-feedback system.

If the mapping (6.4) is simply the identity function, then it can be dropped so that the raw values of the feature extractor are directly fed to the generator. In general, this is not a good idea, because the feature extractor may have a completely different numerical range than the synthesis parameter; besides, it is usually overly optimistic to hope that such direct mappings should produce anything of interest. Hence, the most basic function the mapping takes care of is to adapt the numerical range from the feature to the synthesis parameter. Often it serves other functions as well, such as splitting a single feature to several synthesis parameters or mapping several features to a single synthesis parameter.

Some minor extensions of the basic autonomous instrument equation will be called for in this chapter, including filters between the mapping and the generator, which implies that the generator operates on some weighted average of past parameter values. A matter of great importance is the updating rate of the feature extractor. In the simplest case, it receives a new sample and outputs a new value at the audio sample rate, but for FFT-based feature extractors, a sample rate update is less feasible (the sliding DFT that was briefly discussed in Section 2.2.2 will not be considered here, although conceptually it fits very well into the basic autonomous instrument equation).

Next, we introduce one of the simplest examples of a feature-feedback system.

6.1.2 A simple self-modulating oscillator

To begin with, we describe a self-modulating oscillator which has been partly analysed previously (Holopainen, 2009). One of the simplest possible feature-feedback systems is obtained by taking a sinusoidal oscillator and estimating the frequency from its output, and then mapping the estimated frequency back to the oscillator's frequency control parameter. Hence, we have the output

$$x_n = \mathrm{osc}(f_n) \tag{6.5}$$

for the instantaneous frequency f_n, which is obtained from some mapping

$$f_n = \mathcal{M}(\hat{f}_n) \tag{6.6}$$

of the estimated frequency $\hat{f}_n$. As discussed in Chapter 2, there are several alternatives for the frequency estimation of a single sinusoid. The zero crossing rate

$$\hat{q}_n = \mathrm{ZCR}(x_n), \tag{6.7}$$

will be used here because of its simplicity and flexible choice of window length. Here $\hat{q} \in [0, 1)$ is a normalised frequency variable, related to physical frequency as $\hat{f} = \hat{q} f_s/2$ Hz. Then, eq. 6.6 should read $f_n = \mathcal{M}(\hat{q}_n)$, and the complete system is given by the above equations (6.5–6.7).

This model may at first sight appear to be related to feedback FM, given by $x_n = \cos(\omega_c n + \beta x_{n-1})$. In both cases, the oscillator's output is used to modulate the frequency variable. However, one should not fail to notice two huge differences; in feedback FM, only the last output sample occurs as a feedback term, whereas in the above model, the ZCR feature extractor uses a window of some size L, over which the estimated frequency is averaged; moreover, the feedback is applied to the instantaneous phase in feedback FM, but to a more slowly varying frequency control parameter in the above system.

Preliminary experiments have revealed that the simple self-modulating oscillator (6.5–6.6–6.7) is hard to push into a regime where it shows more complicated behaviour than more or less long-lived transients that eventually settle on a fixed pitch. As discussed in Chapter 4, the analysis window of a feature extractor contributes with a smoothing of the dynamics. Hence, there appears to be a trade-off between analysis window length and degree of nonlinearity of the mapping function. The concept of "degree of nonlinearity" would need to be made more precise, although a crude qualification for functions on the real interval might be to count the number of preimages of the map (see Section 6.2.4). For now, let us study some concrete examples.

For instance, consider using the linear map $m : \mathbb{R} \to \mathbb{R}$ given by

$$m(\hat{q}; C, F) = (1 + C\hat{q})F \tag{6.8}$$

in (6.6), with non-negative constants C and F, where C is the coupling strength from the estimated frequency, and F determines the lowest possible frequency. For $C > 0$, the frequency should increase until it hits the Nyquist limit (for this example we have used $f_s = 48$ kHz). In practice, there is a limit for C below which the frequency does not increase as far as predicted, instead it sticks at some lower level. In Figure 6.1, left part, it can be seen that this transition occurs somewhere near $C = 20$, where the resulting frequency rises sharply if the ZCR window is short, and more smoothly if the window is longer. As C increases above this critical value (with the other parameters constant), the frequency variable becomes unstable and increases "without bound", which in this case means all the way up to the Nyquist frequency and further on, thus resulting in aliasing. Given the inherent frequency limits of a discrete time system, it is possible that for some mappings the system will reach a limit cycle close to the Nyquist frequency. Likewise, oscillation death can happen if the DC frequency is attracting. Most of the curves in Figure 6.1 are remarkably smooth, except for in the right part of the figure, where the curve corresponding to $F = 8$ kHz becomes quite irregular above $C = 50$. The reason for the irregularities may be the effects of transients, so letting the system run for more than two seconds as used here might be necessary to allow the transients to die out.

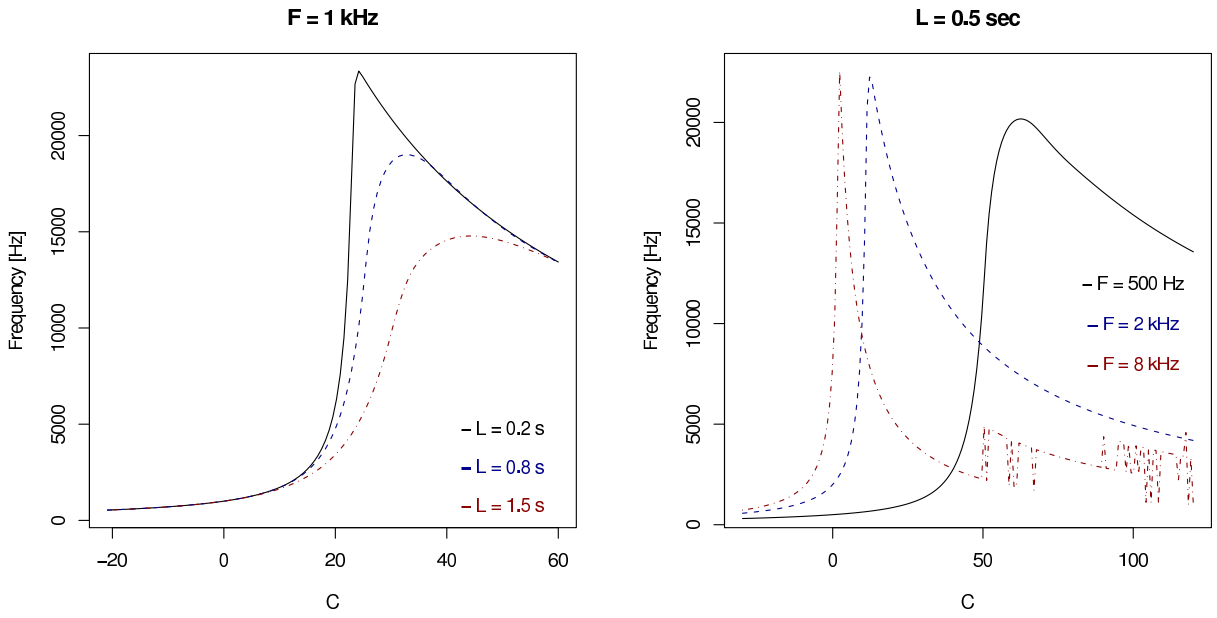

Figure 6.1: Measured frequencies with the linear map (6.8) as a function of C, taken as averages of ZCR (in Hz) over four seconds after an initial transient of two seconds. Left: $F = 1\ kHz$, which is attained for $C = 0$ as expected. The sharp curve (solid black) was obtained with a ZCR window of $L = 0.2$ seconds; the effect of increasing the ZCR length to 0.8 or 1.5 seconds is reflected in the two curves below. Right: $L = 0.5$ seconds is kept fixed, and the three curves correspond to different frequencies: $F = 500$ Hz (solid black), $F = 2$ kHz (blue dashed line), and $F = 8$ kHz (dotted-dashed red line) showing some irregularities above $C = 50$.

The map (6.8) is about as simple as can be, yet the actual behaviour of the system cannot easily be predicted from it. Since the oscillator gradually settles on a stable frequency, the system cannot be suspected of being chaotic. Actually, it appears to be difficult to produce chaos with this system, even with highly nonlinear mappings, but this example with a linear mapping should serve as a warning that things are not always as simple as they seem.

6.1.3 Spectral bifurcation plots

Autonomous instruments may have many parameters whose influence upon the sound is not known in advance of experimentation. The problem is to acquire a synoptic overview of how the sound behaves as a function of the parameters, preferably without having to listen to lots of sounds in search for a usable parameter region.

A construction analogous to the bifurcation diagram may be revealing: instead of plotting the collection of amplitude values as a function of the parameter, the spectrum is plotted as a vertical slice for every parameter value. This technique, apparently first used by Lauterborn and Cramer (1981), will be referred to as the *spectral bifurcation plot*. In ordinary bifurcation plots, the first few iterations are omitted in order to avoid plotting transients. However, in a sonification setting, these transients could potentially

be audible—in fact they would be all there were to hear if the system settles on a fixed point. Hence, it makes a difference where in the course of iterations the time window for spectral analysis is positioned. For many kinds of dynamic systems, feature-feedback systems included, the transient may go on for a long time or it may be very short-lived. Regardless of its length, it is practical to set a fixed duration into the sound after which the spectrogram is computed. Spectral plots are not very popular in studies of chaotic systems, and for good reason: a chaotic orbit usually does not converge to one particular amplitude spectrum over time. Therefore, it matters where the time window is positioned, and moving it to the next non-overlapping position might change the appearance of the spectrum fundamentally. This fact must be remembered; it means that we cannot know for sure if the chosen spectral slice is representative of a larger portion of the sound or not. However, if the sound is periodic and the periodicity is robust to small parameter changes, this will show up as smoothly varying partials in the plot.

Now, what information may be gained through the spectral bifurcation plot? It should be noted that such diagrams are quite different from sonograms of the system as the same parameter is swept through time. Suppose a musician is controlling the parameter p by sweeping it from its lowest to its highest value and we plot a sonogram of the ensuing sound. This procedure is similar to, but not identical with that of plotting the spectral bifurcation diagram over the same range of the parameter p. When plotting bifurcation diagrams, the map is usually initialised with the same value for each parameter value, and as noted, the start transient is usually skipped. In contrast to this, the sonogram of a sweeping over the parameter range will show the effect of running the system with a slowly varying parameter, which makes it a non-stationary model. The sonogram may show transients or the effects of hysteresis, especially if the parameter changes swiftly. Hence, a spectral bifurcation plot maps out features of a dynamical system as a function of a parameter, as they appear some fixed amount of time after starting the system from one and the same initial condition.

Since spectral bifurcation plots may look indistinguishably similar to sonograms, the difference might be clarified if we consider how long durations of sound one would actually need to generate to produce the spectral bifurcation plot as compared to a sonogram. If an initial transient of, say, five seconds is skipped and if thousand different parameter values are plotted, one would have to generate about one hour and twenty minutes of sound, whereas for the sonogram, the same parameter range may be swept across during a few seconds, or any arbitrarily short duration. The sonogram of such a parameter sweep is illustrated in Figure 6.2 with the oscillator of eqs. (6.5–6.6–6.7).

Example 6.1. Here we use the squared cosine mapping

$$m(\hat{q}; C, F) = F \cos^2(C\hat{q}), \tag{6.9}$$

and the bifurcation parameter sweeps linearly across a very broad range of values, $C \in [-500, 1500]$. The other parameters are $F = 1$ kHz and $L = 0.01$ seconds for the ZCR length (with $f_s = 48$ kHz). As can be seen in the sonogram, at 2.5 seconds there is only a pure sinusoid corresponding to the parameter value $C = 0$. For this system, the ZCR length turns out to be important as it has a direct effect on the period length

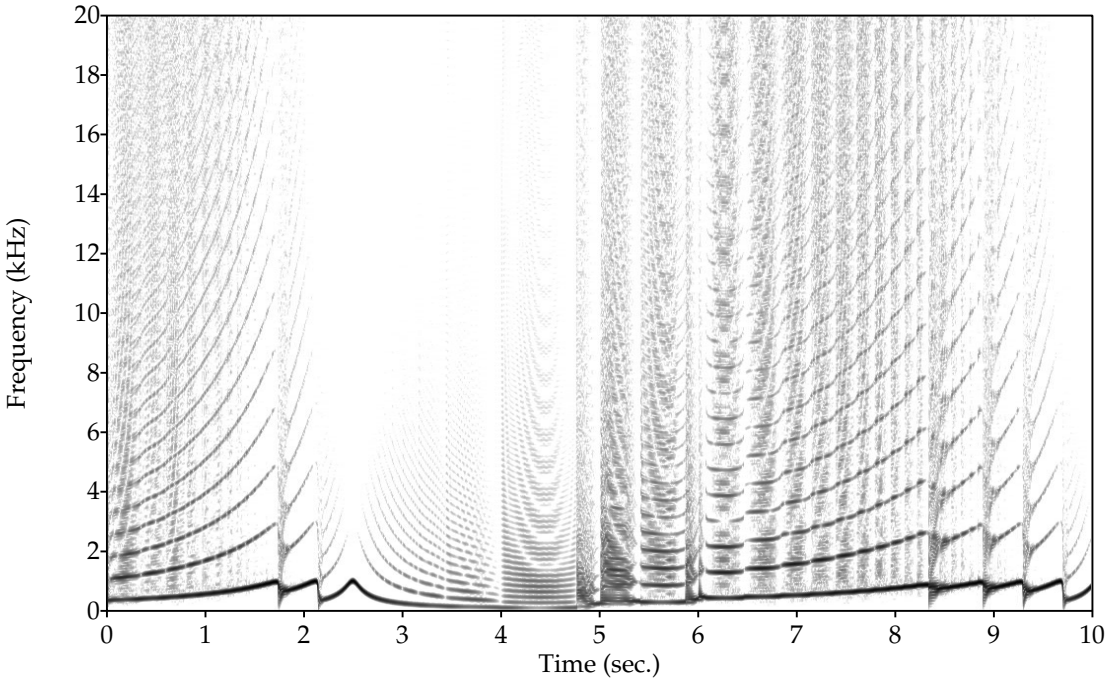

Figure 6.2: Sonogram of the simple feedback oscillator as the coupling parameter $C \in [-500, 1500]$ is increased linearly over a duration of 10 seconds. The other parameters are $F = 1$ kHz, $L = 0.01$ seconds, and the squared cosine mapping (6.9) was used.

of oscillations. Listening to the sound, it appears as though overtones of a common fundamental were successively excited, but with some extra glissando added to it. This implied fundamental is located at about 50 Hz, whereas $L = 0.01$ seconds corresponds to a fundamental of 100 Hz.

The spectral bifurcation plot of the same system and parameter values appears in Figure 6.3. Unfortunately, the spectral slices here are not obtained with the same analysis parameters as those of the sonogram. In particular, the gray scales used in these two figures are not exactly the same. Nevertheless, a comparison should be possible since we are primarily interested in more conspicuous differences than subtle variations in the partials' amplitudes. Some similarities can be seen across the figures, such as the peak of the fundamental frequency at $C = 0$ in Figure 6.3 which is easily identified with the peak at 2.5 seconds in Figure 6.2. Apart from that, there are many obvious differences between the two figures that clearly illustrate the need for being careful with the interpretation of spectral bifurcation plots (the warning applies as much to sonograms of hysteretic feature-feedback systems with time-varying parameters).

For a typical FFT window size, say, 1024 points, each window represents a very short fragment of sound. In cases where the system has a high degree of variation on a longer time scale (in other words, high spectral flux), the particular time segment chosen for plotting in a spectral bifurcation diagram matters. Because of hysteresis, the sonogram might look very different if a parameter is swept from low to high, rather than from high

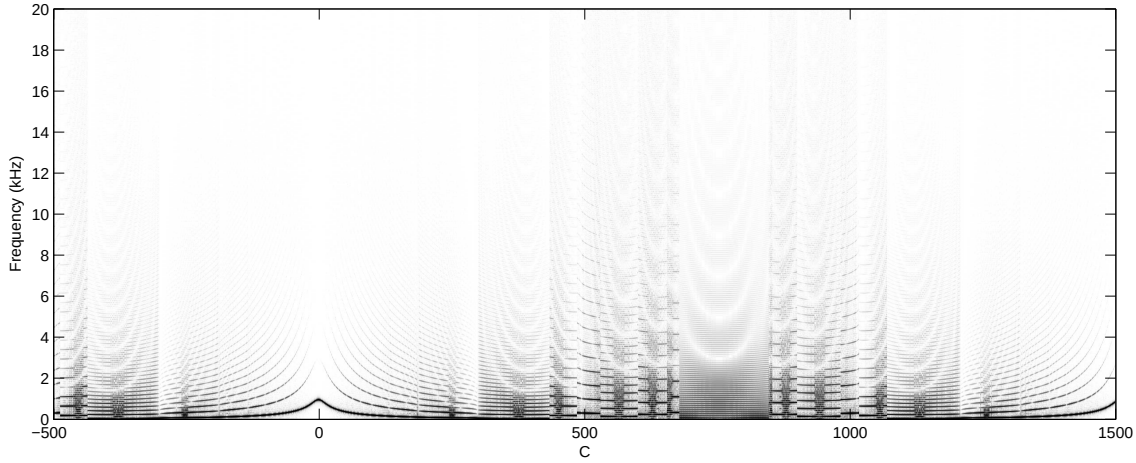

Figure 6.3: Spectral bifurcation plot with $C \in [-500, 1500]$, $F = 1$ kHz and $L = 0.01$ seconds, taken about five seconds after the beginning using a 2048 point FFT with hanning window.

to low. If there were no hysteresis, the only difference would be to reverse the time axis of the sonogram. For such reasons, the spectral bifurcation plot may nevertheless be preferable.

Another conceivable method of visualisation is to make animations of spectral bifurcation plots. Then, each frame would show a complete bifurcation plot for the entire range of parameter values, and the frames would be taken from successive time windows in the sound. Clearly this would be extremely costly in terms of computation.

Apart from spectral bifurcation plots, we shall also have occasion to plot various features in a similar manner. This can be very informative since several complementary features can be displayed in parallel. Still, the feature extractors operate on local time windows and are not necessarily representative of larger portions of the sound, unless time averages are used.

6.1.4 What is an initial condition?

In any dynamic system with an N-dimensional state space, an initial condition is a point $x_0 \in \mathbb{R}^N$ at the initial time from which the system evolves. Similarly, initial conditions can be defined for autonomous instruments. One might study the computer programme and identify each variable that has to be initialised for the system's dynamics to be specified (uninitialised variables is a common programming bug in some languages). Then, the variables that are needed to specify an initial condition need to be distinguished from any other parameters that have to be set before the programme is started. The crucial difference is that the latter parameters, including the sample rate f_s, are *constant*, whereas the variables defining an initial condition will be *time variable*. Sometimes we will study the effect of varying such user defined constants. Potential terminological confusions may arise from the contrasting conventions of dynamic systems, where parameters are usually

constant over time, and synthesis models, where the control parameters are most usefully time-variable. Whenever it matters, we will try to be specific as to whether parameters are constant or not.

Now, consider the simple self-modulating oscillator (6.5–6.7) again, and all the data needed to specify its initial condition. First, the oscillator $x_n = \sin(\theta_n)$ has to have its initial phase θ_0 specified. Then, the phase value has to be incremented by $\theta_{n+1} = \theta_n + 2\pi f_n / f_s$, so the initial frequency f_0 must be known. The mapping is a memoryless function which needs no initialisation. The feature extractor, in this case zero crossing analysis, uses a buffer that stores the last L samples, implemented as a delay line. If the current sample x_n was a zero crossing, a counter is incremented by one; if the delayed sample x_{n-L} was a zero crossing, the counter is decremented by one. Hence, the delay line needs to be initialised with L values. It is common practise to initialise the delay line uniformly to zero, as is usually done with filters. The output of a filter with zero input but a nonrelaxed (non-zero) initial state is called the zero-input response (Proakis and Manolakis, 2007). The zero-input response is not always discussed in elementary expositions of digital signal processing, nor are nonrelaxed initial states used very often in practice; but since we need to be precise about what constitutes an initial condition, the initial state of the feature extractors must also be considered.

Suppose that the delay line of a feature extractor is initialised to some arbitrary set of numbers. An intuitive argument leads us to believe that certain initial conditions of the feature extractor cannot be reached again as the system is iterated. Consider the RMS amplitude extractor implemented as a delay line, $A_n = \langle x_n^2 \rangle^{1/2}$, with the average taken over the last L samples. Now, the RMS unit stores the input signal x_n in a sample buffer $a_n, n = 0, \ldots, L-1$, which is usually initialised to zero. Then, for an arbitrary input signal x, the sample buffer will just contain a copy of the past L values of x. Now, suppose we insert the RMS unit in a feature-feedback system where the output of the oscillator $x_n = \mathcal{G}(A_n, n)$ somehow depends on the RMS amplitude A_n. For any plausible signal generator $\mathcal{G}$, there will be signals of length L that it cannot produce. (It is certainly possible to use a universal signal generator capable of synthesising any signal one wishes, but for the synthesis techniques actually employed here, such a flexibility is unrealistic.) Thus, if we manage to find one of those impossible signals and initialise the sample buffer a_n with it, we know that it will never enter the sample buffer again. This is of course no proof of the assumption that certain initial configurations will be unreachable. It would be necessary to show that a typical signal generator is unable of producing arbitrary sample sequences. However, the existence of attractors that occupy a limited volume of state space provides another argument. There may be initial conditions that do not lie on the attractor; hence, these initial conditions will not be reachable after a few iterations of the system.

Perturbations of initial conditions is an essential part of Lyapunov exponent estimation. Recall from Chapter 4 that the divergence of two trajectories started from infinitesimally separated initial conditions is used to calculate the greatest Lyapunov exponent. Translating this into the variables of the simple feedback oscillator, we might pick an initial condition from the frequency and phase variables. Thus, we take as initial conditions the points $\pi_0 = (\theta_0, f_0)$ and $\pi_0' = \pi_0 + \epsilon$ and ignore the delay line, hoping that its exclusion will not matter. In fact, it is well known that the greatest Lyapunov

exponent will be found for an initial displacement in almost any direction (e.g. Tél and Gruiz, 2006). Only if one should happen to put the displacement exactly on the stable manifold will the two trajectories approach each other, but for reasons of limited numerical accuracy, this may not go on for very long.

For estimations of the whole Lyapunov spectrum, an N-dimensional phase space volume must be monitored as it evolves. Evidently this is not a feasible approach in feature-feedback systems where feature extractors typically contribute hundreds or thousands of nominal dimensions (or degrees of freedom—of which there are as many as there are delayed samples), even though the dimension of the attractor may be quite low.

Coexisting attractors with different basins of attraction is another reason for studying the effect of varying the initial condition. If the dynamic system has several attractors that can be reached just by small changes of an initial condition, then, choosing an arbitrary initial condition will engender dynamics that are very hard to predict, even in some approximate sense. Thus, robustness to small changes of initial conditions implies a higher degree of predictability.

6.1.5 What is an attractor?

As we just mentioned, an initial condition may be specified in such a way that as the system evolves, it will not be able to return to the state it was initialised from. This has an immediate consequence for the character of the sounds produced with feature-feedback systems: they often possess a prominent initial transient, after which the dynamics observed during that initial phase will not be observed again. The reason for this is the existence of an attractor in dissipative systems (see Section 4.1.3), which occupies only a small subset of the entire state space. Next, we briefly summarise the concept of an attractor; for a fuller treatment, see the literature on the topic (e.g. Elaydi, 2008; Eckmann and Ruelle, 1985).

Consider a dissipative map $f : \mathbb{R}^N \to \mathbb{R}^N$ and a set $\Omega \subset \mathbb{R}^N$ such that $f^k(\Omega) \subset \Omega$ for all $k \geq 1$. Then, Ω is called a *trapping set* with respect to this map, but unlike an attractor it may contain such things as repelling fixed points or cycles. However, as the set of points $\{x : x \in \Omega\}$ is iterated, this set gradually approaches an attractor A. The attractor is then the smallest set of points that remains invariant under iteration of the map. Equivalently, if the initial point x_0 belongs to the attractor, all its forward iterates also belong to the attractor, which is then defined as the set

$$A = \{x_k : x_0 \in A,\ x_k = f^k(x_0),\ k \in \mathbb{N}\}. \tag{6.10}$$

The basin of attraction $B(A)$ is the set of points in state space that will end up arbitrarily close to the attractor A under iteration of the map. As long as the initial point is within the basin of attraction, the orbit will finally approach the attractor after an initial transient whose length will depend on the initial condition.

The importance of the initial condition for the dynamics is highlighted in cases such as the *expanding* logistic map $f(x) = rx(1 - x)$ with $r > 4$ (see Figure 6.4). The initial conditions of the logistic map are drawn from the interval $I = [0, 1]$, but as some points are mapped outside of this interval, they will wander off to infinity. In fact, this happens

sooner or later for almost all initial conditions, but there remains a set of exceptional initial conditions that will stay trapped in the interval. The set of remaining points,

$$A = \bigcap_{k=1}^{\infty} f^k(I) \tag{6.11}$$

is known as a strange repeller.

In this context, it is worth mentioning transient chaos, and the phenomenon known as "escape from almost attractors" (Eckmann and Ruelle, 1985) or *chaotic saddles* (Tél and Gruiz, 2006). When running a chaotic system for a finite time, it is not always easy to guess whether it will stay on a strange attractor forever or if it will sooner or later escape from it and approach some regular (periodic or fixed point) behaviour. Chaotic transients can be found in the expanding logistic map, before the orbit escapes through the top, but in other systems it is also possible to have chaotic transients that land on finite fixed points or periodic cycles. Apparently, low-dimensional maps usually have rather short-lived chaotic transients, except near bifurcation points. In the context of feature-feedback systems with their high dimensional state space, longer transients may typically be expected; this has also been found in practice, as will be demonstrated below.

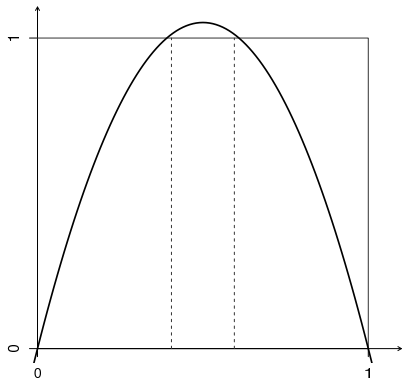

Figure 6.4: Expanding logistic map. Most points eventually escape to infinity.

When the dynamics is chaotic in the sense that there is a positive Lyapunov exponent, the attractor takes on the complicated fractal structure known as a strange attractor. However, not all strange attractors are accompanied by chaotic dynamics. There are exceptional cases, such as oscillators driven by two incommensurate frequencies, which are not chaotic but nevertheless have strange attractors (Grebogi et al., 1984). According to the definition given by Grebogi and his coworkers, a strange attractor is not a finite set of points, and it is not piecewise differentiable. In other words, it consists of an infinite number of points, which do not belong to simple geometrical objects such as connected and smooth curves or planes. Eckmann and Ruelle (1985) also provide an example of a non-chaotic attractor with fractal structure, namely that obtained with the logistic map at the accumulation point after the first sequence of period doubling bifurcations; however, they take the strangeness of an attractor to be a property of its dynamics, i.e., a strange attractor according to them has sensitive dependence on initial conditions.

So, although some rare counter-examples exist, it is usually the case that a strange attractor is the result of chaotic dynamics. Driving a dynamic system with quasi-periodic

oscillations does make it more complex. An example of quasi-periodicity will be given towards the end of this chapter (Section 6.5.5).

For a feature-feedback system with, say, two feature extractors ϕ_1, ϕ_2 that map to a number of synthesis parameters, one could plot the time series of ϕ_1 against ϕ_2 and thus get a visual representation of the system's dynamics. In a 2-D map with variables x and y, the attractor can be visualised precisely by plotting all points (x_n, y_n) during some time $N_0 < n < N_1$. Feature extractors of course represent their window length worth of information, whence it may be misleading to use them as illustrations of attractors. Nevertheless, if the dynamics is periodic, the plot of (ϕ_1, ϕ_2) would show some closed orbit, whereas chaotic dynamics would be seen as a complicated tangle, but less irregular than a set of completely randomly distributed points.

In maps, a fixed point evidently satisfies $f(x^*) = x^*$, but for a map that is constructed from a Poincaré section of a flow, its fixed point corresponds to a periodic orbit in the flow. Often it makes sense to think of feature-feedback systems as flow-like, because their variables tend to change smoothly. Therefore, it is reasonable to think of any periodic signal with period p as a "fixed point" of the system, since it is a fixed point of $f^p(x^*)$; hence, we will sometimes use this imprecise terminology.

6.1.6 Notions of chaos

For an autonomous deterministic dynamic system, the typology of long-time behaviour includes periodic or quasi-periodic orbits, reaching fixed points, blowing up (though some regard infinity as an attracting fixed point), and chaos. A reasonable assumption is that, for a deterministic feature-feedback system to produce varied long-term behaviour of a kind that supports continuing attention from a listener, the system should be chaotic. This assumption is by no means self-evident, and may be wrong for at least three reasons. First, quasi-periodicity over long time-scales might also cause some kind of perpetual variation, whether it is sufficiently interesting or not. Second, there are the very long transient processes that may go on for several minutes in feature-feedback systems before eventually reaching an equilibrium, so the process may offer enjoyable listening until it begins stabilising. Third, chaos comes in many varieties, and it is not claimed that the white noise produced by most low-dimensional chaotic maps provides sufficient perceptual variation when used as a stream of audio samples. These considerations notwithstanding, a deterministic feature-feedback system with regular behaviour will eventually produce either a constant DC offset, a periodic or quasi-periodic waveform, or it will blow up. Only chaotic dynamics can provide any variation over time.

Different definitions of chaos have been proposed, sometimes with redundant criteria (Elaydi, 2008). The definitions usually agree, but contrived examples can be found where they do not. From the experimentalist's point of view, a pragmatic working definition stated in plain language such as that given by Strogatz (1994, p. 323) is quite sufficient: "*Chaos* is *aperiodic long-term behavior* in a *deterministic* system that exhibits *sensitive dependence on initial conditions*." Aperiodic behaviour rules out fixed points and periodic orbits, or everything but quasi-periodicity, chaos and noise. Determinism means that the irregularities are not caused by randomness. Sensitive dependence on initial conditions rules out quasi-periodicity, and is an important aspect, since it is experimentally

measurable as positive Lyapunov exponents.

More technical definitions of chaos have been given, and it can be revealing to review two of them as they are described by Elaydi (2008). Li and Yorke (1975) introduced a definition which has later turned out to include redundant criteria.

Suppose there is a continuous map $f : X \to X$, where X is a compact set, and there exists an uncountable subset S of X. Then, two necessary criteria for chaos are, according to Li and Yorke:

$$(i)\ \limsup_{n\to\infty} d(f^n(x), f^n(y)) > 0 \text{ for all } x, y \in S, x \neq y$$

$$(ii)\ \liminf_{n\to\infty} d(f^n(x), f^n(y)) = 0 \text{ for all } x, y \in S, x \neq y$$

The first point means that if two different points in S are iterated indefinitely, there will be times when they will be separated by some positive distance. On the other hand, as the second criterion says, there will also be times when the points come infinitely close together. A third criterion says that the greatest distance from an arbitrary point to a point that belongs to a periodic orbit will remain positive. Lastly, a fourth criterion states that f must have periodic points of all periods, whence the famous title of the paper by Li and Yorke: *Period three implies chaos*. Later on, it has been shown that the first two criteria imply the other two. There is an interesting conclusion to be drawn from the two criteria (*i*–*ii*): The points x and y need not be started simultaneously, but may represent different iterations of the same initial point. This implies that a chaotic orbit will always return infinitely close to points of the state space that it has visited earlier.

The other definition of chaos that will shortly be given necessitates the introduction of two other concepts, those of density of periodic orbits, and topological transitivity.

A well known dense set is the rational numbers; given two arbitrary rational numbers $p < q$, another rational number c can be found such that $p < c < q$. A set D is dense in an interval I, if any neighborhood of a point in I contains a point in D. Hence, if a map has a dense set of periodic points, then points belonging to various periods can be found arbitrarily close to any point in the map's domain. It was precisely this property that was used for chaos control by Ott et al. (1990), as mentioned in Chapter 4. It should be said that density of periodic orbits is an idea that applies to different initial conditions of an orbit, and hence has nothing to do with the system's behaviour under parameter changes.

Topological transitivity is related to the two criteria of Li and Yorke, but may be stated differently. If a map on an interval I has a dense orbit, then it is topologically transitive. In essence, this means that as the map is iterated, the orbit eventually visits all regions of I and come arbitrarily close to any point in I. For instance, it can be shown that the map $D(x) = 2x \pmod 1$ is both transitive and has dense periodic points (Elaydi, 2008).

According to the second definition (due to Devaney), a continuous map f on an interval I is chaotic if f is transitive, the set of periodic points is dense in I, and f exhibits sensitive dependence on initial conditions. However, the last condition has been

shown to be unnecessary since it is implied by the first two conditions. According to Elaydi, this definition generalises to other metric spaces such as $\mathbb{R}^n$.

Intermittent chaos is known to occur in the logistic map just below the period three window, and has been observed in many physical systems (Frøyland, 1992; Strogatz, 1994). It is characterised by quiet periods of small amplitude oscillations called the *laminar phase*, interspersed with wild, irregular oscillations called *bursts*. Intermittency may be a possible explanation of certain complex phenomena observed in some feature-feedback systems, where two or more distinct types of behaviour alternate over very long time scales. An example of what may be intermittent chaos will be given in section 6.5.5.

An intriguing question is whether there is some general correspondence between the functional form of a map and its likelihood of producing chaos, or the complexity of its dynamics. Nonlinear functions are evidently needed for chaotic dynamics to be possible; however, as Zeraoulia and Sprott (2010) have shown in the case of the general 2-D quadratic map which may have anything from none to six nonlinearities, both chaos and hyperchaos are possible already in certain maps with just a single nonlinearity (see also Zeraoulia and Sprott, 2008; Holopainen, 2011).

Consider a feature-feedback system written in state space form, and suppose it is given by a differentiable map $f : \mathbb{R}^n \to \mathbb{R}^n$. Replacing each nonzero entry of its Jacobian with 1 we get a connectivity matrix. Metaphorically speaking, this matrix is similar to the patch cords of an analogue modular synthesizer; it shows which units connect to which other units. As a crude measure of the complexity of a system (if not of its dynamics), we may count the ones of the connectivity matrix and divide their sum by the total number of elements. This number would quantify the degree to which the different variables of the system connect to each other. It can be shown (although we will not do so here) that delay lines written in state space form yield sparse connexion matrices with many rows (or columns) with a single non-zero entry; the same holds for feature extractors implemented with delay lines. An interesting and difficult problem would be to find out whether there is some relationship between the sparsity of this matrix and the resulting dimension or Lyapunov spectrum of the map's dynamics. A plausible conjecture seems to be that sparse connexion matrices should correspond to a low ratio of the attractor's dimension to the state space dimension of the system, but this remains a speculation.

6.1.7 An assortment of transients

Transients typically occur in the beginning and end of a note. In common usage of the term, transients in audio signals are short phenomena, although no definite limit of their durations can be given. In psychoacoustics, there may be good motivations for thinking of transients as short-lived phenomena, such as being too short to be perceived as a clear pitch. However, it should be stressed that we will consider transients of a much longer duration whenever it makes sense from a dynamic systems perspective, which means that there is some way to estimate the transient's duration. Most importantly, this means that the system eventually settles on some steady state that is clearly distinguishable from the transient phase.

In dynamic systems, usually transients can be conveniently defined and measured. Many studies of dynamic systems have downplayed the transient phase and focused on

the stationary behaviour, but there are exceptions. For an eventually periodic orbit (including fixed points) one may study the length of transients as a function of initial conditions or as a function of parameter values. Then, the transient duration τ of an observed periodic orbit x_n can be defined as

$$\tau = \inf\{n : |x_n - \rho_n| < \epsilon\} \tag{6.12}$$

where ρ_n is a periodic reference orbit and the condition holds for all $n > \tau$, and the reference orbit is aligned to the observed orbit so as to minimise the difference (Holopainen, 2011). The transient duration then depends on the choice of ϵ, as well as the initial condition and parameter value. Some interesting scaling phenomena and self-similarities have been observed in 1-D maps where transient durations are averaged over initial conditions (Hramov et al., 2004). A subtle kind of bifurcation takes place in periodic orbits as the parameter is varied. If the transient length of a periodic orbit is plotted as a function of initial value in a 1-D map such as the logistic map, certain initial values are eventually periodic, that is, they land exactly on a periodic orbit after a finite number of iterations, whereas other initial conditions are only asymptotically periodic. For asymptotically periodic orbits, the transient process theoretically goes on forever, but due to the finite measurement precision ϵ, it appears as a finite, though longer duration than the eventually periodic orbits. If transient lengths are fascinating to study already in one-dimensional maps, the transients of feature-feedback systems may exhibit some really puzzling traits, as will be shown in Section 6.2.2.

Transients leading to chaotic attractors must be handled differently than eventually periodic orbits. (Since only dissipative systems have attractors, it is pointless to look for transients in conservative chaos.) The idea of aligning a reference orbit to an observed chaotic orbit does not make sense. Suppose a large, finite number of points are recorded as the reference orbit, after a sufficiently long sequence of initial transient points has been discarded. Then, for the initial transient, the distance to the nearest point in the reference orbit could be measured, and as this distance goes down under some ϵ for a sufficient number of successive points, the transient phase can be regarded as finished. Apart from the computational difficulties associated with this approach, there is also the problem that if more points were generated on the reference orbit, these new points would fall in gaps between existing points. Thus the "nearest distance" as measured will depend on the number of reference points.

The bifurcation scenario known as *internal crisis* happens when an attractor suddenly changes its structure. Internal crises can be observed in one-dimensional maps, where below a critical parameter value μ_c the attractor consists of a set of disjoint intervals $I_1, I_2, \ldots, I_N$, and at the critical parameter value the attractor covers one single interval J (Grebogi et al., 1983). Then, for $\mu > \mu_c$ the dynamics will have a transient phase where the orbit at first occupies the intervals $I = \cup I_i$, but after some number of iterations the orbit escapes to the rest of the attractor, or the set complement $J \backslash I$. In this case transient lengths are well defined as the time the orbit is trapped in I, which can be measured.

Chaotic transients also occur in systems where the orbit escapes to infinity after some finite time, such as the logistic map $f(x) = rx(1-x)$ for $r > 4$, where a subset J_1 of initial conditions $x_0 \in [0, 1]$ is mapped out of the unit interval in one iteration; and the set of preimages of J_1, or $J_2 = f^{-1}(J_1)$ will escape after two iterations, and so on.

Chaotic scattering can be illustrated with everyday phenomena such as a kind of pinball game where a particle is shot into a region with three discs. Assuming there is no loss of momentum as the particle bounces off the discs, it can move in a chaotic way inside the trapping region, until it escapes (Tél and Gruiz, 2006). In these cases, the structure corresponding to an attractor is called a chaotic saddle. Usually it is well worth the trouble to avoid transients of this kind in synthesis models. Variables that wander off to infinity are seldom any good for sound synthesis, almost regardless of how they enter the algorithm. Transient chaos, however, does not have to exit to unbounded orbits; the exit path may as well be a stable fixed point or period.

All of the mentioned types of transients are relevant to the understanding of feature-feedback systems. However, as these systems are generally more complex than most low-dimensional chaotic systems, the range of transient phenomena is not necessarily restricted to those that have been discussed so far. We will give an example of a transient process that goes on for some 45 seconds before settling on a more stationary behaviour (see Section 6.2.3).

Transients may manifest themselves in several ways. Basically, this happens because the transients may occur in synthesis parameters. Thus, the pitch contour, the amplitude, the spectral content, or the vibrato rate may be some transient aspects of the sound. Because of the plethora of transient processes that one might encounter in a feature-feedback system, it seems wiser to devise tailor-made transient detectors for each particular system to begin with, than to try to force everything into the same mould. Fair comparisons of transient lengths across different systems of course cannot be made using different measures, but that is not a problem that will be dealt with here.

6.1.8 Estimation of Lyapunov exponents

Usually, the Lyapunov exponents of a system that is explicitly known can be calculated by simultaneously evolving the map (or flow) and its derivative or Jacobian as evaluated in the current point of the state space. When the system equations are unknown, there are many time series methods for the estimation of Lyapunov exponents. Both of these cases were mentioned in Chapter 4. However, there is a situation that falls between these two cases: the system is explicitly given and controllable, but for various reasons its Jacobian may not be known. This is the situation facing us in the investigation of most feature-feedback systems. In principle, it may be possible to derive an expression for the system equations and the Jacobian, but this can be too much of a daunting task to bother with. On the other hand, we have detailed control over each parameter value and initial condition, so a routine for checking exponential divergence of nearby initial conditions should be feasible to implement. Even better, not only are we given the final resulting output signal, but we have full access to each and every internal state variable of the system, should we want to study them. This is good news for the estimation of Lyapunov exponents. Nevertheless, in practice this route is full of pitfalls. Despite all the benefits, reliable estimation of Lyapunov exponents can be difficult through this method and numerical results must be treated with some caution.

A naive algorithm would be to pick two close initial points, x_0^1 and x_0^2 separated by a very small distance $\delta_0 = \|x_0^1 - x_0^2\|$, then the distance $\delta_n = \|f^n(x_0^1) - f^n(x_0^2)\|$ is

monitored and the system is iterated until δ_n has reached the size of the attractor and its size fluctuates around some mean value instead of increasing exponentially. Then, the largest Lyapunov exponent λ is given by

$$\delta_n = \delta_0 e^{\lambda n},$$

that is,

$$\frac{\log(\delta_n/\delta_0)}{n} = \lambda.$$

In theoretically oriented discussions, δ_0 would be infinitesimal and the limit $n \to \infty$ would be taken at this point. However, δ_0 must be a small positive constant, and the number of iterations must be limited because the separation δ_n will approach the size of the attractor. In practice, $\log(\delta_n/\delta_0)$ versus n is plotted and the slope is taken as the value of λ. For a reasonable slope estimation, the points along the line must not be too wiggly, which puts further constraints on the number of iterations that are usable in the calculation.

Estimation of the largest Lyapunov exponent is always easiest. If the orbit is chaotic, it will show up as exponential divergence of closely spaced initial conditions, until the divergence is on the order of the attractor's size. The dynamics on or near a strange attractor can be intuited by comparison with the state space around a saddle node (see Chapter 4, Figure 4.1 on page 118). Recall that a saddle node is a point x^* that lies in the intersection of its stable and unstable manifolds, with the stable manifold (the set of points that will eventually end up on x^*) defined as

$$W^s = \{x_0 : \lim_{n\to\infty} f^n(x_0) = x^*\}$$

and the unstable manifold (those points that approach the fixed point when the time runs backwards, or escape from the vicinity of the fixed point on forward iteration) as

$$W^u = \{x_0 : \lim_{n\to\infty} f^{-n}(x_0) = x^*\}.$$

Notice what happens to a point that belongs to either the stable or the unstable manifold after a slight perturbation: Points starting on the stable manifold have plenty of space around them where they will fall out of the stable manifold, and hence never reach the fixed point. The unstable manifold is more robust in the sense that even if the perturbed point should not coincide with W^u it is likely to be carried farther away from the fixed point anyway. Stable and unstable manifolds exist for periodic orbits as well as for strange attractors. In the chaotic case, and let us consider a 2-D map for simplicity, the motion along the stable manifold corresponds to the smallest Lyapunov exponent and motion along the unstable manifold to the greatest (and positive) exponent. Now, it should be evident why the lesser Lyapunov exponents are so hard to find; it is necessary to start from an initial displacement exactly on the stable manifold. The difficulty is compounded by the small size of δ_0 which has to be used for an accurate estimation of the largest Lyapunov exponent, whereas for negative Lyapunov exponents, the already small initial distance is further squeezed together, which leads to numerical problems. Increasing δ_0 works as long as one can be sure that the separation is along the stable manifold, but

as mentioned, this is hard to ensure, and any numerical errors will propagate and push the orbit off the stable manifold. There is a solution to this problem that requires the Jacobian to be known, but here we have assumed that it is unknown.

As a solution to the problem of overestimating the largest Lyapunov exponent, we may look at its dependence on δ_0 and take the minimum value it attains over a range of initial displacement vectors. More precisely, one looks at the dependence on different initial displacement vectors $v_0 = x_0^1 - x_0^2$. Thus, the estimated largest Lyapunov exponent is

$$\lambda = \inf \lambda(v_0)$$

with the infimum taken over a range of different v_0-vectors.

Fortunately, the initial condition and the direction of initial displacement may be arbitrarily varied. Theoretically, almost any choice of direction of the initial displacement should lead to an estimate of the largest Lyapunov exponent. In some cases there may be certain initial displacement vectors that give quite different estimates. Often it does not matter very much exactly which initial point is chosen from the basin of attraction, because they all yield the same Lyapunov exponent. This fact can be exploited by averaging the so called *local* Lyapunov exponents (as calculated from one particular initial point in state space) over a larger set of initial points.

Hence, writing the dependence on the magnitude of initial separation δ_0 and initial coordinate x_0 as $\lambda(\delta_0, x_0)$, it makes sense to take the averages over different x_0,

$$\bar{\lambda}(\delta_0) = \frac{1}{N} \sum_{n=1}^{N} \lambda(\delta_0, x_0^n) = \langle \lambda(\delta_0, x_0^n) \rangle \; w.r.t. \; x_0^n$$

and taking the minimum of $\bar{\lambda}$,

$$\lambda = \inf\{\bar{\lambda}(\delta_0^i)\}$$

over a large range of magnitudes of initial separation. To make things even more complicated, the direction of the vector of initial separation v_0 may sometimes matter. An example where this is the case will be given below (see Section 6.3.2).

Apparently, it can be much more difficult to find Lyapunov exponents for high-dimensional systems than for low-dimensional maps. In theory, δ_0 is an infinitesimal quantity, but numerical resolution puts a strict limit to its smallness. Assuming that one single variable from an N-dimensional system carries the displacement, values around $\delta_0 = 10^{-15}$ work well for some systems, whereas for higher dimensional maps much greater initial separations can be required. A validation of the proposed method for estimation of Lyapunov exponents by the analysis of systems with known values should be undertaken; this remains to be done before any conclusive assertions about the chaoticity of feature-feedback systems can be made.

6.2 Cross-modulated AM/FM oscillator

The simple self-modulating oscillator presented in Section 6.1.2 has a single feature extractor that is mapped to the same synthesis parameter that is captured by the feature, in this case the oscillator's frequency. Quite a different mapping results when the estimated frequency is instead mapped to another synthesis parameter that does not influence the frequency, such as the amplitude. In fact, such a mapping is not likely to produce any interesting results at all, because there would be nothing to set the frequency control parameter in motion. Hence, a more promising strategy is to use cross-coupling.

Feedback FM and the slightly less familiar feedback AM and feedback oscillators that mix both AM and FM (Valsamakis and Miranda, 2005) were discussed in Chapter 3. Here, we introduce an oscillator with another kind of cross-coupled AM and FM, but using feature extractors for amplitude and frequency. The feature extractors that will be used are the RMS amplitude, the ZCR, and the instantaneous amplitude and frequency as obtained with a Hilbert transformer. All four combinations of these feature extractors will be investigated, always with one extractor for amplitude and one for frequency.

Now, we have the oscillator

$$\begin{aligned} x_n &= a_n \sin \theta_n \\ \theta_n &= \theta_{n-1} + \omega_n \end{aligned} \tag{6.13}$$

in which the amplitude is given as a function of estimated frequency $\hat{\omega}$, and the frequency as a function of estimated amplitude $\hat{a}$, thus

$$\begin{aligned} \omega_{n+1} &= f(\hat{a}_n) \\ a_{n+1} &= g(\hat{\omega}_n). \end{aligned} \tag{6.14}$$

Before specifying the mapping functions, a few comments valid for any mapping will be given.

6.2.1 Composed maps and fixed points

Cross-modulated oscillators are a simple paradigm for varied and complex behaviour. However, finding suitable mappings from estimated amplitude to frequency and from estimated frequency to amplitude is not trivial. We will study the system in its full detail, as well as reduced to maps and filtered maps.

The functions (6.14) make this system an iterated map. The question is whether it is legitimate to simplify these mappings by substituting the previous actual amplitude and frequency values for the estimated ones; in other words, are any of the insights one might gain by studying the corresponding two-dimensional map still valid when transferred to the full feature-feedback system? If so, the system becomes

$$\begin{aligned} \omega_n &= f(a_{n-1}) \\ a_n &= g(\omega_{n-1}) \end{aligned} \tag{6.15}$$

which is further separable into the two decoupled equations,

$$\begin{aligned} \omega_n &= f \circ g(\omega_{n-2}) \\ a_n &= g \circ f(a_{n-2}) \end{aligned} \tag{6.16}$$

in which case the initial condition has to include two time steps of each variable.

However, the feature extractors necessarily operate on some window of the signal. Thus, the simplification (6.16) is not valid in general. An important consequence is that maps being prone to chaotic behaviour (provided, at least, they are nonlinear) will be smoothed—the input to the function is in some sense an average of the last input—hence chaos will tend to be suppressed. Still, there is a chance that the simplified map (6.16) may be used to locate fixed points a^* and ω^* such that $a^* = g \circ f(a^*)$ and similarly for ω^*.

The discussion will be simplified if both variables are defined on the same interval. Therefore, we normalise both amplitude and frequency to the interval $[0, 1]$, and call the normalised frequency variable q, which satisfies $q = \omega/2\pi$, where $qf_s/2$ is the frequency in Hz.

Now we introduce a mapping $\mathcal{M} : [0, 1]^2 \to [0, 1]^2$ of the form

$$\begin{aligned} q_{n+1} &= S(a_n) \\ a_{n+1} &= T(q_n) \end{aligned} \tag{6.17}$$

For the maps S and T, let us consider

$$S(x) = [\mu(1 - x)(\mathrm{mod}1)]^2 \tag{6.18}$$

as the mapping from amplitude to frequency, and

$$T(x) = \frac{1 - \beta}{1 + \kappa x^2} + \beta \tag{6.19}$$

mapping frequency to amplitude (see Figure 6.5). For the moment (until Section 6.2.4), we fix the parameters to

$$\begin{aligned} \mu &= 2 \\ \beta &= 0.05 \\ \kappa &= 120. \end{aligned}$$

These functions are somewhat arbitrary; many other functions could have been used instead. Notice that the map T causes high frequency to be associated with low amplitudes, and low frequency with high amplitude. The map S has a more complicated dependence of frequency on amplitude, but in each of the two intervals $0 \leq a < 0.5$ and $0.5 \leq a < 1.0$, increasing amplitude maps to decreasing frequency. Hence, a certain correspondence between amplitude and frequency can be designed by the choice of map, although as we shall see, things are not that simple. Since the feature extractor tends to smooth the feedback signal, it appears to be necessary to use highly nonlinear functions for the mappings if any nontrivial behaviour is to be expected.

The assumption that this system can be treated as the two independent maps obtained by the compositions $q_n = S \circ T(q_{n-2})$ and $a_n = T \circ S(a_{n-2})$ may be tested experimentally. These two composed functions each have a set of fixed points, but we must keep in mind that these fixed points actually correspond to period two orbits, since it takes two iterations to get from x_n to $S \circ T(x_n)$. A period two solution $\{a, b, a, b, \ldots\}$ may also

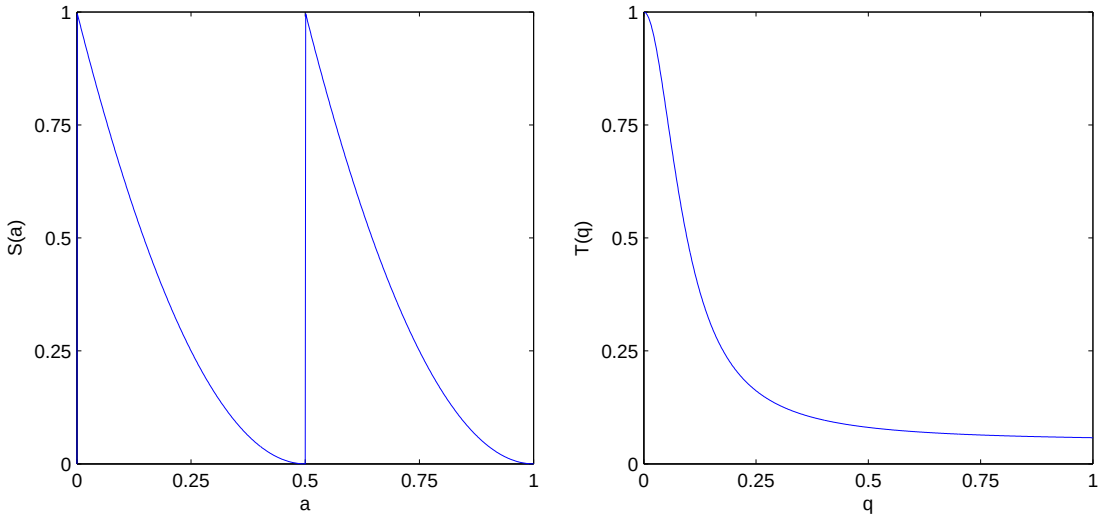

Figure 6.5: Left: the map $S(a)$ takes amplitude to frequency. Right: the map $T(q)$ takes frequency to amplitude. The lowest possible amplitude is approximately β.

degenerate into a period 1 solution if $a = b$. Whenever one of the fixed points is reached, the approximation of the oscillator as a 2-D map is valid.

The composed map

$$S \circ T(x) = \left[2 \left(1 - \left(\frac{1-\beta}{1+\kappa x^2} + \beta \right) \right) \bmod 1 \right]^2 \tag{6.20}$$

applies to frequency (see Figure 6.6). It has four fixed points (and a discontinuity at $x = 0.0962$ which looks like a fixed point on inspection of the graph, although it is not). These are the solutions of $S \circ T(x) - x = 0$, which can be found by the interval bisection method (Press et al., 2007); we have $x^*_{1,2,3,4} = \{0.0; 0.02851; 0.15022; 0.76238\}$. These numbers stand for normalised frequency, so assuming a sample rate of 44.1 kHz, they correspond to the frequencies 0.0, 628.6, 2121.8, 3312.3, and 16810.5 Hz. The sampling rate $f_s = 44.1$ kHz will remain fixed in the rest of this section. Next, let us see which ones of these fixed points are attracting or repelling. An attracting fixed point of a map $f(x)$ has $|f'(x)| < 1$.

Fixed point $x^*_1 = 0$ is attracting, and even superstable since the derivative is zero.
Fixed point $x^*_2 = 0.0285$ is repelling.
Fixed point $x^*_3 = 0.1502$ is repelling.
Fixed point $x^*_4 = 0.7624$ is attracting.

Amplitude values are given by the other composed map

$$T \circ S(x) = \frac{1-\beta}{1+\kappa[2(1-x) \bmod 1]^4} + \beta \tag{6.21}$$

with the following fixed points:

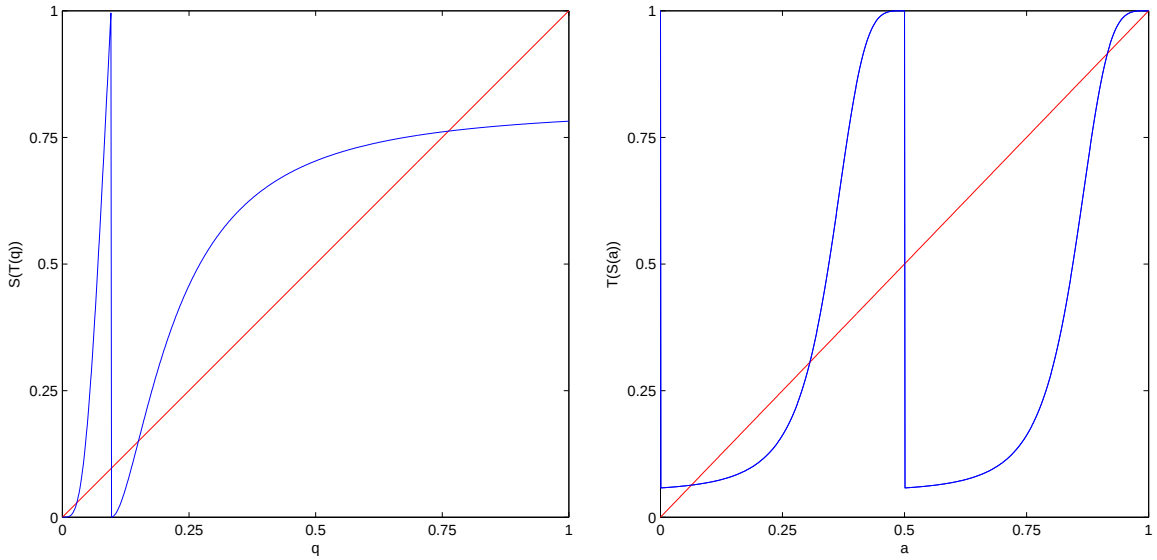

Figure 6.6: Composed maps $S \circ T(q)$ for normalised frequency, and $T \circ S(a)$ for amplitude, showing locations of fixed points where the function crosses the diagonal line.

$x^*_{1,2,3} = \{0.06343;\ 0.30621;\ 0.91558\}$ of which only x^*_1 is stable, and there are discontinuities at $x = 0.5$ and $x = 1.0$.

As the initial condition $x_0, y_o \in [0, 1] \times [0, 1]$ is varied in the map (6.17), it turns out that there are both period 1 and period 2 solutions, with period 1 occurring almost three times as often as period 2.

Now, we shall see what happens if a moving average filter is inserted in (6.17). This is not exactly equivalent to inserting a feature extractor, but it shows the effect of temporal smoothing. Then, we have the system

$$\begin{aligned} q_{n+1} &= S(\bar{a}_n) \\ a_{n+1} &= T(\bar{q}_n) \end{aligned}$$

with $\bar{a}_n = \frac{1}{N} \sum_{k=0}^{N-1} a_{n-k}$ and similarly for $\bar{q}$. Already a two-point average turns all period two solutions into fixed points. Even if the filter is applied only to one of the variables, there are only period one solutions. Longer moving average filters seem to give the same result. More interesting dynamics tends to occur for the map with feature extractors, so the approximation of the complete system in the form of a filtered map is clearly not valid in this case. In particular, the instantaneous amplitude and frequency followers oscillate around their estimated average value (see Section 2.3.2). The deviation of this oscillation may be rather large, thus feeding the system with a further signal source, as it were.

6.2.2 Dependence on window length

Now we turn to experimental investigations of the cross-coupled amplitude-frequency map (6.18–6.19). First, we will study the effects of varying the length of the analysis window. To begin with, the instantaneous amplitude and frequency will be analysed using the Hilbert transform, as explained in Chapter 2. In addition to the fixed filter length due to the Hilbert transformer (here, a 256 point FIR filter is used), moving average filters will be used to smooth the two estimated variables. Since they are independent, the two smoothing filters can be treated individually. Three cases will be considered in some detail: First, applying the moving average filter to the amplitude estimator only; second, to apply it to the frequency estimator only; and third, applying moving average filters to both amplitude and frequency. A fourth possibility is not to apply any filtering at all.

Running the cross-coupled amplitude-frequency map with instantaneous amplitude and frequency estimators but no smoothing filters, the resulting sound is coloured noise with a resonant peak of small amplitude centered around 7 kHz. No dependence on initial amplitude and frequency values is apparent in this case, except that starting from zero amplitude results in silence. Instantaneous frequency estimation does not really work for noisy sounds, such as this.

i) Smoothing of amplitude only Next, if only the amplitude variable is smoothed with a moving average filter of order L, timbral changes occur gradually as L increases. The sound remains noisy, but becomes more coarse-grained and reminiscent of boiling water or rain for longer filters. For short filters of order $L \ll 200$ samples, there is a spectral formant around 7–8 kHz, whereas for filter lengths of about 200 samples, there is instead a deep notch around 8 kHz. The amplitude of the spectral peak is about 6 dB for short filters, and the notch to peak level is about 18 dB for filters of order 200.

When analysing the sounds with several feature extractors, it was found that the feature values remained quite stable across a range of filter orders up to 200. The mean values over filter lengths 1–200 samples are, for spectral entropy: 0.96; flux: 0.25; centroid: 0.44. The voicing varies more, but generally assumes low values, or about 0.1. These values were obtained by FFT analysis with a 1024 point window; furthermore, each value is the time average over several successive FFT windows. There are obvious audible variations in the sound as the filter length varies, but these variations are probably not well captured by any of the chosen features since the trend for most features is very small, with the exception of flux which shows a slightly increasing trend.

ii) Smoothing of frequency only If the frequency variable is smoothed instead of the amplitude, we get quite a different scenario. For a two-point average, the result is rather similar to no smoothing, although the noise is slightly more coarse-grained. When the filter is increased to third order or more, a dramatic change occurs. There is a short initial transient (about 30 ms) in the form of a noise burst, followed by a quasi-periodic oscillation with lower amplitude at approximately 12.2–12.4 kHz for filter orders up to 200 samples.

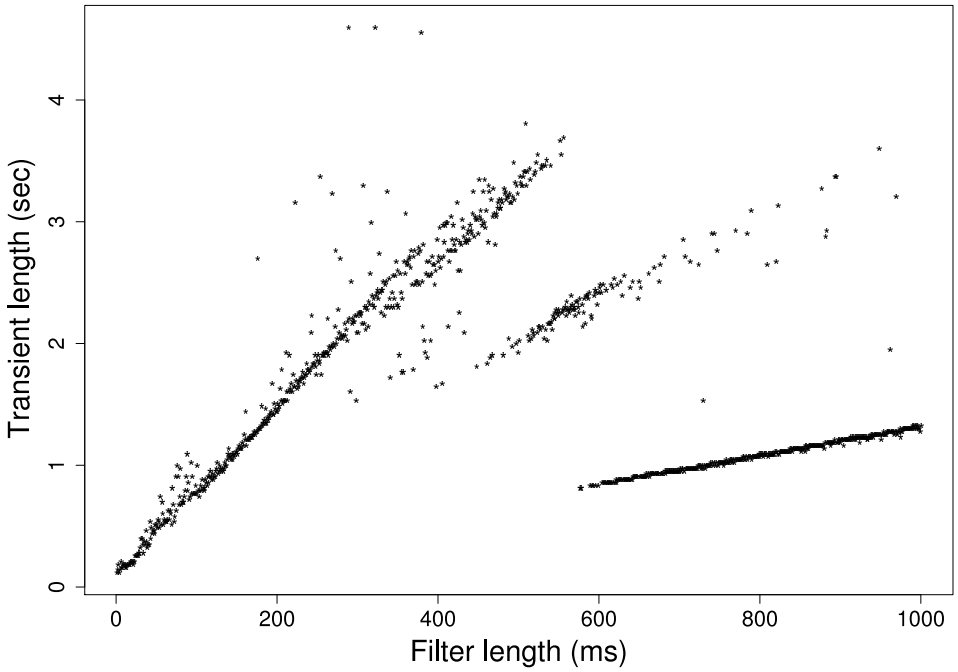

Figure 6.7: Transient length as a function of filter length in the cross-coupled map. Both the amplitude and the frequency variable are smoothed with moving average filters of equal duration, from 0 to 1000 ms.

iii) Smoothing both Finally, when both the amplitude and frequency variables are smoothed with filters of equal order, the dynamics are essentially the same as for frequency-only smoothing; that is, there is a short initial transient followed by periodic oscillation at 12.4 kHz. The difference is that the transients differ more in length, but not in a linearly increasing way; furthermore, the oscillation is more focused and has a narrower spectral peak.

The transient part can easily be isolated by visual inspection of the waveform. Then, by measuring typical values of various relevant feature extractors at the transient and the stationary part respectively, we can set up a criterion for the length of a transient. We use averages over the four latest frames of flux $\hat{\Phi}$, voicing $\hat{v}$, and spectral entropy $\hat{H}$ as follows: If

$$\langle\hat{\Phi}\rangle < 0.1,\ \langle\hat{v}\rangle > 0.8,\ \text{and}\ \langle\hat{H}\rangle < 0.4 \tag{6.22}$$

all hold simultaneously, then the transient is assumed to be over and the steady state has begun. The transient length is quantised to the nearest number of 1024 point windows. In Figure 6.7, there appears to be at least three subpopulations with linear dependence of transient length on filter length; these have different slopes and intercepts, but there are also a few outliers. Consequently, if one varies the filter length incrementally, there will be critical points where just a slight increase in filter length causes a much longer (or shorter) transient than what is typical at similar filter lengths.

6.2.3 Substitutions of feature extractors

Apparently, there is no long term interesting dynamics beyond the initial transient in the cross-coupled map with the particular mapping functions and feature extractors used so far. As a supplement to the instantaneous features, we could use RMS for amplitude and ZCR for frequency estimation, in any combination. In fact, the combination of RMS and ZCR is qualitatively similar to the case discussed above where moving average filters were used on instantaneous amplitude and frequency.

For the combination of instantaneous amplitude and ZCR there is only a short click transient followed by a quasi-periodic oscillation at 16,780 Hz with a secondary spectral peak at 6.5 kHz which is some 30 dB below the main peak. These resonances are there regardless of the length of the RMS extractor. According to the prediction using fixed point analysis in section 6.2.1, the map (6.20) should have a stable fixed point at 16,810 Hz, which is in close agreement with the observed frequency.

A much more interesting case is when the instantaneous frequency is combined with RMS amplitude. Figure 6.8 shows four spectral features as a function of RMS window length. Note that no filtering is applied to the frequency variable.

Typical sounds start with a short downwards chirp followed by a wiggling pitch profile with noise superimposed on it. As the RMS length increases, the pitch profile stays unstable or wavering, but it changes at a slower rate. For sufficiently long RMS windows, one can speculate that a steady pitch is reached after some time, but nothing can be proven about the sound's further development past the stretch of time that is generated. As an example, four sounds of two minutes duration were generated with identical initial conditions ($a_0 = 0.01$ and $f_0 = 100$ Hz) and RMS lengths of 50, 100, 150, and 200 ms respectively. All sounds have the same noisy quality, which would count as an unwanted artefact in most circumstances.

For the shortest RMS window (50 ms), the sound is dominated by rapid pitch fluctuations interspersed with relatively short stable pitches. Looking at the spectrum, it has a peculiar U-shape with most energy below 2 kHz and above 20 kHz (still using 44.1 kHz as sample rate).

Example 6.2. The case of 150 ms RMS window is remarkable (see Figure 6.9): After a short downward chirp, it starts out with a focused pitch for two seconds, which then fluctuates and mostly descends. It turns out that the first pitch is the 16th harmonic of an overtone series with a fundamental at approximately 176 Hz, and the other pitches are lower harmonics in the same series. A melodic motion on this scale eventually takes the pitch down to the harmonics 8, 7, 6, 5, and 4, and this appears to be a point of no return. What follows is a similar texture of rapid fluctuations as in the case of a 50 ms window, but interlaced with short tones at some of the harmonics, mainly below the 8th harmonic.

Very long tones at steady pitches occur for the 200 ms window. A tone may last for 35 seconds and then change, as happens twice in this case; this just shows how hard it is to tell whether the system has reached a fixed point. The strong noise that accompanies all tones may however be sufficient to prevent the system from reaching a fixed frequency and amplitude.

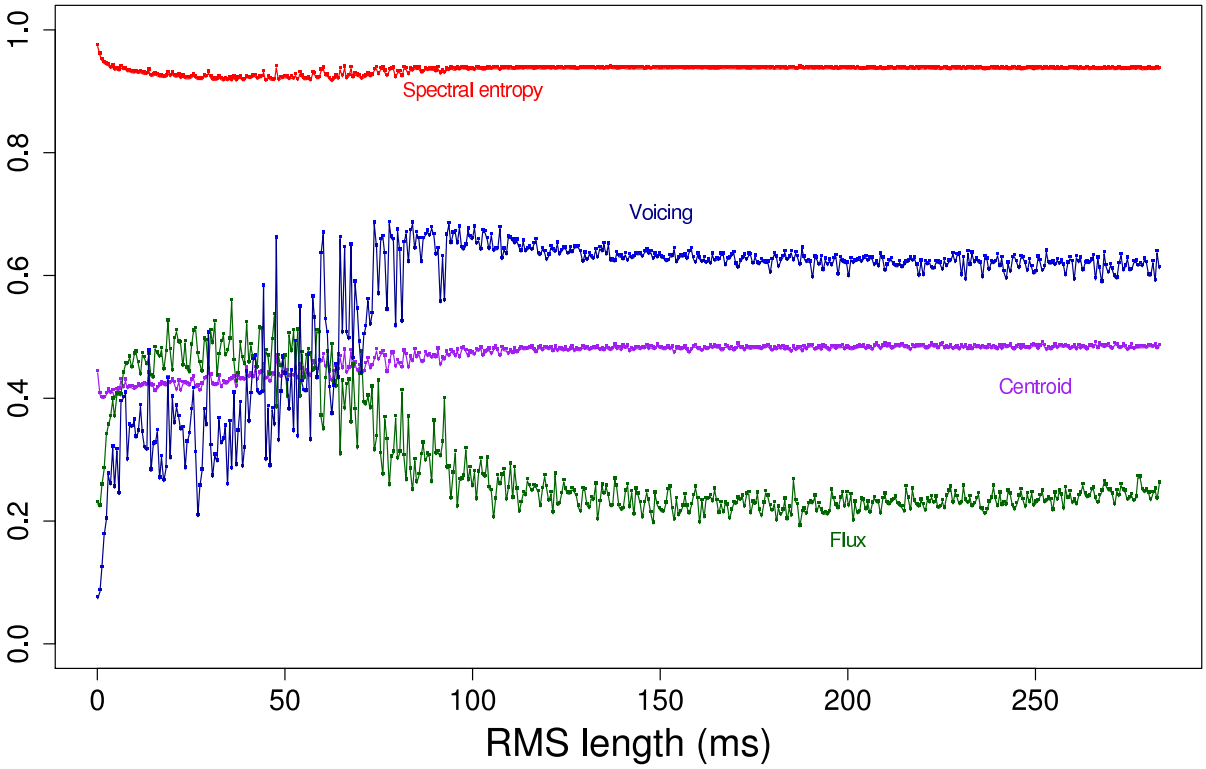

Figure 6.8: Cross-coupled map with instantaneous frequency and RMS amplitude extractors. Spectral attributes are shown as a function of RMS length.

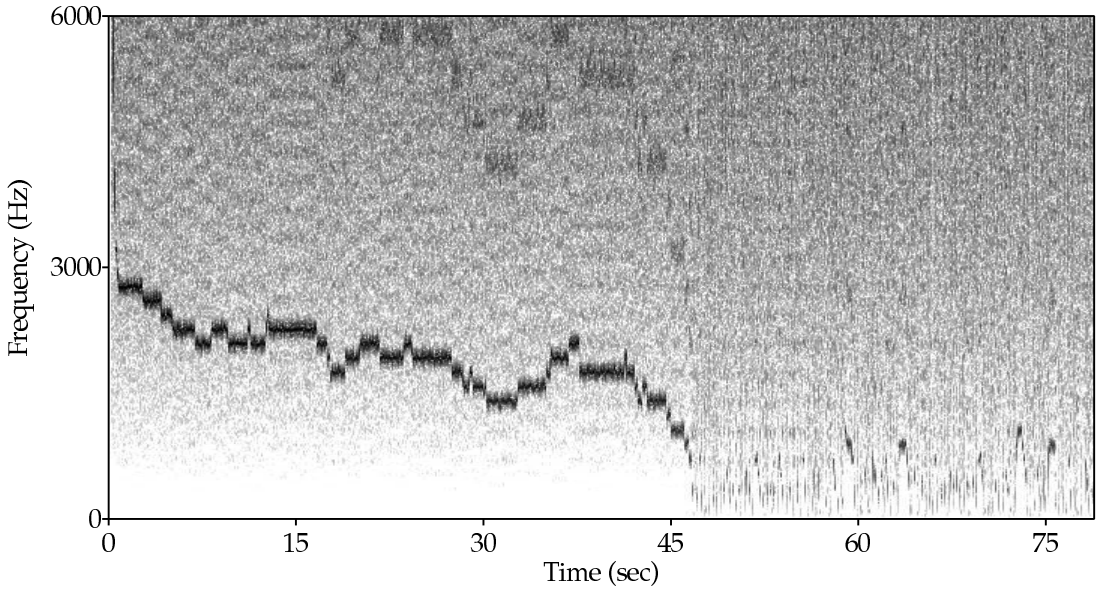

Figure 6.9: Spectrogram of the cross-coupled map with a 150 ms RMS window. Notice the meandering pitch in the first part and the sudden change after 45 seconds.

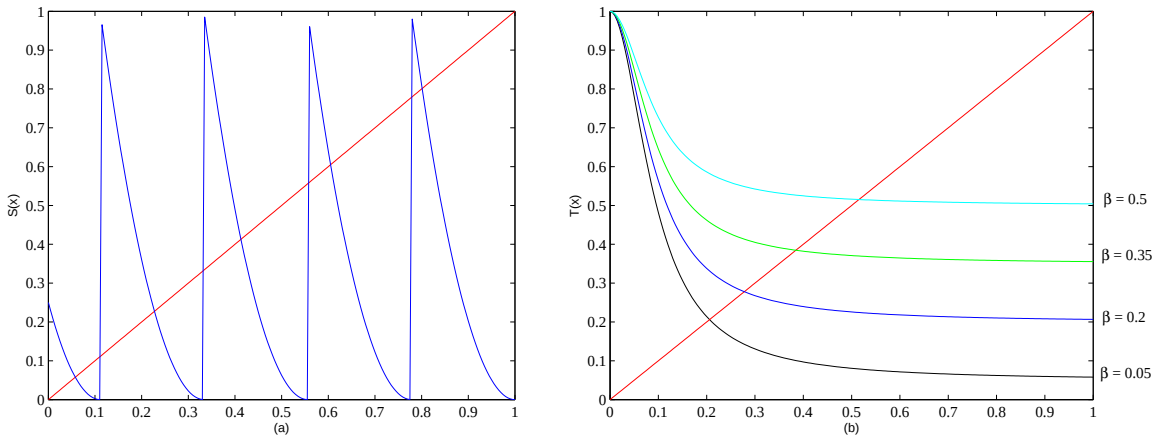

Figure 6.10: *a*) The map $S(x) = [\mu(1-x)(\text{mod}1)]^2$ with $\mu = 4.5$, and *b*) the map $T(x) = \frac{1-\beta}{1+\kappa x^2} + \beta$ with $\kappa = 120$ and $\beta = 0.05$, 0.2, 0.35, and 0.5.

The combination of RMS amplitude and instantaneous frequency estimators give the most promising results, especially for RMS windows somewhere in the range 100 – 200 ms. However, these sounds are infected with noise, and the question is whether one can get rid of it without losing the interesting melodic profile. A possible solution would be to apply a moving average filter to the frequency variable, although this alters the dynamics in various ways depending on the two time constants for the RMS and filter lengths. Short transients leading to steady high frequency oscillations are easily obtained. As general observation, the introduction of a moving average filter needs to be compensated by reducing the RMS length, but even so, the dynamics is fundamentally different. The only foolproof way to reduce the noise without affecting the dynamics would be to filter it outside of the loop, i.e., to resort to post processing. The drawback is the same as for aliasing: once the disturbance has entered the signal it is too late to remove it.

6.2.4 Bifurcations of the ST-map

We have by no means run out of variations of the cross-coupled map yet. The maps (6.18 and 6.19) have three free parameters that can be tuned to find interesting behaviour. Above, an extra moving average filter was inserted to smooth the estimated frequency and amplitude variables, although there is an even more obvious place to insert it, namely after the output signal, but inside the feedback loop. Before trying that, let us use the RMS and ZCR features and vary the parameters of the maps.

The parameter $\mu \in \mathbb{R}^+$ in the S-map (6.18) determines the number of repetitions of the function's basic shape within the interval $[0, 1]$. Here, the crucial thing to note is that each continuous segment of this function maps to the interval $[0, 1)$ or part of it; furthermore, there are $\lceil \mu \rceil$ (the smallest integer not less than μ) preimages of the map. A symbolic dynamics for a one-dimensional map requires the use of as many symbols as there are preimages (Hao and Zheng, 1998). Graphically, it is immediately obvious that the map gets more folded as μ increases, and thus more nonlinear in a quite precise sense

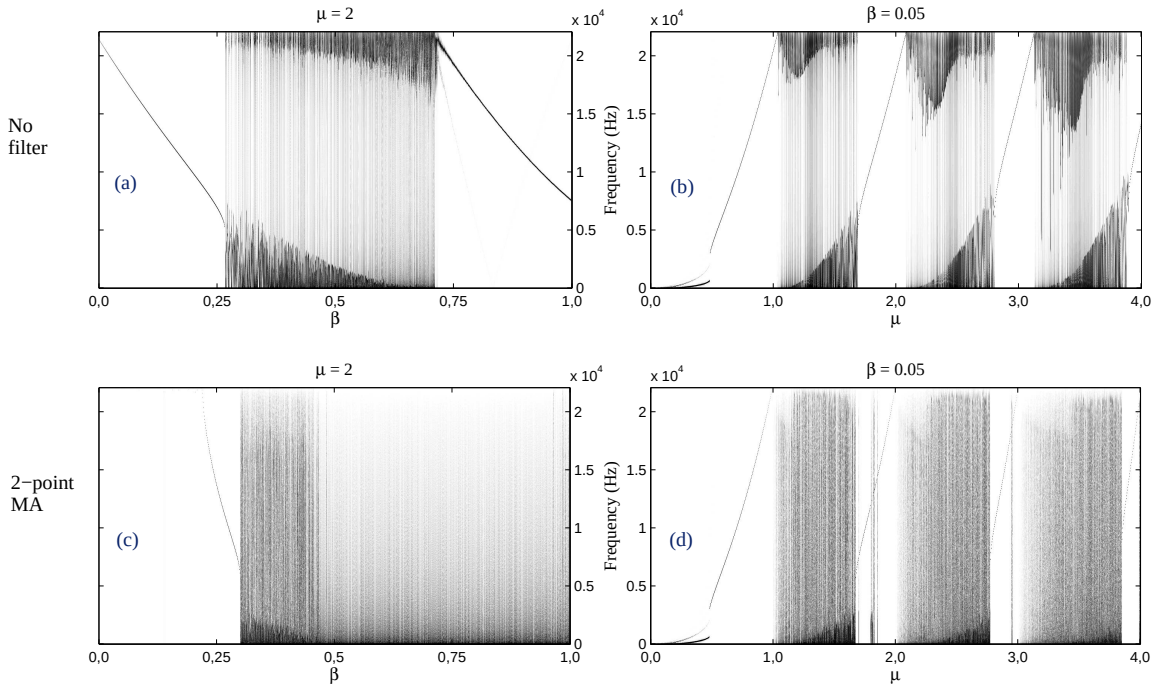

Figure 6.11: Spectral bifurcation plots for the cross-coupled map with 25 ms RMS and ZCR windows. The amplitude spectra are taken from about 2.8 seconds into the sound, using a 2048 point FFT. The upper row is without filtering, and the lower with a two point moving average after the oscillator. In the left column, $\mu = 2$ is fixed and $\beta \in [0, 1]$ is the bifurcation parameter. To the right, $\beta = 0.05$ and $\mu \in [0, 4]$. In all cases $\kappa = 120$ was used.

(compare figures 6.5 and 6.10*a*).

The parameter $\beta \in [0, 1)$ influences the range of the T-map (6.19), and moves its one and only fixed point lower or higher as β decreases or increases (see Figure 6.10*b*). The steepness is controlled by κ. Also, using interval notation $I = [0, 1]$, we have that $T(I) = [\alpha, 1]$ with

$$\alpha = \frac{1 + \kappa\beta}{1 + \kappa}.$$

In the limit $\kappa \to \infty$ (which is a useful approximation for the large value we have been using), this simplifies to $T(I) = (\beta, 1]$. We still keep $\kappa = 120$ fixed and vary each of the other parameters in turn. This is shown in Figure 6.11, both with and without a two point moving average filter, which is inserted after the oscillator. Then the oscillator's output x is given by

$$\begin{aligned} u_n &= a_n \mathrm{osc}(q_n) \\ x_n &= 1/2(u_n + u_{n-1}) \end{aligned} \tag{6.23}$$

when the filter is applied, where a and q are still given by the same system as in Section 6.2.1.

There is a clear difference between the spectral bifurcation plots depending on whether the filter is inserted or not. A slight lowpass filtering effect is to be expected, and is visible too, but there are more striking differences in how the boundary between a pure sinusoid and a noisier spectrum are altered, as well as in the generally more noisy appearance of the filtered versions. The sudden transitions from a fixed point or periodic orbit to chaos, which is probably what is seen here, is typical of the *crisis* phenomenon (Grebogi et al., 1983).

Spectral bifurcation plots of course cannot provide any indication of the sound's temporal evolution at each parameter value. What appears like a pure tone of decreasing or increasing frequency as β is varied actually only represents the stable behaviour after a more or less long initial transient, in the same manner as discussed above and as shown in Figure 6.7.

6.3 The extended standard map

The Chirikov standard map was introduced in Section 4.4.2. Here we will consider another model that is built upon the standard map, extending it in several ways by the use of filters and coupling between two or more instances of a basic model. Recall that the standard map is given by

$$\begin{aligned} v_{n+1} &= v_n + K \sin \theta_n \\ \theta_{n+1} &= \theta_n + v_{n+1} \end{aligned} \tag{6.24}$$

and both the variables are taken modulo 2π. The parameter K is a coupling constant which we will keep, although the coupling function will take another form. Suitable output signals for sound synthesis are either $\sin \theta_n$ or $\sin v_n$. The model we will end up with is quite complex on its own, even without inserting feature extractors and a mapping to any synthesis parameters. Therefore it has to be investigated closely before even trying to make a feature-feedback system of it. The use of moving average filters in the new model already makes it similar to feature-feedback systems, even before any actual feature extractors have been inserted.

6.3.1 Filters and coupling

A natural place to insert a filter in the map (6.24) is after the signal $\sin \theta_n$. From the variety of conceivable filters we choose to use a biquad bandpass filter B. Written as a difference equation,

$$B_{f_c} * x_n \equiv y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} - b_1 y_{n-1} - b_2 y_{n-2}, \tag{6.25}$$

it can be seen that the filter introduces four internal state variables, thus increasing the dimension of the entire system by four. It will be useful to control the filter's centre frequency (f_c). A moving average filter will also be used in the same place, and since these filters are linear, their order does not matter.

The standard map is an interesting model since, by itself, it is an oscillator, one variable representing the position along a circle and the other variable representing the

velocity of this point. Unfortunately, it is quite limited sound-wise, while being difficult to control under parameter changes. At the same time it is very amenable to modifications, although it is hard to find any that leads to dramatic improvements. If the following model appears like a rabbit drawn out of a hat, it is because it has been arrived at after testing a few similar, but less interesting systems which will not be discussed here. Although many other useful models may also be derived from the standard map, suffice it to remind of the utility of filtered maps as synthesis models, as discussed in Section 4.3.

Essentially, we keep the two variables of the standard map, but the coupling between them will be different. As in the original standard map, the variables x and v are here taken modulo 2π. The standard map is defined on a topological 2-torus; the extended system consists of N coupled maps which live on a $2N$-torus.

Apart from the coupling between two or more maps, there is the bandpass filter B (6.25) and a moving average filter, which also increase the order of the system. Using circular coupling between N oscillators, we notate the neighbouring oscillator with index $i \oplus 1 = i + 1 (\mathrm{mod}\, N)$. Then, the complete system is:

$$\bar{z}_n = B_{f_c} * \frac{1}{M} \sum_{m=0}^{M-1} z_{n-m} \tag{6.26}$$

$$v^i_{n+1} = v^i_n - K(\sin(x^i_n - x^{i\oplus 1}_n) + \bar{z}_n) \quad (\mathrm{mod}\, 2\pi) \tag{6.27}$$

$$x^i_{n+1} = x^i_n + \frac{1}{2}(v^i_{n+1} + v^{i\oplus 1}_{n+1}) \quad (\mathrm{mod}\, 2\pi) \tag{6.28}$$

$$z_n = \frac{1}{N} \sum_{i=0}^{N-1} \sin v^i_n \tag{6.29}$$

Here, the indexing convention is used that superscripts indicate different variables, and subscripts are used for the time index. In practice, two coupled maps ($N = 2$) produce the most malleable sounds. Increasing the number of coupled maps tends to give noisy and homogenous sounds. For audio output, an alternative is to use the mixture z_n of all oscillators. When multi-channel output is feasible, each oscillator $y^i_n = \sin(v^i_n)$ may be sent to its own channel.

The variables x and v are of special importance, while the z variables can be regarded as auxiliary. Strictly speaking, the extended standard map written in state space form should expand all the delayed variables in the filters, but doing so would be inconvenient.

Coupling between the oscillators happens in both x and v. In the general case of $N > 2$ oscillators, the coupling is circular in two out of three places. Notice the circular coupling term $\sin(x^i - x^{i\oplus 1})$ in (6.27) which is of a similar functional form as the coupling in the Kuramoto model (see Section 4.5.4). In (6.28) the average of two adjacent oscillators (on the circular topology) makes up the coupling; and lastly in (6.29) all oscillators are mixed together, so the variable $\bar{z}$ is both a spatial and a time average.

Notice that for $N = 2$ the system is symmetric in v^i and x^i, $i = 0, 1$, in the sense that the equations for v^0 and v^1 are interchangeable, and likewise for x^i. For $N > 2$ the symmetry breaks down because of the circular coupling—with all-to-all coupling the symmetry would have been retained. From now on we consider only the case $N = 2$.

Starting from initial conditions $x_0^0 = x_0^1$ and all other variables set to zero, the system will be stuck on a fixed point because in (6.27), v_{n+1}^i will evaluate to v_n^i, so there is nothing to set the system in motion.

Now we look at the parameters. From top to bottom in the complete system, there is $M > 1$, the moving average filter order; f_c $[Hz]$ the bandpass filter's centre frequency (for simplicity the Q factor is fixed); K is the coupling strength (typically $|K| < 1$ but larger magnitudes can be used). Of these parameters, the filter order M has to be kept fixed for each run of this system, leaving the filter centre frequency f_c and the coupling variable K to be controlled. In fact the filter order M has a strong influence, so it will have to be considered too.

Amplitude death—or rather oscillation death, which means that the system stops oscillating altogether—necessarily happens for $K = 0$, because then there is nothing to increment the variable v in (6.27). Since it is inefficient use of a synthesis model to calculate silence with such a complicated algorithm, it may be a good idea to avoid running the system with $K = 0$. Fortunately this condition is not permanent, so as soon as K becomes non-zero, oscillations can start over again.

Other peculiarities as parameters are continuously varied include pitch transitions seemingly along an overtone series and sudden timbral transitions. The signal z_n may contain a substantial DC offset, especially in periods of oscillation death. A DC blocking filter can be practical as a post processing unit. Inserting the DC blocker inside the system's loop (in eq. 6.26) does not appear to alter its dynamics radically. In the following, no DC blocker is used inside the loop, so the system is as appears in eqs. 6.26–6.29.

6.3.2 Investigation of parameters

Now the entire extended standard map can be encapsulated and used as the signal generator in a feature-feedback system with stereo output, which we write as $y_n = \mathcal{X}_M(f_c, K)$, where $y_n^i = \sin(v_n^i)$ for $i = 0, 1$. Before attempting such encapsulations, however, it will pay off to study the system on its own since it is already quite complicated.

First, we need to study the behaviour of the extended standard map under parameter variations with the help of some feature extractors. Then we will return to the question of how to devise an autonomous control scheme for it which will turn it into a feature-feedback system.

The sound is often pitched, and as K increases, the pitch rises. The exact relation is complicated though; there are frequency locking regions as well as chaotic, unpitched regions. Sudden timbral contrasts are to be expected; there is scarce hope to smooth out these transitions, which must rather be accepted as a distinguishing trait of this system.

Example 6.3. For small values of $|K|$ the sound is pitched and more or less stable, but initial transients in the form of peculiarly shaped glissandi can be heard, for instance at parameters $f_c = 50$ Hz, $M = 4$, and $K = -0.1$.

Example 6.4. Intermittency may occur in the form of rapid and noisy switching between two pitches, such as happens at $f_c = 500$ Hz, $M = 4$, and $K = 0.2$.

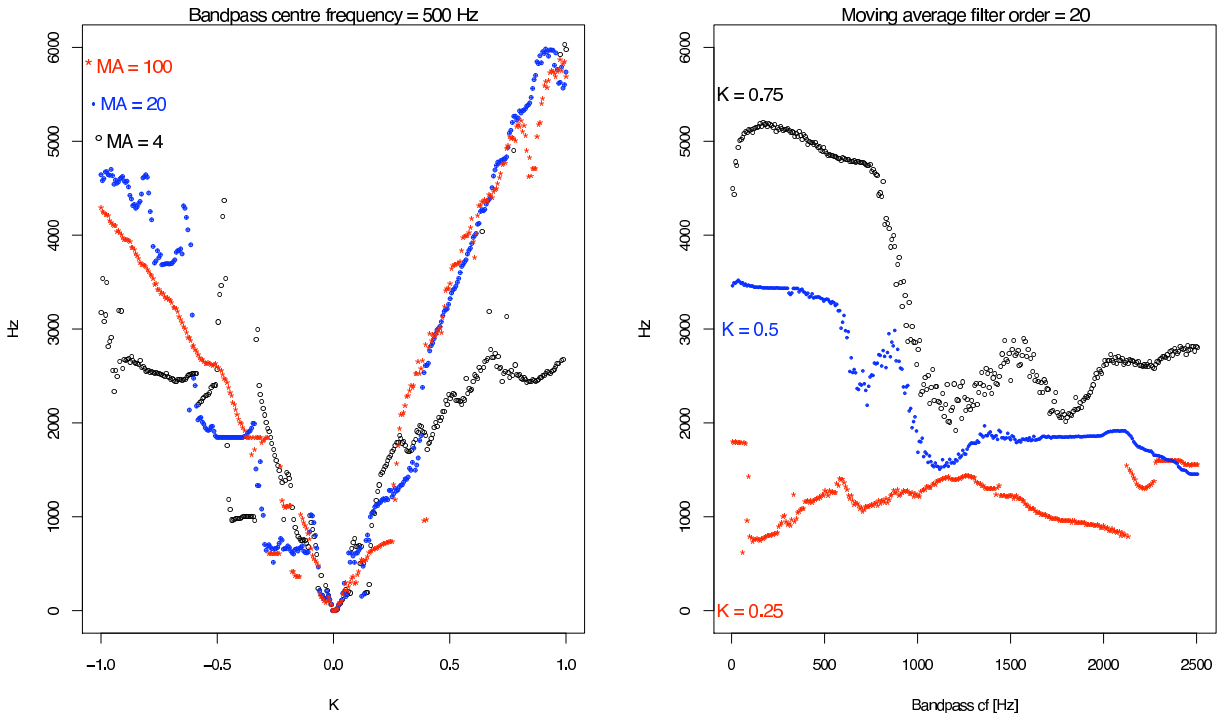

Figure 6.12: (Left) Resulting frequencies as a function of coupling K. Different lengths of moving average filters are shown: black, 4th order; blue, 20th order; red, 100th order. (Right) Measured frequencies as a function of bandpass centre frequency. Note that increasing K at any bandpass frequency tends to raise the observed frequency. Frequencies have been measured with the autocorrelation method over a 2048 point window, and taking an average over 100 consecutive windows.

In general, the shape of the obtained frequency as a function of K is similar for a range of choices of M and f_c, although the details and slope may vary. Some examples are shown in Figure 6.12. Note the jumps in frequency as K is varied (left plot): a structure similar to the "devil's staircase" can be seen for negative K and moving average filters of low order. As the bandpass filter frequency is varied, more complicated curves result for the observed frequency, but for the bandpass centre frequency range up to 2.2 kHz, it can be seen that increasing the coupling strength (for $K > 0$) has the effect of increasing the observed frequency.

The Lyapunov exponents are qualitatively similar to the measured frequency as a function of K. Figure 6.13 shows the greatest Lyapunov exponent for different directions of initial displacement. For the blue curve, the initial displacement has been taken along the x^0 coordinate, the black curve shows that of x^1, and to avoid clutter, an average of the two curves for displacements along v^0, v^1 is plotted (in red). For most values of K, the two v-dependent curves are close together. What is more remarkable is the portions of the figure where the different displacement vectors yield completely different Lyapunov exponents. The method of Lyapunov exponent estimation is the one described above (Section 6.1.8), but as said before, this method has not yet been validated by comparison with known systems. Hence, the exact values should be regarded with a certain caution,

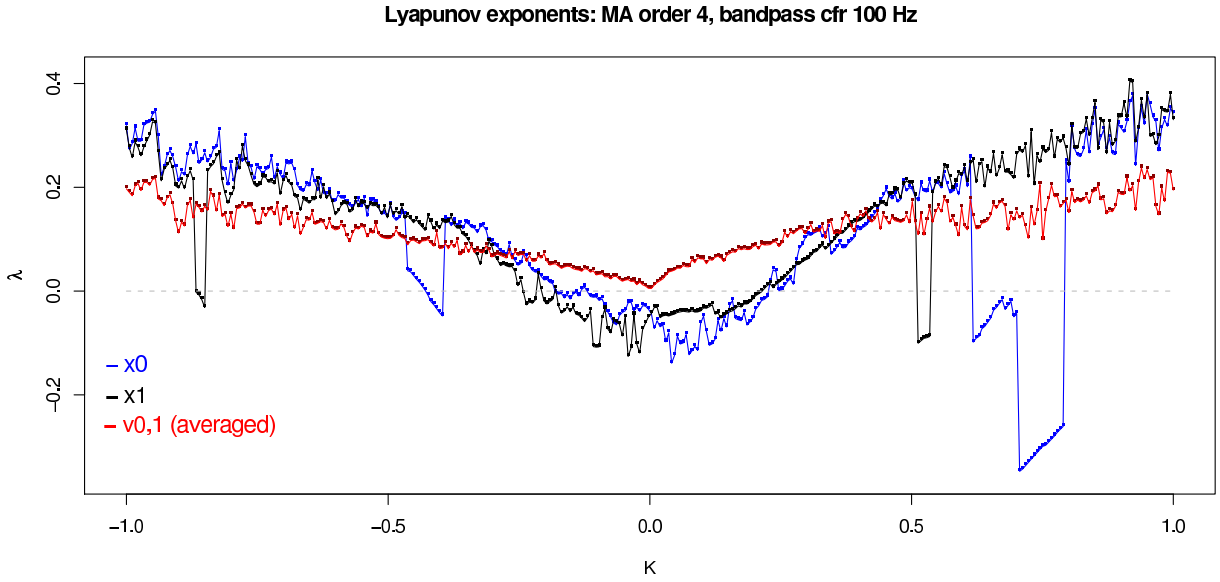

Figure 6.13: Lyapunov exponents of the extended standard map with a fourth order moving average filter and bandpass centre frequency set to 100 Hz, as a function of coupling. The three curves correspond to different vectors of initial displacement; along x^0 in blue, x^1 in black, and $v^{0,1}$ in red being the average for displacements along both v-directions.

although there is no reason to doubt that the system is chaotic over a large range of its parameters.

The state space dimension for the system is the number of variables needed to express (6.26–6.29) explicitly in state space form. Then we need M variables for the moving average filter, another four variables for the bandpass filter (6.25), four x and v variables in total for the case of two coupled maps (the z variables are just there for notational convenience), yielding a total of $8+M$ nominal dimensions. If one would take the trouble of writing out the Jacobian matrix of this system, it would be a sparse matrix with many rows having a single entry equal to 1 and the rest full of zeros, corresponding to time delay operations in the filters. Hence, the system can be of high dimension, in particular if it uses long moving average filters. The point to note here is that this is similar to feature-feedback systems with much simpler signal generators, where the delay lines of the feature extractors contribute a comparable number of extra dimensions.

6.3.3 Phase synchronisation

Given that there are two sets of variable pairs which play identical roles (x^i and v^i), one may study their synchronisation. The synchronisation between v^0 *and* v^1 is particularly interesting to look at, since they are used for the output signal. If they are synchronised in a general sense, the two output channels will be in a constant phase relation. Strict synchronisation means that two or more variables take on identical values, but other generalised forms of synchronisation may occur. Here, a sufficient synchronisation criterion is that the phase offset between the two variables is constant. Lack of synchronisation is equivalent to *decorrelation*. For a stereo image, the decorrelation between the channels is

perceptually important. High decorrelation results in wide stereo images, with the sound source's apparent position outside the head when listening through head phones, whereas for high positive correlation, the apparent location is inside the head (Kendall, 1995).

Since the variables $x, v \in [0, 2\pi)$ can be interpreted as angles, circular statistics must be used (Beran, 2004, ch. 7). Let the complex variable $\rho = e^{i\theta}$ be the complex representation of the angle $\theta \in [0, 2\pi)$. In this case we are interested in measures of angular differences rather than average directions. Let us define the distance of two angular variables u, v as

$$d(u, v) = \frac{1}{2} \left| e^{iu} - e^{iv} \right| \tag{6.30}$$

so that the distance $d \in [0, 1]$ is normalised. After some simplification, this reduces to

$$\frac{1}{\sqrt{2}} \sqrt{1 - \cos(u - v)}$$

from which it is convenient to calculate an RMS measure of angular distances over a window of N samples (after further simplification) as

$$D(u, v) = \left\langle d^2 \right\rangle^{1/2} = \sqrt{\frac{N-1}{2N} \sum_{n=1}^{N} \cos(u_n - v_n)}. \tag{6.31}$$

Average angular differences are not so informative on their own, however, since the same value of $D(u, v)$ may result if all the differences $u_n - v_n$ are equal, or if the angular differences vary more over time. For a perceptually motivated statistic some measure of angular spread or concentration is preferable. A simple measure of angular concentration is the mean resultant length $\bar{R}$ (Beran, 2004), which is easily stated in the complex exponential representation as

$$\bar{R} = \frac{1}{N} \left| \sum_{n=1}^{N} e^{i\theta_n} \right|, \tag{6.32}$$

which takes values close to zero if the angles are spread out evenly or in opposing directions, and values close to one if all angles are focused in one direction. Circular variance $V = 1 - \bar{R}$ is more convenient to use since we will look at the angular spread. Again, it is the phase differences that matter, but this is easily handled by setting $\theta = u - v$ in (6.32). Hence, we will use

$$V = 1 - \frac{1}{N} \left| \sum_{n=1}^{N} e^{i(u_n - v_n)} \right| \tag{6.33}$$

as the measure of average circular variance of phase differences.

The average circular variance V (eq 6.33) as a function of K is shown in Figure 6.14 (a). For small positive K the variance is low, indicating that both v variables are highly synchronised. A functional dependence is revealed in Figure 6.14 (b) by plotting the variance V against average phase difference D (6.31). The same curve is more or less filled in by choosing different bandpass centre frequencies. Fluctuations in the phase

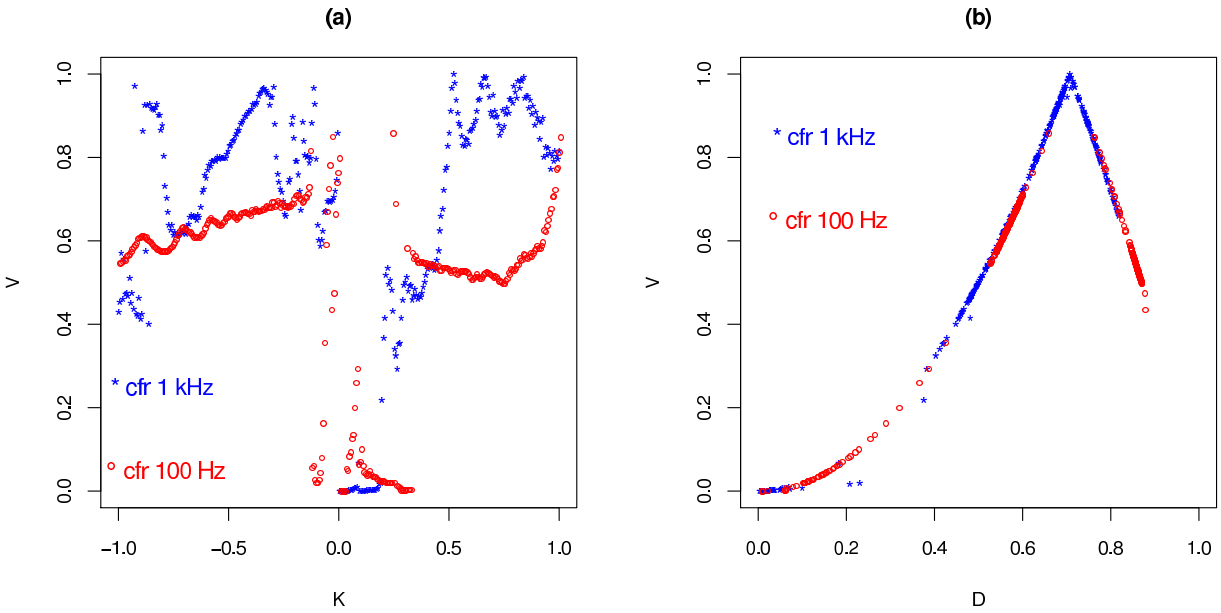

Figure 6.14: Phase differences in the v variable of the extended standard map with moving average filter order 20. (a) Average circular variance of phase difference as a function of coupling. (b) Variance V for the same data as a function of mean angular distance D. Centre frequencies of the bandpass filter are 100 Hz (in red) and 1 kHz (blue). Notice the functional dependence of V on D which holds for both bandpass frequencies. The data are collected after letting the system run for 1 second (48000 iterations) by averaging over 9.8×10^7 iterations of the map.

difference between the stereo channels are easily heard, and the interpretation of the left part of the figure is that these fluctuations do not occur for small positive K, thus leading to a stable stereo image.

In Chapter 4, the order parameter $re^{i\psi} = \frac{1}{N}\sum_{n=1}^{N} e^{i\theta_n}$ was introduced, where in fact r is equivalent to $\bar{R}$ in (6.32). The mean direction ψ might also be studied, but that is of little use here.

6.3.4 Finding feature extractors

The extended standard map, though complicated, produces quite static sounds unless its parameters vary over time. Therefore it will be used as the signal generator in a feature-feedback system. It must be stressed that the extended standard map is quite complex by itself, with a high number of state space dimensions mainly given by the length of the moving average filter.

Now, the question is what feature extractor to use. An optimal choice of feature extractor should be one that captures as much of the output signal's variety as possible as the signal generator's control parameters are varied. For instance, suppose that the spectral centroid remains virtually constant even though other features change, then it would presumably be wasted effort to use a centroid feature extractor. However, the numerical range of variation in the feature is not the decisive factor here, because even a

small variation that consistently follow parameter changes may be more useful than large variations that seemingly follow no predictable pattern. Moreover, the perceptual salience of the changes captured by the feature extractors determine how intuitive they will be to use. Listening to a few typical sounds (and looking at waveforms and spectra) can give clues as to which feature extractors might be worthwhile trying. A more ambitious approach is to do an automated search through the parameter space and monitor minima, maxima, and perhaps other statistics of a large set of feature extractors. In Figure 6.8, a limited demonstration of that approach was shown, in which only one parameter was varied. Exhaustive automated searches of this kind can be very computationally costly; this is a situation where evolutionary algorithms or other efficient search techniques could prove to be useful.

With only three parameters, the extended standard map is simple enough to search exhaustively. Still, a search using a fine-masked resolution of the parameter space would be computationally demanding. Figure 6.15 demonstrates what can be found after a quite coarse-grained search in the three parameters K, f_c, M (although the moving average filter order M is not shown). For each feature extractor, its values are sorted and the ten highest are plotted as red '+' signs and the ten lowest as blue circles. Note that the points have different filter orders M, but for such attributes as RMS amplitude, spectral entropy and voicing the separation in the K, f_c plane is nevertheless good; that is, regardless of filter order M, the features occupy different regions of the plane. What appears to be closely spaced points in the K, f_c parameter plane may thus be safely separated if those points have different filter order.

Let us summarise a few traits in qualitative terms. We use fuzzy terms such as "high" and "low" parameter values, understanding them to come from the highest (lowest) half of the investigated parameter ranges, which are $K \in [-1.25, 1.25]$ covered in 75 steps, $M = \{4, 9, 20, 45, 100, 225, 500, 1125, 2500\}$ and $f_c \in [10, 2500]$ Hz covered in 12 equidistant steps on a logarithmic frequency axis.

The estimated frequency is typically low for K near zero and for long filter orders M, whereas for K large and negative and for short filter orders, the frequency tends to be high. Spectral entropy attains as low values as about 0.3 for short M, low f_c and small positive K. For large negative K, high f_c, and short M, the spectral entropy reaches up to about 0.96 at its highest. It can be seen from the figure that RMS amplitude is neatly divided along the f_c axis with high amplitudes for low filter frequencies and lower amplitudes for high f_c, for K around zero. Actually a zero amplitude (indicating that the system stopped oscillating at some point) is measured for small negative K and long M. The strongest amplitudes are also measured for long M.

Panel (c) of Figure 6.15 shows the time averaged stereo channel difference, $d = \langle |x - y| \rangle$ where x is left and y is right channel. Inspection of the waveform shows that often the phase between left and right channels is either in sync, or else it is 180° out of phase. Intermediate values are observed because of time averaging over periods when the phase may switch between these polarities. From the figure it might look as if the phase difference would change wildly when K is near zero. On closer inspection the channels are in phase for positive K and out of phase for negative K. Such phase changes are clearly audible if they occur during a sound.

There are problematic aspects of the proposed search method that must be kept in

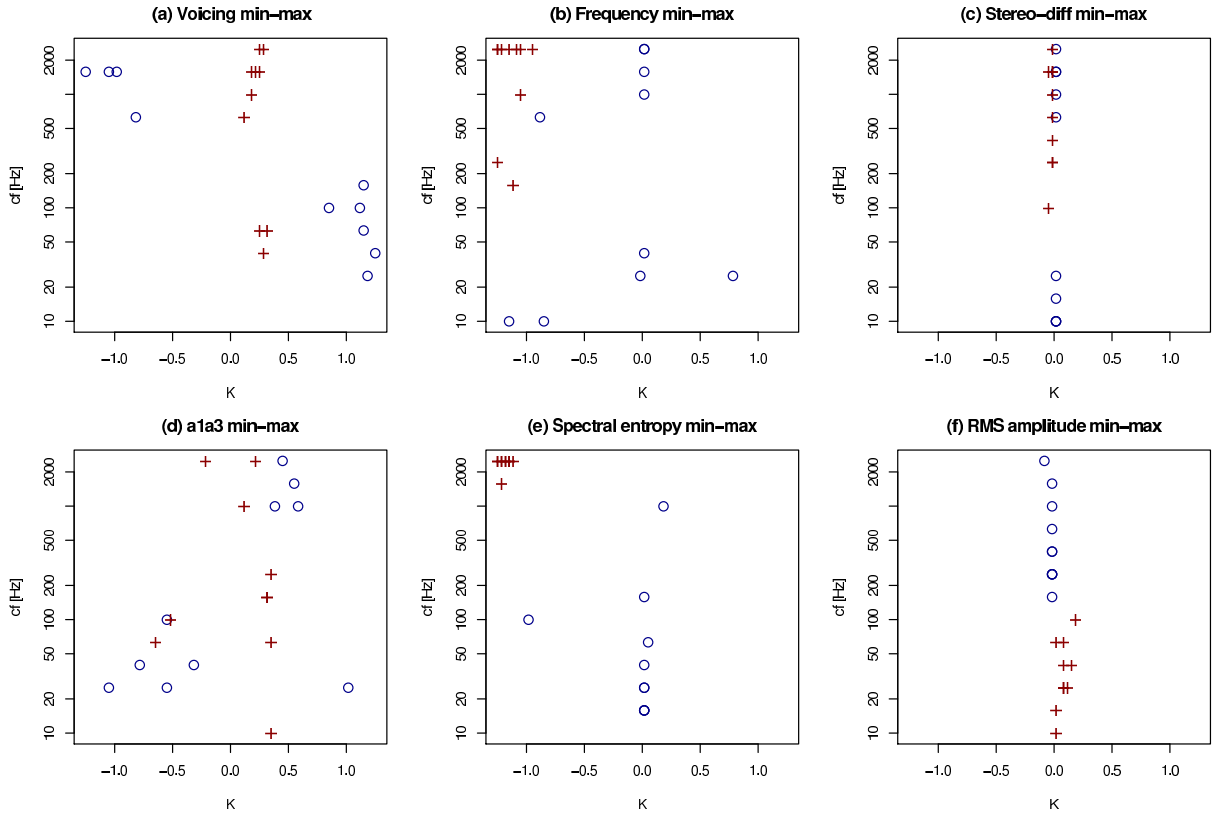

Figure 6.15: Parameter locations in the K, f_c plane of the extended standard map where the minimum and maximum values of feature extractors are found. A total of 8892 parameter points were searched. Blue circles are the ten lowest values, red plus signs are the ten highest values. All measurements have been done 1 second into the sound ($f_s = 48$ kHz), with a 2048 point FFT window in the case of spectral extractors. Diagrams (a) and (b) are the voicing and fundamental frequency estimated by autocorrelation; (c) stereo difference is an time average of absolute difference between left and right channels; (d) is the a1a3 measure obtained with time domain filters; (e) is the spectral entropy; (f) RMS over a 200 ms window. Not shown in this figure is the third control parameter M, the length of the moving average filter, which ranges from 4 to 2500 points in nine steps. For voicing, spectral entropy and RMS amplitude, the separation is clear even without taking M into account.

mind. First, the synthesis model has been shown to be chaotic in large portions of its parameter space, and transients are common. The sounds may be non-stationary in pitch and other attributes, whence the particular time point picked out for analysis (in this case one second into the sound) may not be representative. Taking time averages of all the searched feature extractors is a conceivable solution, although in combination with the exhaustive search through a large set of parameter locations the computational cost would become very high. Another problem is that the extreme values of the features may be outliers, and hence not likely to be obtained under usual conditions. Furthermore, the filter order M must be fixed when running the instrument, so the search could have been simplified, had the filter order been set first.

The next stage is the design of a mapping function that translates the data from a feature extractor to synthesis parameters. At this point it can be instructive to look at the complete statistical distribution of the feature, and not only its extreme values.

6.3.5 Nesting with a1a3

Now we fix $M = 100$ and introduce a mapping from a feature extractor to K and f_c. We will use the sliding version of the a1a3 feature, so that it outputs its values at the audio sampling rate (see Section 2.3.7). Its output is in dB, and from the investigation of extreme values above, it was found that the extended standard map is capable of producing almost the theoretical minimum and maximum of a1a3. The total range of a1a3 is $[-180, +180]$ dB, which follows from its implementation

$$a1a3 = 20 \log_{10} \left(\frac{a_1 + \epsilon}{a_3 + \epsilon} \right)$$

where a_i are the amplitudes of the two formants to be compared, and $\epsilon = 10^{-9}$ is used to avoid zero division. Across all investigated parameters, the observed range of a1a3 was $[-167.8, +172.5]$ dB. In recorded acoustic sounds the range is typically much narrower, such as perhaps -40 to $+40$ dB; furthermore, the lower formant is often the strongest, favouring the positive part of the range.

Figure 6.16 shows the dynamics as measured with a1a3 four seconds into the sound on the K, f_c plane. There are white regions for K close to zero and $f_c > 300$ Hz, as well as for $K \approx 0.35$ and $f_c \approx 1$ kHz; these may indicate oscillation death, unless the system is oscillating and the two formants are delicately balanced in amplitude (actually the interval plotted as white is $-0.5 < a1a3 < 0.5$ dB). The second alternative is unlikely; from an inspection of the a1a3 data it was found that a1a3 is exactly zero for most points of the white regions.

Making this kind of plots is very time-consuming, even with present computer power. If one were to generate a sound file corresponding to the total length of the signal needed to calculate a 320×475 pixels plot such as this figure, its duration would be well over seven days. There is much computation to be saved by shortening the initial part of the signal which is generated in order for transients to die out, but then one will know even less about what state the system eventually stabilises on (if it ever stabilises). It must be stressed that other a1a3 values could have been measured if a shorter or longer transient had been skipped.

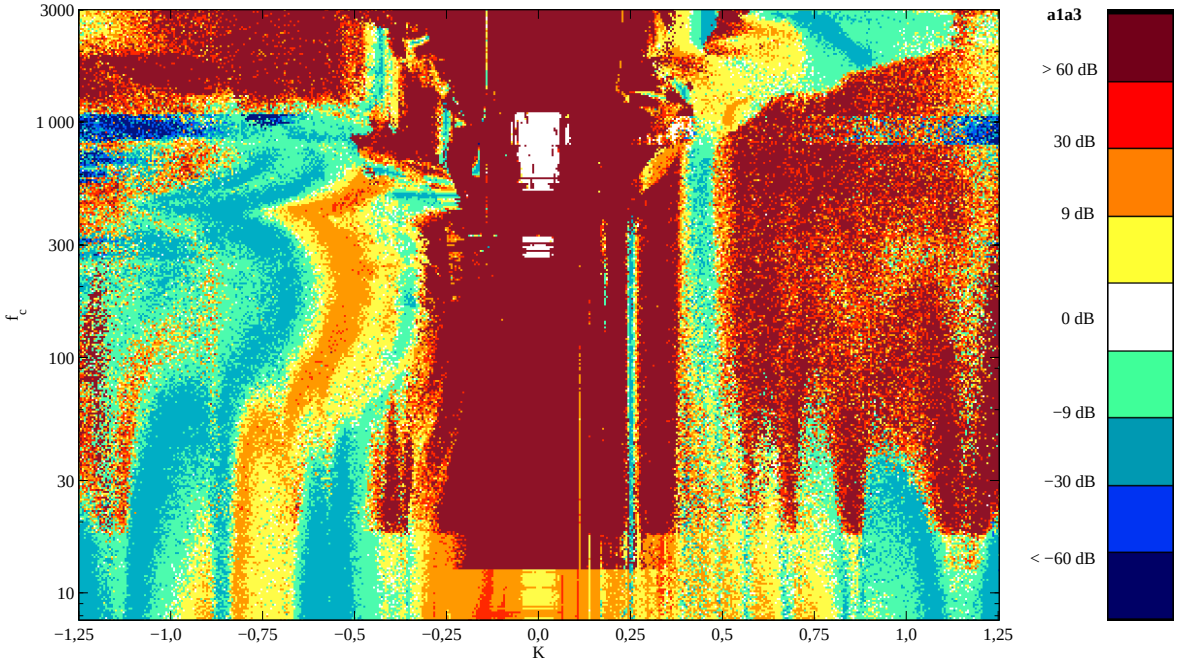

Figure 6.16: Extended standard map with $M = 100$ over the range $K \in [-1.25, 1.25]$ and $f_c \in [10, 3000]$ Hz. The feature extractor a1a3 is plotted in hot colours for positive values (where the lowest formant is the stronger one) and cold colours for negative values. Each parameter point has been measured after a transient of four seconds, and represents a time average over 0.34 sec.

For the mapping function, we know the domain of its input argument (a1a3). Since it will be used to map to the K and f_c parameters, we can impose appropriate bounds on its range. At this stage it is still possible to design the mapping so as to avoid the white regions in Figure 6.16 that are suspected of causing oscillation death. Here it is convenient to introduce a variable substitution $\alpha_n = a1a3(y_n)/180$ so that $\alpha_n \in [-1, 1]$. Since the signal $y_n = \mathcal{X}_M(f_c, K)$ is actually in stereo, the input to the feature extractor will use a sum of the channels. Reversing the phase of one channel before mixing them may result in anything from subtle to striking differences.

For the mapping, we can begin by restricting the range of the function to a rectangular box $I \times J$ on the K, f_c plane given by the intervals I and J. Thus the map should be of the form $\gamma : [-1, 1] \to I \times J$. Let us try the map

$$\begin{aligned} K_n &= A \sin(\pi B \alpha_n) + C \\ f_{c,n} &= D(1 - |\alpha_n| + E) \end{aligned} \tag{6.34}$$

with five tunable parameters $A, \ldots, E \in \mathbb{R}$. In order to avoid continuing fast changes, B should not be of very large magnitude. With $B = 1$, a1a3 could potentially map to a full period of the sine function. The parameter A is the amplitude determining the total range of K, and C is a bias. Then, if $B \geq 1$ the range of K will be $C - |A| \leq K_n \leq C + |A|$,

and for f_c the range will be $E \leq f_c \leq E + D$.

One can imagine a graphical user interface where the current location in the K, f_c plane could be chosen arbitrarily by the user by pointing at an arbitrary location on a rectangular plane. In contrast to that, maps such as (6.34) only make a small subset of the plane reachable, namely the set of points that lie on the curve γ. Consequently, the map imposes a constraint on possible parameter *combinations*: any $K \in I$ is accessible, but then the choice of $f_c \in J$ is restricted, and vice versa. Hence, this restriction introduces some structure on the possible dynamics.

Without going too deep into abstruse mathematical matters, this situation is related to the fact that curves in the plane normally do not have area. Certainly there are exceptions, such as Peano's and Hilbert's area-filling curves, and a less known construction by W. Osgood (Hanche-Olsen, 2005; Lakoff and Núñez, 2000). Nevertheless, in practice we will have to be content with curves that have positive finite length, but no area.

In fact, it can prove useful to calculate the relative arc length of a map as a function of its parameters. Suppose the mapping $\gamma : I \subset \mathbb{R} \to E \subset \mathbb{R}^2$ is contained in a region E with finite area, and that there are at most a countable number of pairs of points $t_i, t_j \in I$ that both map to the same point in E; in other words, the curve does not overlap itself except for, perhaps, at some exceptional points. Then, if a curve is curled up and fills a large part of E, it will have a longer arc length than a curve that is less curled. This way of reasoning is of course not very strict since we have not said what it means to "fill up" a region, but it is only meant as a motivation for measuring the arc length.

The arc length of a path $\boldsymbol{c}(t), t \in [t_0, t_1]$ is $\int_{t_0}^{t_1} \|\boldsymbol{c}'(t)\|\, dt$, or, in the case of the proposed map (6.34),

$$\int_{-1}^{1} \sqrt{(\pi AB)^2 \cos^2(\pi B\alpha) + D^2}\, d\alpha. \tag{6.35}$$

As expected, C and E play no role in determining the arc length. The only problem with this formula is that it mixes two different units: D is a quantity in Hz whereas A and B are unitless, and typically much smaller in magnitude than D. As it happens, the effect of parameter changes in K is considerably stronger than that of f_c, in particular if compared in unitless numbers versus Hz; a change of 1 Hz might go unnoticed while a change in K from, say, 0.1 to 1.1 is enormous. This shortcoming notwithstanding, (6.35) indicates which parameters will stretch the available parameter locations most. The double occurrence of B can be taken as evidence that changing it slightly has a stronger effect than changing A as much. Ultimately, what effects are obtained depend on what points in the parameter space the curve maps to. Different shapes of the curve (6.34) are shown in Figure 6.17. Increasing B clearly makes the curve more curled up.

The filter centre frequency may be updated slower than at sample rate, and this control rate could even be flexible. Rapid modulation of filter coefficients is usually avoided, except when it is deliberately used as a synthesis model, such as in the audio rate modulation of allpass filter coefficients (Kleimola et al., 2009). Linear interpolation could be used if the sudden variable changes need to be smoothed out, but given the smaller effect of changing the filter parameter this is not an issue.

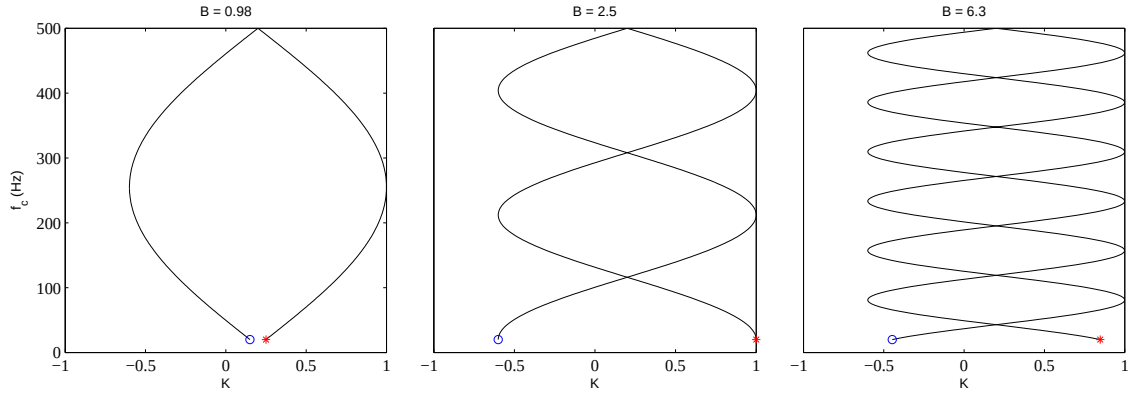

Figure 6.17: Mappings of a1a3 of increasing arc length; from left to right: $B = 0.98$; $B = 2.5$ and $B = 6.3$. The other parameters are: $A = 0.8, C = 0.2, D = 480, E = 20$. The point corresponding to $\alpha = -1$ has been marked with a blue circle, and the point at $\alpha = 1$ is marked with a red star. At integer values of B, the endpoints join.

Example 6.5. A behaviour similar to the intermittency in the previous example (6.4) is found with the mapping (6.34) at the parameters $A = B = 0.6$, $C = -0.3$, $D = 500$ and $E = 20$ Hz, $M = 100$, where a high pitch is regularly interrupted by shorter episodes on lower pitches. Subtle timbral changes follow the descents in pitch.

This model with the mapping (6.34) may be studied by trying out a few static parameter combinations and listening to the output, but it is more efficient to sweep slowly across ranges of one parameter at a time. In this synthesis model, there are lots of qualitatively different sounds to find, but its control parameter space is by no means smooth. Frequency locking, abrupt changes from tone to noise or from one pitch to another, as well as hysteresis effects must be counted on.

6.4 Brownian motion in frequency

Noise generated by pseudo-random number generators have many important applications in computer music, and particularly in sound synthesis. A synthesis model using noise to produce effects of shimmer and jitter was considered in Chapter 3. In this section, we introduce a nonlinear oscillator that uses noise to escape from oscillation death. Solutions to the problem of evading static behaviour in various guises will be a recurrent theme in the next chapter.

As soon as noise enters a dynamical system, it is no longer deterministic and the usual bifurcation theory cannot be used to describe the dynamics, or at least it has to be cautiously applied. Instead of a single realisation from one particular initial condition there will be a distribution of trajectories, each with its own probability of occurring. In the end of this section, a practical method of pitch control will be introduced. This method would also be suitable to adopt for the control of nonlinear oscillators, as suggested in Section 4.4.3.

6.4.1 Theoretical preliminaries

Let ξ_n be a stochastic process producing white noise. Then a Brownian motion (or random walk) in one dimension is described by

$$x_{n+1} = x_n + K\xi_n, \tag{6.36}$$

with noise intensity K. Often the white noise process ξ is assumed to have a Gaussian distribution, although it may have other distributions. The Brownian motion (6.36) may be thought of as filtering the white noise input signal with a marginally stable one-pole lowpass filter.

A stochastic time series materialises in a specific *realisation*, where each observation occurs according to some probability distribution. Time averages are usually calculated from the mean of a single realisation, whereas ensemble averages (at one particular time instant) are calculated from several realisations of the stochastic process. If the time and ensemble averages are equal, then the time series is said to be *ergodic* (equality of autocorrelations may also be a concern (Proakis and Manolakis, 2007), although we will not consider them).

Since times series from measured physical variables are often only available as a single realisation, one may hope for the variables to be ergodic so that estimates of ensemble statistics can be made. In computer simulation, however, several realisations can easily be generated and ensemble statistics may be estimated directly. This is what we will do in the following, since Brownian motion is a notorious case of a non-ergodic process.

For a simple example of Brownian motion, consider the case where $\xi_n \in \{-1, 1\}$ is a binary stochastic variable with $p(-1) = p(1) = 1/2$, and let $K = 1$ in (6.36). Then, the ensemble averages $\langle\xi_1\rangle = \langle\xi_2\rangle = \cdots = \langle\xi_N\rangle = 0$ are all equal, and the sum of these ensemble averages over time gives the ensemble average of x_n. Hence, $\langle x_n\rangle = \sum_{i=1}^{n} \langle\xi_i\rangle = 0$, so on (ensemble) average, x_n does not drift away from 0. However, the variance of ξ_i is 1, and from this it is possible to derive the famous formula for dispersion over time, namely that the standard deviation of x_n grows as $\sqrt{n}$ over time (Lemons, 2002). Simply stated, this means that the magnitude of x_n is approximately proportional to the square root of time if the initial value was $x_0 = 0$.

Brownian motion in the frequency of an oscillator is easily achieved by adding white noise to its phase variable. Consider an oscillator with nonlinear feedback FM given by a function g. The function g will be chosen such that the oscillator runs the risk of reaching a fixed point and to stop oscillating. In order to avoid this situation of oscillation death, a noise signal r is added:

$$\begin{aligned} y_t &= \sin(\varphi_t) \\ \dot{\varphi} &= \omega + g(y) + r_t \end{aligned}$$

In absence of the stochastic signal r, this system takes the form $\dot{\varphi} = \omega + g(y)$, where ω is the driving radian frequency. The fixed points must satisfy $\omega + g(y) = 0$. Since the range of $\sin(\varphi)$ is the interval $I = [-1, 1]$, any value $x \in I$ satisfying $g(x) = -\omega$ will be a fixed point. Assuming that g is continuous and that $\omega > 0$, the minimum of g needs to

be at or below $-\omega$ for a fixed point to exist. Various ways of introducing the noise signal may be considered, but we will use the form

$$r_t = f(y_t)\xi_t,$$

such that the noise intensity can be suitably scaled depending on the oscillator's output amplitude.

The addition of noise can induce oscillations with a certain distribution of period lengths in systems near a saddle node bifurcation (Sigeti and Horsthemke, 1989). For instance, the system $\dot{\theta} = 1 - \cos\theta + K\xi_t$ has a fixed point at $\theta = 0$ in the absence of noise, but may be brought into oscillation by turning on the noise. This system perfectly captures the essence of the situation that we will explore, namely that weak noise can be used to induce oscillations in some systems that would otherwise be stuck on a fixed point.

6.4.2 Nonlinear feedback FM with noise

The feedback FM oscillator with Brownian frequency modulation just introduced may seem contrived: First, we deliberately introduce the problem of amplitude death, then we solve it by injecting weak noise. But then the Brownian motion causes a new problem—or a valuable feature depending on your view—that the pitch fluctuates in an uncontrollable fashion. In fact, the pitch can be controlled to some extent, as will be demonstrated below.

To begin with, we will consider the feedback FM system

$$y_n = \sin\varphi_n \tag{6.37}$$
$$\varphi_{n+1} = \varphi_n + \omega(1 + g(y_n, \mu)) + K\xi_n. \tag{6.38}$$

The shape of the nonlinear feedback function $g(y, \mu)$ is determined by a parameter μ which will play an important role later; furthermore, the feedback function has been made independent of the radian frequency ω. Hence, in the noiseless limit $K \to 0$, the oscillation will always stop as soon as $g = -1$, whereas if g is identically zero for all y, then (6.38) becomes a plain sinusoidal oscillator. For the feedback function, we will use

$$g(x, \mu) = 2|x|^\mu - 1, \ \mu > 0 \tag{6.39}$$

which quenches oscillation whenever $x = 0$. Apart from modifying the shape of the waveform, this function changes the frequency of the oscillator. A few representative waveforms at various values of μ are shown in Figure 6.18 along with the corresponding graphs for g.

An approximation of the actual frequency can be found through the analytic expression for the resulting period length by setting

$$\dot{\varphi} = \omega(1 + g(\sin\varphi, \mu))$$

and integrating until one full period (or $T = 2\pi$) has been completed. Thus, the expression $\varphi(T) = \int_0^T \dot{\varphi} d\tau$ has to be solved for the variable T, such that

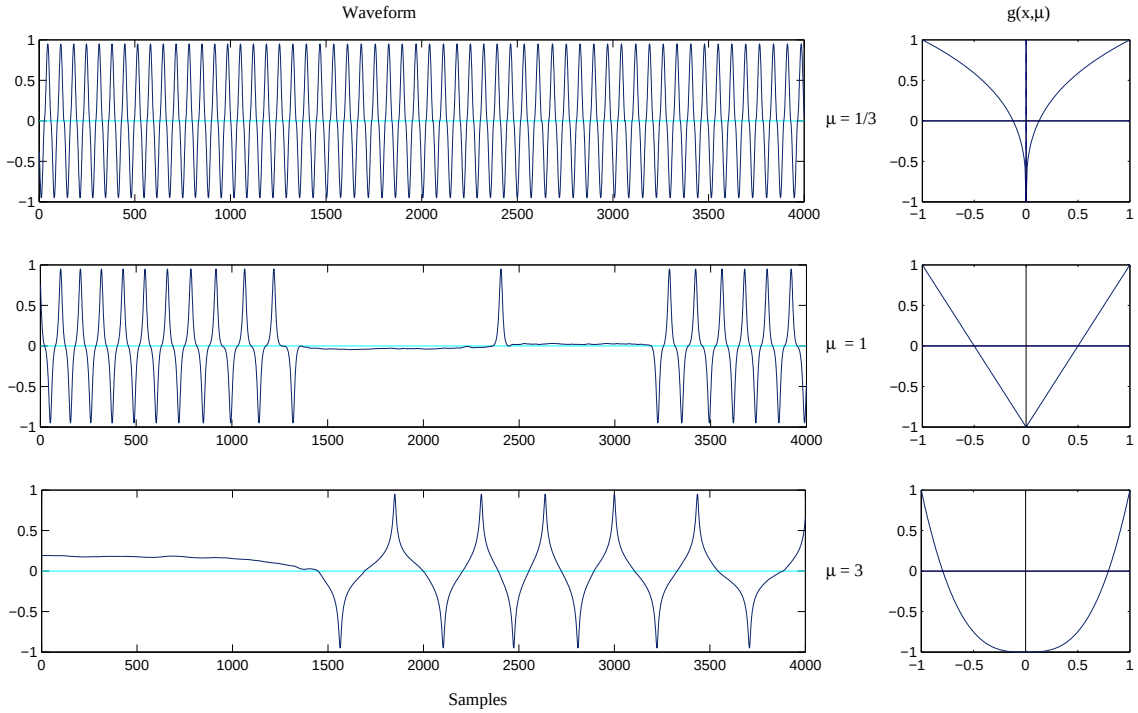

Figure 6.18: Typical waveshapes and the function $g(x, \mu)$ of eq. 6.39 for $\mu = 1/3$, $\mu = 1$, and $\mu = 3$ from top to bottom. Higher values of μ make the frequency variable more likely to dwell around 0 Hz. For $\mu = 3$ the intermittent bursts of oscillation are quite rare. The noise amplitude scaling function (6.41) has been used with noise intensity $K = 10^{-3}$.

$$\int_0^T \omega(1 + g(\sin\tau, \mu))d\tau = 2\pi. \tag{6.40}$$

By setting $\omega = 1$, the solution gives the ratio of the actual to the specified period. In the simplest case, $\mu = 1$, this integral can easily be evaluated; it reduces to

$$\int_0^T |\sin\tau| d\tau = \pi,$$

which yields $T = \cos^{-1}(3 - \pi)$. This multi-valued function has several more or less likely candidate solutions, such as $T_1 \approx 1.713$ or $T_2 = T_1 + \pi \approx 4.854$. It can be shown that the solution must satisfy $\pi < T < 2\pi$; hence the correct solution is T_2, although the prediction that the resultant period will be about 4.85 times longer than specified must only be taken as a first approximation.

The approximation of period length with (6.40) has one big flaw, though: in the noiseless case, the initial condition $\dot{\varphi} = 0$ ensures that the phase will never escape from

$\varphi = 0$, with a corresponding period $T = \infty$. The approximation has been derived for the limit $K \to 0$, but for stronger noise perturbation the actual frequency gets higher.

Next, we introduce a feeble perturbation whose strength is a function of the output. Thus, we take advantage of the fact that the oscillator only risks getting stuck when its amplitude is zero. For $|y| \ll 1$ the perturbation should be strong, whereas for large amplitudes perturbations are not needed. Such a function might be

$$f(x) = \frac{1}{B|x| + 1}, \quad B \gg 1 \tag{6.41}$$

where we set $B = 100$. Thus, we retain (6.37), and replace (6.38) with

$$\varphi_{n+1} = \varphi_n + \omega(1 + g(y_n, \mu)) + K f(y_n) \cdot \xi_n \tag{6.42}$$

where we have added a uniformly distributed noise $\xi \in [-1, 1]$ scaled by the inverse output amplitude (6.41). With added noise, the instantaneous frequency will wander off in some kind of Brownian motion, so the next concern will be how to control its actual frequency.

6.4.3 Frequency diffusion through Brownian motion

Let us simplify the model again, this time in order to study the evolution of the instantaneous frequency as a function of noise perturbation. The line of reasoning is analogous to that of solving stochastic differential equations for Brownian motion (Lemons, 2002), but fortunately we only need to derive the solution for discrete time, which is much easier.

The bare-bones case with no feedback term ($g(x) = 0$), constant noise ($f(x) = 1$), and a fixed carrier frequency ω_c is

$$\phi_{n+1} = \phi_n + \omega_c + K\xi_n,$$

with the resulting instantaneous frequency

$$\hat{\omega}_{n+1} = \omega_c + K\xi_n.$$

Now we can deduce the time evolution of the probability density function for $\hat{\omega}$ as follows. At the first time step the frequency is a sure variable, and the probability

$$p(\hat{\omega}_0) = \delta(\hat{\omega} - \omega_c)$$

is the Dirac delta function centered at the carrier frequency. Using uniformly distributed noise with a probability density function $p_K(\omega)$, it will cover the interval $I_K = [-K, K]$ with amplitude $1/2K$. Thus, the next time step will have a uniform probability distribution centered around ω_c, that is,

$$p(\hat{\omega}_1) = \delta(\hat{\omega} - \omega_c) * p_K(\omega)$$

which is the convolution of the two distributions. Continuing in this fashion, we get the recursive evolution

$$p(\hat{\omega}_n) = p(\hat{\omega}_{n-1}) * p_K(\omega), \tag{6.43}$$

so as time passes, the distribution at time n will be spread over a total maximum range of $(-nK, nK)$ in a shape that gradually approaches a normal distribution. In particular, there is nothing that stops $\hat{\omega}$ from attaining negative values or reaching above the Nyquist limit. When that happens, the probability density function of observed, positive frequencies in the range $[0, f_s/2]$ Hz will show the effects of foldover, causing the tails of the several times iterated convolution (6.43) to have higher amplitude than expected, had the same process been carried out on an unlimited range of real numbers.

Retaining the function g (6.39), but still using $f(x) = 1$, the above analysis no longer holds; in fact, an analytic treatment seems to be either impossible or very difficult. Similar difficulties arise if $g = 0$ and f is defined as in (6.41). Hence, we resort to simulations. Now, the frequency variable may be estimated from the generated signal y_n, which is perhaps the most obvious method. But then, there is the problem of time-frequency uncertainty; in order to obtain a fine-grained frequency resolution we need long time windows, but during those time-spans, the instantaneous frequency will be non-stationary, so ideally the window should be as short as possible. On the other hand, we have direct access to $\hat{\omega}_n$, which is known accurately at each sample. It turns out that the process of diffusion is so rapid that frequency estimations from the output signal can be unpractical for showing the time evolution of the distribution. However, the rapid fluctuation of $\hat{\omega}$ means that measuring it at a single point in time is not very representative, although this can be solved by averaging a few successive values. Figure 6.19 compares the two methods of obtaining histograms for the frequency distribution. Obviously the two histograms are strikingly different in shape. In particular, there are many excursions into negative frequency that are not visible in the frequency measured with ZCR, since the latter can only detect positive frequency.

Because of the design of the feedback function g (6.39), much of the time oscillations will cease since $\hat{\omega}$ will be close to zero, and more so the larger μ is. Hence, the distribution has a high peak at $\hat{\omega} = 0$, which increases in size over time. There is also a broad, bell-shaped peak somewhere near ω_c, which travels upwards in frequency over time.

This nonlinear feedback FM model switches between periods of oscillation and periods of locking to 0 Hz. The quotient of time spent in the oscillatory state clearly depends on the noise intensity as well as on the value of μ. Let us define the quotient of oscillation Q as

$$Q = \lim_{T \to \infty} P/T, \tag{6.44}$$

where T is the total time of a signal x, and P is the total duration over which $ZCR(x)$ is non-zero. Figure 6.20 shows estimates of Q as a function of noise intensity K for three values of μ. A remarkable feature of this figure is that for $\mu = 1/2$, there are two ranges of K that yield constant oscillation ($Q = 1$), but at intermediate values, the quotient drops. This phenomenon is remotely reminiscent of stochastic resonance (see Section 4.1.5 and Wiesenfeld and Jaramillo (1998)), or rather its opposite, since here, an anti-resonance is found for a certain noise intensity.

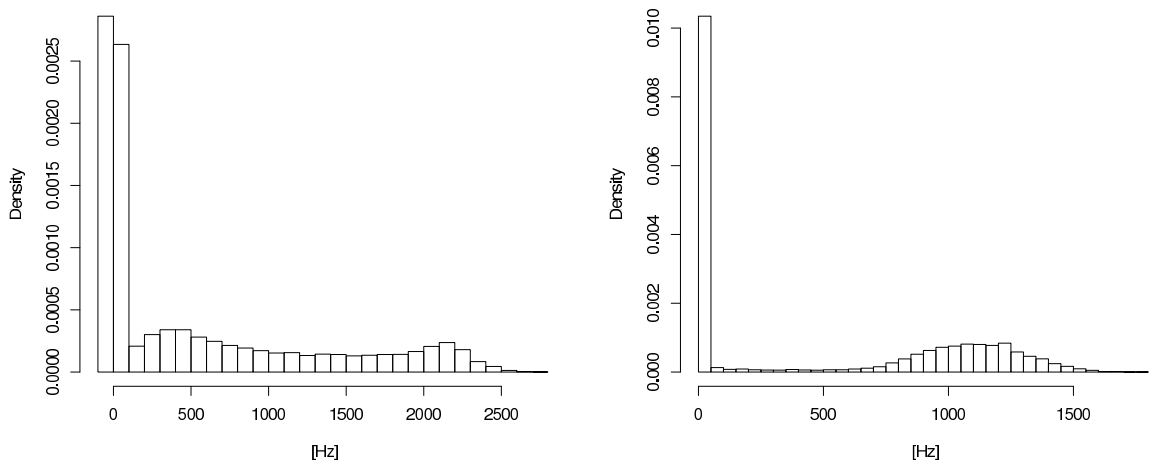

Figure 6.19: Comparison of histograms of frequency distributions for $\mu = 1$ and $K = 10^{-3}$ after 2 seconds at 48 kHz sampling rate with driving frequency at 1 kHz. Left: true instantaneous frequencies, i.e. $\hat{\omega}$ converted to Hz. Notice the larger deviations to high frequencies and that the bin with largest amplitude is just below 0 Hz. Right: frequencies estimated with ZCR using a 200 ms window. The distributions are taken from simulations over 10^4 runs.

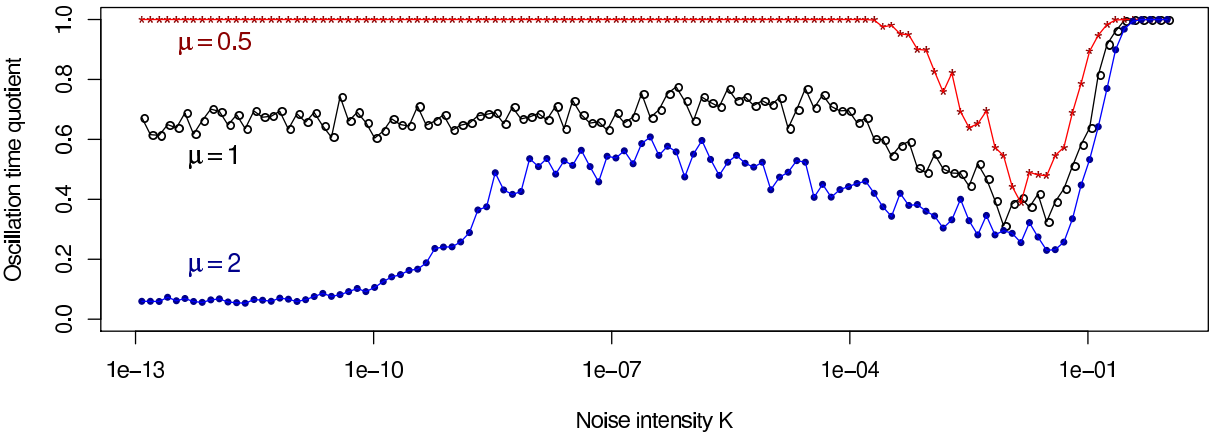

Figure 6.20: Oscillation time quotient as a function of noise intensity K, for $\mu = 0.5$ (red), $\mu = 1$ (black) and $\mu = 2$ (blue). Notice that the scale of the K-axis is logarithmic. Driving frequency is 1 kHz, $f_s = 48$ kHz. Each data point has been obtained from ZCR measurements (window length 200 ms) by averaging over 100 runs of 1.3×10^5 samples. For all plotted values of μ the curve saturates at a plateau for strong noise perturbations, meaning that the oscillation is constant. For slightly smaller noise intensity, all curves have a minimum of oscillation time quotient.

The ensemble averages used here have not been widely used for descriptions of sound synthesis techniques—perhaps they have not been used at all—and for an obvious reason: from the perspective of deliberate sound design, it is unpractical to have an instrument behave in entirely different ways each time it is played with identical parameter values. If it is unpredictable in the sense of being non-ergodic, then the ensemble statistics may not be very helpful anyway, since they cannot predict what will actually happen on a single realisation. That reservation notwithstanding, ensemble averages can be helpful at least in a design phase when one tries to get a picture of the range of possible outcomes. Otherwise, there might be some rather unlikely circumstances under which something goes terribly wrong with the instrument. Although such an instrument might seem well-behaved on a few casual tests, it might offer some serious surprise in the middle of a concert... The question of what is a fault in this kind of instrument depends entirely on the musical context, and in the particular context of experimental music there may be an apprehension that "anything goes", hence nothing really can go wrong. Indeed, the *fault tolerance* is high among some practitioners of electroacoustic sub-genres such as glitch (Cascone, 2000). But that is a different discussion, which we will not enter here.

6.4.4 Adaptive pitch control

Now we turn to the pitch control issue. By the inclusion of a simple pitch tracker this oscillator becomes a feature-feedback system. First, we will see how to stabilise pitch as far as possible, although this new system may also make for an interesting autonomous instrument with quite unpredictable behaviour. Without doubt, the injected noise is responsible for much of this, even when it is relatively feeble.

Let us introduce a ZCR frequency follower in the nonlinear feedback FM oscillator. The strategy will be to monitor the measured frequency $\hat{f} = \mathrm{ZCR}(y_n) \cdot f_s/2$ and compare it to the specified frequency $F = \omega_c f_s/2\pi$.

Since μ in (6.39) influences the frequency quite dramatically, this is the variable to adjust. For low values of μ the generated frequency will be too high; and conversely, for high values of μ the signal will spend most time close to zero with short pulses with long intervals between, thus lowering the generated frequency considerably. The actual adjustment function can be surprisingly simple. It takes three arguments: the specified and the measured frequencies, and the current value of μ. For the sake of convenience, both frequency variables should be in the same units, so we will use $\hat{f}$ and F in units of Hz. Then, the adjustment function takes the form

$$\mu_{n+1} = \mu_n + \phi(F, \hat{f}_n, \mu_n) \tag{6.45}$$

and

$$\phi(F, \hat{f}, \mu) = \begin{cases} +\epsilon & \text{if } F < (1-\alpha)\hat{f} \\ -\epsilon & \text{if } F > (1+\alpha)\hat{f} \text{ and } \mu > \mu^* \\ 0 & \text{otherwise} \end{cases} \tag{6.46}$$

so that μ is increased or decreased by some small amount $\epsilon > 0$ if the ratio between specified and measured frequency is greater than $1 \pm \alpha$ for some small, positive α, and

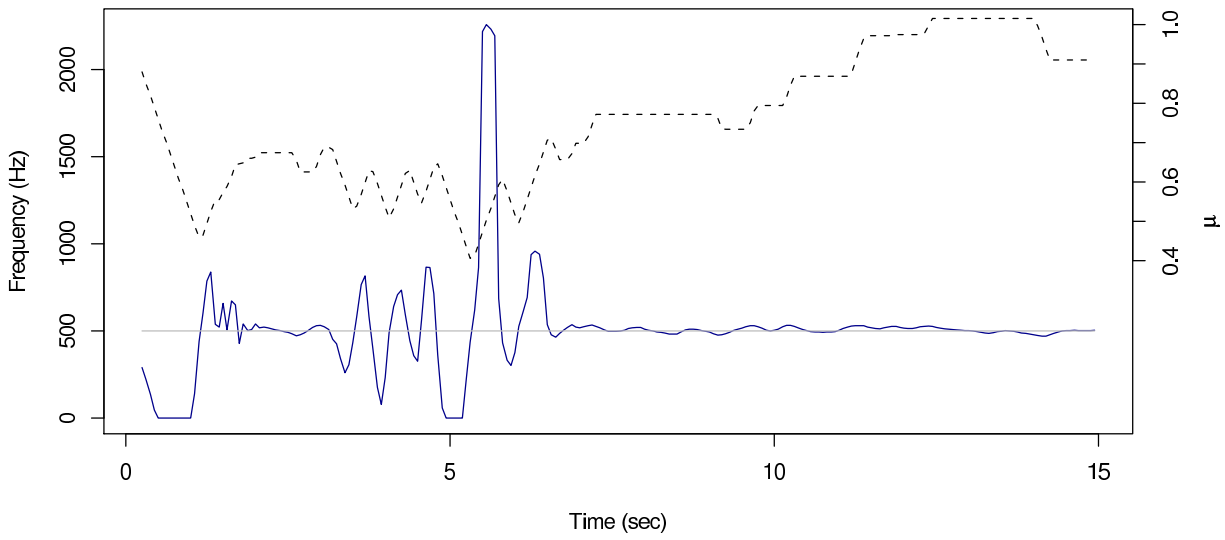

Figure 6.21: One realisation of a frequency trajectory (solid blue curve) and μ (dashed black curve). The frequency appears to converge to the specified frequency, which is 500 Hz, but stability is not guaranteed. In this run, the range of μ happens to be about $0.4 < \mu < 1$ (right axis), although there is no upper limit. The noise intensity is $K = 10^{-4}$.

$\mu^* > 0$ is a small constant.

Now, the question is, how well does this work in practice? If fact, it is far from robust. Parameter values have to be found that make this control scheme work; it is clear by trying a few haphazard parameter combinations that it does not give satisfactory results with just any values of α, ϵ, and μ^*. The ZCR window length L and the noise intensity K are also highly influential with respect to what kind of dynamics one gets.

According to (6.39), μ should not be negative; hence the check for $\mu > \mu^*$ in (6.46). There is a dead interval $(1-\alpha)\hat{f} \leq F \leq (1+\alpha)\hat{f}$, where μ does not get pushed in either direction. If it is too wide, the parameter μ will be able to meander endlessly and the pitch control will be poor. On the other hand, if the interval is made too narrow in comparison to ϵ, there will be constant perturbations that overshoot the equilibrium position. The noise intensity is a critical parameter; if it becomes too strong, the pitch control breaks down. Values such as $K = 10^{-2}$ are too high, whereas $K = 10^{-4}$ may work depending on other parameters. This explanation of the mechanisms of the pitch control scheme is certainly rudimentary, but we leave it at that and instead turn to a practical investigation.

In qualitative terms, a wide range of sound types can be found just by varying K. Now, we fix the parameters $\alpha = 0.05$, $\epsilon = 10^{-5}$, $\mu^* = 10^{-3}$, $L = 250\,\text{ms}$, $F = 500\,\text{Hz}$, and μ_0 is initialised to 1. Rapid frequency modulations occur for high noise intensities, and pitch control is impossible. For sufficiently low noise intensities the correct pitch can be reached, even if it may drift away momentarily. With the above given parameter values, sufficiently quiet noise means roughly $K < 10^{-3}$; then pitch control is achieved *on average* (over time), even though the pitch may be far off most of the time. It is crucial to note that two runs with the same parameters may result in utterly different sound trajectories (provided the random number generator is seeded with different initial values

each time). Figure 6.21 shows an example where the pitch control apparently works, if only successfully about 7 seconds into the sound. The frequency oscillates wildly before settling down around the specified value of 500 Hz, but there is no guarantee that it will stay locked to the specified value. For some parameter settings, quasi-periodic vibrato may be produced. It should be said that the single realisation in the figure is by no means representative; the next realisation may see the frequency variable dwindle down to zero.

Example 6.6. Two realisations with identical parameters demonstrate how μ may drift upwards or downwards from its initial value $\mu = 1$. The parameters are: $K = 5 \times 10^{-4}$, $\epsilon = 1.5 \times 10^{-4}$, $\alpha = 2 \times 10^{-2}$, and the carrier frequency is $F = 500$ Hz. In the first realisation, μ increases and reaches values around $\mu = 13.9$ towards the end, whereas in the second realisation, μ reaches the lower limit $\mu^* = 10^{-3}$.

The μ variable may drift to very high or low values, where small changes of its value are ineffective for pitch control. Surely better pitch control mechanisms may be constructed. For instance, it may be useful to regulate the noise strength dynamically, as well as the size of ϵ in (6.46). On the other hand, this oscillator produces quite interesting sounds even when the pitch control malfunctions, or precisely because of that.

6.5 The wave terrain model

In Chapter 3, a wave terrain model was introduced, which will now be extended with a feedback path. We rewrite the formula here with the parameters that will be used throughout this section:

$$x_n = a u_n v_n + b(u_n^2 - v_n^2) + c(u_n + v_n)^3 \tag{6.47}$$

As before, the path over the terrain is given by

$$\begin{aligned} u_n &= A\,\mathrm{osc}(F) \\ v_n &= B\,\mathrm{osc}(G) \end{aligned} \tag{6.48}$$

with time-varying parameters A, B, F, G, a, b, c. Thus, the wave terrain oscillator is the seven parameter system $x_n = \mathcal{W}(a, b, c, A, B, F, G)$. In order to avoid clipping, the amplitudes A and B of the sinusoids should be limited, and likewise with the constants a, b and c, so that $|x_n| \leq 1$. Once the two driving frequencies (F, G) are set, the other parameters only influence the spectral balance between partials. Qualitatively different sounds are obtained depending on whether the oscillator frequencies are harmonically related or not. Slight detuning from simple ratios tend to produce beats. With these sonic characteristics in mind, the question is, which feature extractors will capture some of this sonic variation best? There are many feature extractors that are apt for the task; the existence of beats might be captured as fast amplitude variation; potentially pitched sounds needs a pitch tracker; variations in spectral shape and harmonicity might be analysed; in fact, this model is sonically very rich and it exhibits variation in several timbral dimensions. We will not repeat the exhaustive search for suitable feature extractors that

was demonstrated in Section 6.3.4, but rather just pick two features without any deeper motivation.

6.5.1 Mappings from two features

Let us choose the centroid and RMS amplitude, both using sliding feature extractors, as the two attributes to analyse. The window lengths of these feature extractors can be set independently and to arbitrary lengths. The mapping is thus a function $f : \mathbb{R}^2 \to \mathbb{R}^7$ from centroid and amplitude to the seven synthesis parameters, only subject to the amplitude limiting constraints. There is an immense number of ways such functions can be constructed. Once the basic form of the mapping has been decided, the rest consists in tweaking its parameters. In fact, we will introduce a general form of the mapping which takes an even larger number of parameters.

Let $\hat{a}$ be the RMS amplitude of x_n, and $\hat{c} \in [0, 1]$ the centroid. If the signal's peak amplitude is restricted to $|x_n| \leq 1$, then the RMS amplitude is of course also guaranteed to be restricted to the same interval, but using the mapping that will soon be introduced, the peak amplitude may possibly exceed these limits. At this point we need to introduce some slightly unusual notation to denote that an index remains fixed, but the variable to which it is attached is selected from a set, or that both the variable and the index are taken from sets. Thus, we write $X_i, X \in \{A, B\}$ to denote the coefficients $A_i \cup B_i$.

For each of the variables a, b, and c we use the general 2-D quadratic map (which is of the same functional form as the system extensively studied by Zeraoulia and Sprott (2010), although here it is used in a completely different way):

$$x = x_0 + x_1\hat{c} + x_2\hat{a} + x_3\hat{c}^2 + x_4\hat{a}^2 + x_5\hat{c}\hat{a},\ x \in \{a, b, c\} \tag{6.49}$$

Thus, there are 18 coefficients in all for the variables a, b, c. Bounded functions such as cosine can be a useful ingredient in the mappings of the other variables:

$$X = X_0 + X_1\hat{c} + X_2\hat{a} + X_3\cos(X_5\hat{c} + X_6) + X_4\cos(X_7\hat{a} + X_8),\ X \in \{A, B, F, G\} \tag{6.50}$$

Hence, for A, B, F, and G, there are a total of $4 \times 9 = 36$ coefficients, which sums to a total of 54 coefficients for the complete mapping (6.49–6.50). Clearly the hitherto used strategy of exploring small dimensional parameter spaces breaks down at this point, and some automated search operation is called for. But first, a few observations are worth making before turning brute force computing power at the problem.

It will pay off to restrict the search space as much as possible before starting the search. In other words, we would like to find relevant ranges for each of the coefficients. The phase variables in (6.50) can be conveniently restricted to $X_i \in [0, 2\pi)$, $i = 6, 8$. Without loss of generality, the coefficients that influence the degree of nonlinearity can all be non-negative, thus $X_i \geq 0$, $i = 5, 7, X \in \{A, B, F, G\}$.

The amplitude constraints impose some limitations on the set of coefficients of the form (6.49), as well as on the oscillator amplitudes A and B. In order to avoid clipping, one might restrict the oscillator's amplitudes so that $|A| + |B| \leq 1$, whence it follows that $|a| + |b| + |c| \leq 1$ in (6.47) must also hold. These constraints then propagate down

to eqs. 6.49 and 6.50. Guessing adequate constraints, they might be that the sum of all the coefficients should add up to no more than 1 in both of these mapping functions, but in fact, such constraints may be overly cautious, resulting in very low output amplitude.

From a computational efficiency point of view, it is of course very costly to calculate each of the 54 coefficients while searching the parameter space, so there is a strong motivation to limit the number of coefficients as much as possible. The complete mapping consumes 26 multiplications and 8 calls to the cosine function per sample. Nevertheless, its general form lets us explore a large parameter space containing a wealth of qualitatively differentiated sounds.

Now, all that remains to do is to define the search criteria. To begin with, we actually do not know what to look for. But a random sampling of the parameter space is still possible, and as selected points of the space are visited, the sound at that point may be concatenated to a long sound file, and statistics may be collected. That is the strategy we will follow. Questions to ask now include the probability of unlimited amplitude growth as a function of the amplitude-related coefficients, and apart from that, we will record data from the centroid estimator.

6.5.2 Sampling the coefficient space

In order to make it convenient to sample the coefficient space $\{a_i, b_i, c_i, A_j, B_j, F_j, G_j\}$, where $i = 0, \ldots, 5$ and $j = 0, \ldots, 8$, it would be preferable if each coefficient had the same range. Unfortunately, this is not practical since the coefficients apply both to amplitudes and frequencies and are expected to occupy vastly different ranges. Nevertheless, there are groups of coefficients that can be searched within identical intervals.

An explosive amplitude increase will happen if the RMS amplitude $\hat{a}$ is amplified too much. This can happen in the coefficients x_i, $i = 2, 4, 5$, $x \in \{a, b, c\}$ and in A_2 or B_2. The constant coefficients x_0, $x \in \{a, b, c, A, B\}$ are also responsible for the total gain. While the centroid related variables contribute to the overall amplitude, they do so in a more controllable fashion since it is known that $\hat{c} \in [0, 1]$ will always hold. It can easily happen that the peak amplitude reaches well over 1 without the system actually becoming unstable. Henceforth, we adopt the arbitrary convention that if $|x_n| > 8$, this is taken as a sign that the amplitude increase is unbounded.

Negative coefficients can be useful in some places. Hence, we sample the coefficient space where $x_i \in [-1, 1]$, $i = 0, \ldots 5$, $x \in \{a, b, c\}$, $X_i \in [-1, 1]$, $i = 0, \ldots, 8$, $X \in \{A, B\}$ and, for the frequency related variables, $F_i, G_i \in [1, 1500]$, $i = 0, \ldots, 8$.

From a test with 10^3 randomly generated coefficient sets in these ranges, the amplitude blows up in about 45 % of the cases, and attains a high but stable level ($1 <$ peak amplitude < 8) in about 24 % of the cases. That is pretty bad statistics, indicating that coefficients cannot just be chosen at random from this range.

Apart from that, a general observation is that most sounds are characterised by turbulent noise, whereas a minority of the sounds are either pitched or inharmonic but not noisy, often with some modulation of the pitch profile. It is likely that the noisiness is caused by the rather high typical values of the coefficients $F_{5,7}, G_{5,7}$, which control the nonlinear feedback of the features into the frequency variables. When the amplitude blows up, the centroid cannot be calculated, so it is quite pointless to try to say anything

about general centroid values for this simulation.

Another idea that will now be explored is that the amplitude instabilities can be avoided by first generating the random coefficients, and then normalising them in an appropriate way. Therefore, we introduce the new condition

$$\sum_{i=0}^{4} |A_i + B_i| = 1 \tag{6.51}$$

for normalisation. It is more difficult to estimate the resulting peak amplitude of x_n in (6.47) as a function of all the rest of the coefficients, but we can try

$$\left| \sum_{i=0}^{5} a_i + b_i + c_i \right| = 1. \tag{6.52}$$

After normalising the coefficients and imposing the restriction $F_{5,7}, G_{5,7} \in [0, 24\pi]$, the general character of the sounds change to predominantly tonal, usually not noisy, but often with a gradually decreasing vibrato. The amplitude is typically softer, and clipping becomes rare. The normalisation with (6.51–6.52) taken together works in most cases; the amplitude only blows up in less than 4 % of the time, and the high but stable amplitude is reached in 4 % of the cases, based on a simulation with 10^3 runs. But then, the amplitude often becomes very low, which is the price one has to pay for avoiding instability. The average centroid over the randomly sampled coefficients is 0.22, although it spans the entire theoretical range $0 \leq \hat{c} \leq 1$, and 95 % of the centroid distribution is contained in $\hat{c} \in [0.012, 0.509]$. For the average RMS amplitude, its mean is 0.066, its range (excluding amplitude explosions) is $0.0 \leq \hat{a} \leq 1.407$, and 95 % of the RMS distribution is contained in $\hat{a} \in [0.0005, 0.558]$.

Example 6.7. The loudness varies a lot between different randomly chosen coefficients, but in this sound example of the wave terrain model, only those cases have been retained whose amplitude lies within a restricted interval. An assortment of pitched, inharmonic, and noisy sounds can be heard, many of which start with a characteristic transient in the form of a decaying vibrato.

6.5.3 On search by evolutionary algorithms

Huge parameter spaces such as the one we have just introduced are of course hard to search manually, but are well suited for exploration by evolutionary algorithms. This section, although completely based on armchair speculations, should hopefully clarify what may be worthwhile search strategies. For an introduction to basic concepts and various schemes of evolutionary computing, see Eiben and Smith (2003).

Engineering has traditionally been about deliberate design. Evolutionary computing, on the contrary, works by accidental design. Hence, one may not understand why a particular design works, it only happens to meet the specifications. As Toffoli (2000) has argued, it should require less knowledge and engineering efforts to design the fitness function than to design the intended object directly. In the case of autonomous instruments, intended to produce long and sufficiently interesting stretches of sound, the

fitness function is indeed very hard to design. In more realistic cases, however, such as the design of synthesis models that mimic certain short sounds, there are benefits of using evolutionary computing.

Evolutionary algorithms have been used with many fixed synthesis techniques where the problem is to optimise parameters for the closest possible match with a target sound. Wavetable synthesis, several kinds of FM, discrete summation formulas, waveshaping, additive synthesis, granular synthesis, physical modeling and even time domain waveform generation are amongst those synthesis techniques where evolutionary computing (and genetic algorithms in particular) have been used (Horner, 2007b). A more flexible approach is to start with a set of signal generators and operators that combine or process signals; then the matching process consists of the two steps of finding a synthesis graph and optimising its parameters (Wehn, 1998; Garcia, 2001). However, Horner (2007b) notices that genetic algorithms may work better if the search space is constrained by being limited to a single synthesis technique. But then one is absolutely restricted to the capabilities of the chosen synthesis model, which may not be capable of matching arbitrary sounds very well.

Let us consider the wave terrain model, and in what ways evolutionary computing might be used for searching its parameter space. At this point we have one basic ingredient: a scheme for initialisation of a random population consisting of different parameter values. The 54 real coefficients form the genome of our individuals. The phenotype in this case is the sound produced by running the synthesis model with the given parameters. Some kind of fitness function then applies to the generated sounds and forms the basis for further decisions about parent selection, as well as which individuals to remove or keep to next generation.

Most flavours of evolutionary computing use cross-over of genes; this would be straightforward to implement as a weighted sum of the parent's coefficients. There is nothing in the problem space that indicates that binary genes should be used (as is the case in genetic algorithms). Nevertheless, binary variables could be useful if each coefficient would come with an on-off switch. It has already been noted that it is computationally costly to run the full mapping; reducing it to as few non-zero coefficients as possible can thus be worthwhile. However, there is something arbitrary about the separation of the mapping into two distinct functional forms (6.49–6.50). This is a point where it would make sense to combine these two equations into a single, more general function which now would take 84 coefficients. Then, this combined parameter space could be searched, using some penalty for non-zero coefficients. It is crucial to note that there is a design question that no automated search of parameter spaces answers, and that is how to design the general form of the mapping function. Once it has been decided, an algorithm that searches this parameter space will not provide any "thinking outside of the box", that is, it will not come up with solutions that are not encodable by the genome. If a broader exploration is needed, then genetic programming should be used, where different functional forms may be tried out.

It is important to have an adequate representation of the solution, that is, the way the genotype translates into the phenotype. For synthesis models the question of representation is easy; they already have a number of parameters, sometimes even their preferred ranges are known. For the wave terrain model as much as for any synthesis model, there

is also the question of how synthesis parameters relate to perceived characteristics of the sound. A case in point is (6.48) which asks for two frequencies, F and G. Just like in ordinary FM, it would be more convenient to introduce an harmonic ratio $H = F/G$, similar to the control-to-modulator frequency ratio often used in FM. After all, it is this ratio rather than the individual frequencies of F and G that determine the degree of inharmonicity.

Still, the fitness criteria are missing. Using a sound file of, say, a woodwind tone as input would probably work if the goal was to minimise some distance function between the wave terrain model's sound and the target, but that is not our goal. Indeed, we still lack automatable criteria to be used as fitness functions if the goal is to produce non-stationary sound processes that are sufficiently interesting to listen to for some time. Interactive evaluation is the solution that has been proposed in such cases where no fitness function is explicitly known. Interactive evolution has been used in the search for aesthetically pleasing visual designs (Coley and Winters, 1997), where it is not too hard to evaluate a small population of images by displaying them side by side on a computer screen. Dahlstedt (2001) proposed interactive evolution of synthesis techniques where the user has to select among a number of sounds, aided by visual representations of the sounds.

On the downside, interactive evolution can be very time consuming, although it may be time well spent if its side effect is considered: the user gets acquainted with a broader palette of the available sonic characters in the synthesis model. Although genetic algorithms have been successfully applied for interactive evolution on relatively short sound files, they may be less efficacious for the search of long sound files as would be preferable with autonomous instruments. If no great gains of efficiency are to be expected from search aided by evolutionary algorithms, it does not mean that such methods are any worse than manual search.

6.5.4 The amended wave terrain system

Given the great variability of sounds as parameters are picked randomly in the 54-coefficient space (6.49–6.50), it seems like a wasted opportunity to fix any single configuration. Instead, it could be interesting to design a subspace of smaller dimension that captures some of the extreme possibilities of the full parameter space. But then, this step would require yet another layer of mappings and new control functions. This is an example of what we call the *embedding principle*: in order to design a more flexible autonomous instrument, new parameters and a layer of higher level mappings have to be introduced. In other words, it has to be embedded in some new parameter space, with a higher order control that acts on the free parameters that were constants in the previous version. However, we shall not design such a control scheme here, although a new layer of constant parameters will be introduced.

First, we make some modifications to the wave terrain system. The amplitude stability problem may be solved in a much simpler way than by finding stable solutions of the mapping. To that end, we introduce a clipping function

$$x_n = \arctan(\mathcal{W}(a, b, c, A, B, F, G)) \tag{6.53}$$

to the output of (6.47), which not only squeezes the amplitude to an appropriate range, but also acts as a distortion effect which increases the centroid a bit. (Actually, the output written to sound file is scaled further, but x_n in (6.53) is the signal that is sent to the feature extractors.) The harmonic index $H = F/G$ mentioned above will also be useful. Now, the new version of the mapping will be a slightly modified form of (6.49–6.50), where the wave terrain parameters are

$$\begin{aligned} a &= \eta - \hat{a} + \gamma(2\hat{a}\hat{c} - \hat{c}^2) \\ b &= \delta/3 + \hat{a} + \gamma(\hat{a}^2 - \hat{c}^2) \\ c &= \upsilon - \delta\hat{a}^2 - \gamma\hat{c}^2, \end{aligned} \tag{6.54}$$

the oscillator amplitudes are given by

$$\begin{aligned} A &= \upsilon - \delta(\hat{a} + \hat{c}) + \eta\cos(\Omega\hat{a}) \\ B &= \upsilon + (1 - \upsilon)\hat{a} - \delta\hat{c} + \eta\sin(\Omega\hat{c}) \end{aligned} \tag{6.55}$$

and the frequency related parameters

$$\begin{aligned} H &= 2 + 2\delta\hat{c}^2 - \gamma\hat{a} \\ F &= 600 - 900\hat{a}^2 + 1500\hat{c} + 800\cos(\Omega\hat{a} + \pi/6) \\ G &= HF, \end{aligned} \tag{6.56}$$

are now of a different form than the previously given system, as can be seen by multiplying out G. The new constant parameters, with typical values, are as follows:

$$\begin{aligned} \gamma &= 2.5 \\ \delta &= 1.4 \\ \upsilon &= 0.8 \\ \eta &= 0.7 \end{aligned} \tag{6.57}$$

and

$$\Omega = 2\pi h, \ h > 0.$$

Still, there are a few "hard coded" constants in (6.56), which could obviously have been formulated as parameters too. This form of the system offers sufficiently rich behaviour to merit a detailed study. As will be discussed in the following chapter, four sound examples realised with this system have been used in a listening study.

6.5.5 Quasi-periodicity and state transitions

Hitherto, when we have used two feature extractors in parallel (as in Section 6.2), they have been assumed to be of equal length. FFT-based features make it practical to have a single window length, since then, the windowing only needs to be performed once. With sliding feature extractors, however, there is no added cost or difficulty in having

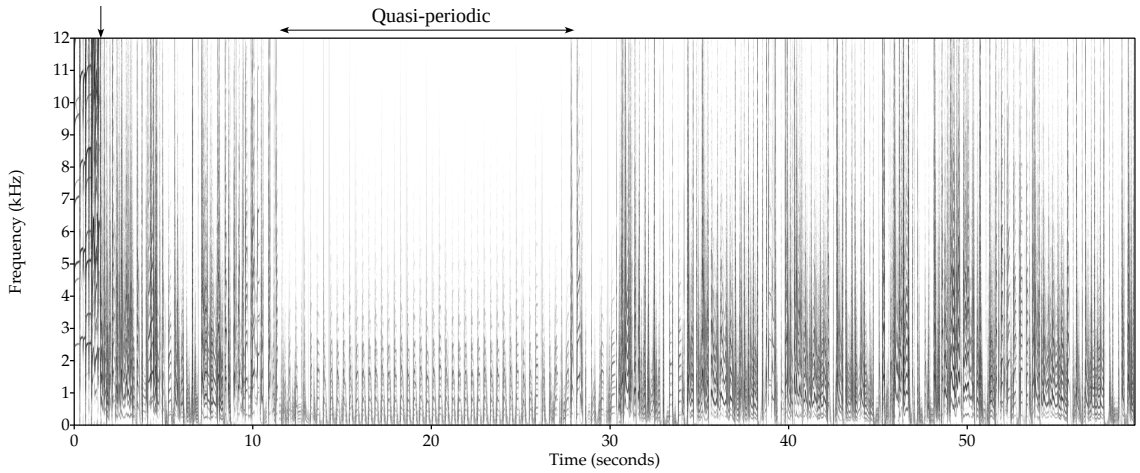

Figure 6.22: Sonogram of 60 seconds from the wave terrain system with the new mappings and $h = 12$. An arrow at 1.5 seconds marks the end of an initial transient. Very irregular behaviour ensues, until at 12 seconds a quasi-periodic cycle persists to 28 seconds, where the irregular character returns.

different window lengths. For the wave terrain system with the above mapping (6.53–6.56), eventually fixed behaviour in the form of a steady short-time spectrum is easy to find, as well as periodic cycles of patterns whose length is proportional to the feature extractor's window length. This is the case if the two windows, L_a and L_c for the RMS amplitude and centroid respectively, are of equal length. If their lengths are set to some ratio $L_a/L_c \neq 1$, then quasi-periodic patterns can arise. In particular, if this is an irrational ratio, there will be no exact repetitions in the cyclic pattern behaviour. It appears as if quasi-periodicity adds complexity to the system, perhaps by allowing escape from what would otherwise be an attracting periodic orbit. Incommensurate window lengths is no guarantee, however, for avoiding that the system eventually reaches a steady state.

We use an approximation to the golden ratio $\phi = (\sqrt{5}-1)/2$ as the proportion between the window lengths and set $L_a = 0.5$, and $L_c = 0.309$ seconds. A typical sonogram using $h = 12$ and slightly different parameter values than those given above (6.57) is shown in Figure 6.22. For this one minute example, there are clearly different regions as marked in the sonogram. A short initial transient of about 1.5 seconds is characterised by having more high frequency content than the rest of the sound. Then, some irregular patterns follow, but after 12 seconds, a quasi-periodic pattern with almost repeating impulses takes over, and at 28 seconds the irregularities return.

There is no indication that this alternation between the two pattern types settles on either, or stabilises on some other regular pattern—but perhaps it does. This alternation is characteristic of intermittent chaos, with its switching between periods of laminar phases and bursts. If this is the correct interpretation of the dynamics observed here, then the laminar phase would correspond to the quasi-periodic patterns and the turbulence would correspond to the irregular patterns.

In fact, when the values in (6.57) are used and only h is varied, the dynamics seems

to be limited to two types: either a short pattern is repeated, usually with minor fluctuations, or an irregular transient phase leads into a steady tone with decaying vibrato.

Example 6.8. The irregular transient may persist for quite some time, even over one minute, before it very slowly enters the decaying vibrato path and finally becomes a stable tone, as happens for $h = 25$ together with the above given constants (6.57).

After a number of tests of the wave terrain system with different parameter values, with h mostly in the range $h \in [6, 12]$, and also with some small variations of the constants listed above, a vague picture of some general traits begins to emerge. Sound files of up to six minutes duration were generated in order to gain an impression of long term dynamics. Certain types of qualitatively distinct textures and processes can be found across ranges of parameters. Two distinct patterns are present in Figure 6.22. The region labeled "quasi-periodic" (which we will abbreviate *Q-P imp.*) consists of slightly irregular impulses. Before and after that region, the behaviour is irregular (*IRR*). Another common pattern is the periodic or quasi-periodic very slowly decaying vibrato (*Decay*). Whenever this decaying vibrato is present, it eventually dies out (possibly after a few minutes) and becomes a static tone (–). A possible summary of observed transitions is given in Figure 6.23. Note that this diagram is only based on impressionistic observations, since the categories are not exactly defined and only a limited number of cases have been studied. The state begins in the node labeled Start, then goes either to IRR or to Q-P imp. From IRR, it may either become quasi-periodic again, or go on to the decaying vibrato, which is the indication of *game over*.

This classification into different categories is not much different from the task that faces the music analyst who tries to segment a piece of electroacoustic music. Here it is necessary to begin with intuition and introduce categories as seems fit for whatever perceptually grounded reasons. Then, it may be possible to approach the problem in a more rigorous manner, should one want to, by devising automatic classifiers that are able to distinguish the intuitively given categories. Finally, a systematic search through parts of the parameter space could be conducted in order to find which nodes are linked, and then a state transition diagram may be drawn from that information. Although this is a conceivable mode of investigation, it would be a formidable task to perform a full scale study of a system as complex as this wave terrain instrument. Similar state transition descriptions of dynamic systems were used by Crutchfield (1994) as a means to classify their complexity, although with such simple systems as one-dimensional chaotic maps, where the undertaking is complicated enough.

The impression of the wave terrain system is that, although it may produce very fascinating sounds, it is usually hard to find them; even more so, if one is looking for persistently irregular behaviour. After all, there is no way of knowing whether the system ever stabilises on a steady tone at a certain parameter value. The only thing one can do is to run the system for a very long time, but still, one cannot predict its future behaviour.

The steady state tone that is often reached may be annoying, but a way to harness its potential occurrence might be to use it as a cue to stop generating sound. Then, there would be a naturally given ending to the music generated by the autonomous instrument, and the instrument would turn itself off when it had reached its final destination.

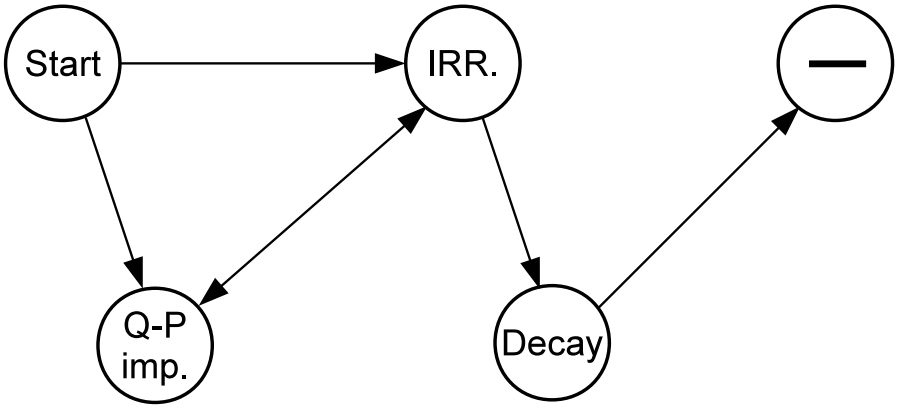

Figure 6.23: Hypothesised state transitions in the wave terrain system. IRR is irregular (chaotic) behaviour; Q-P imp is impulsive quasi-periodic sounds; Decay refers to a tone with a very slowly decreasing vibrato, until a fixed pitch is reached (end station; point of no return).

6.6 Conclusion

This chapter has focused on deterministic systems, with the exception of the noise-induced oscillations studied in Section 6.4. The scheme for pitch control of the noise-drived oscillator can easily be adopted in many other situations as well. The advantage of determinism is that, once the system equations, constant parameters and initial conditions are known, it is always possible to recreate the ensuing process. Deterministic systems are thus simpler to analyse than stochastic models. Nonetheless, randomness is a very useful resource in computer music.

We have made some use of spectral bifurcation plots in the study of how control parameters influence the behaviour of the system. From the limited number of cases studied, it appears that common bifurcation scenarios such as the period doubling route to chaos do not commonly appear in feature-feedback systems, although examples of crisis can be found in the cross-coupled map. Observations of the influence of filter length in the cross-coupled map led to the puzzling observation that the transient lengths tend to increase with the filter length, but in a highly irregular way with different regions of almost linear dependence (see Figure 6.7). Very long-lived transients were also found in the cross-coupled map as well as in the wave terrain system.

A method of searching for feature extractors that show a clear dependence upon synthesis parameters was described in Section 6.3.4. This method can be useful in any situation where the signal generator of a feature-feedback system is quite complicated by itself and its behaviour as a function of its control parameters is not well understood.

Although the feature-feedback systems contain a signal generator which contributes a forcing term, we argued that they nevertheless qualify as autonomous systems in the dynamic systems sense of the word. All units in a feature-feedback system are interconnected, and hence act upon each other. No external signal source is used to drive the system or to push it away from fixed points. This is largely a deliberate design deci-

sion. Although the inclusion of an external source used to modulate system parameters could be useful for obtaining more long-term variation, such mechanisms are slightly unidiomatic in autonomous instruments of the kind that we have developed here.

The purpose of the feature-feedback systems launched in this chapter was not to serve as useful instruments in the first place, but rather to illustrate analysis methods. Nevertheless, all of them may be further elaborated and improved. Of all the systems described thus far, the wave terrain system has been deemed to be the most interesting—from a purely subjective point of view of course, but related to its capability of generating perceptually complex patterns. Therefore, it was used for some of the sound examples in a listener study which will be presented in the next chapter.

6.6.1 Remarks on chaos

Chaos appears to be the guarantor of nontrivial dynamics in a feature-feedback system, and is arguably necessary for its output to be interesting over a longer time-span. This is not to say that chaotic dynamics is sufficient; as is well known from low-dimensional maps, the sound of a chaotic orbit is usually just white noise. The extended standard map $\mathcal{X}_M(f_c, K)$ is already chaotic when the coupling K gets strong enough, so in that case turning it into a feature-feedback system adds another layer that may introduce chaos where $\mathcal{X}$ has regular dynamics, or perhaps suppress chaos when it would otherwise be present, or it might have a neutral effect. What actually happens can only be known by estimating the Lyapunov exponents, and this has been done only in the case of the plain extended standard map.

Nevertheless, the embedding of the extended standard map into a feature-feedback system is necessary for non-stationary dynamics to occur over a longer timescale. This is an example of the embedding principle, stating that in order to get a more varied typology of sounds to occur over time, added control layers need to be inserted. It should be said that the embedding principle is more of a hypothesis; maybe there are counter-examples that will disprove it.

Given that the state space of feature-feedback systems is high-dimensional, the attractor could also potentially be of quite a high dimension. In contrast to Lyapunov exponents, the fractal dimension of the attractor is something that we have not even tried to estimate, although it would be interesting to do so. Thus, there is scope for further studies using nonlinear time series techniques.

At some parameters, the wave terrain system alternates over long periods of time between distinctly different types of behaviour. This could be a very complicated transient process that eventually leads to a final stable state with a steady tone, or it might be an example of intermittent chaos. The fascinating fluctuations between almost-periodicity and irregularity that is the hallmark of intermittency was pointed out by Pressing:

> It is certainly audible. In musical terms, the overall effect is like a variation technique that inserts and removes material from a motive undergoing mildly erratic pitch transformations, in the style of an adventurous but development-oriented free jazz player, perhaps (Pressing, 1988, p. 38).

In contrast to Pressing, who used the chaotic time series on the note level, we seem

to get an emergent higher level loosely corresponding to the note level, albeit not with conventional notes at stable pitches. There are fast time scales as well as very slow time scales, which articulate different patterns. One of the topics to be addressed next, is how to deliberately generate such higher levels of organisation.

Chapter 7

Designs and Evaluations

Given an initial condition and parameter values, it may nevertheless be difficult to predict the future behaviour of a deterministic autonomous instrument with any accuracy if it has not yet been fully studied. If the instrument is complicated enough, its dynamics will only be revealed by running it with specific parameters and initial conditions. Indeed, the reason for working with autonomous instruments is that they can afford some surprise, sometimes in a quite pleasurable way. The generated sound was not exactly what one would have expected, if there were any expectations at all. Nevertheless, as a composer or musician working with these unpredictable instruments, at some point one might like to tweak the algorithm so as to produce slightly different results that go more in some intended direction. Although this is often possible, it is usually a nontrivial task. This chapter explores some design strategies that may be useful to a certain extent when one knows what kinds of results to achieve. In particular, a few recipes for temporal variation or non-stationarity are introduced.

Working with autonomous instruments is fundamentally different from playing an acoustic instrument or writing a score by hand and deciding on everything from the overall form to details of phrasing and articulation. Autonomous instruments offer *global* controls, where parameter changes do not necessarily have a time-localised effect, but rather cause changes to occur all along the sound's duration. In fact, it is also possible to put autonomous instruments to use in a more predictable way and use them as any other synthesis engine. Only when the correspondences between control parameters and the resulting sound have been well understood can they be used in such a deterministic way, although that would be, so to speak, an unidiomatic usage of autonomous instruments. However, giving up the ambition of composing longer pieces worth listening to, at least one may generate textures with less temporal evolution. These textures may then be used as any other sampled sound fragments, as raw material for further arrangement.

This chapter begins with the idea that non-stationarity in some sense is a significant aspect of most musics, and techniques for detecting and eschewing stationary behaviour in feature-feedback systems are introduced. Next, some generalisations of the mapping functions are considered. The ideas about how to build in non-stationarity are presented abstractly, in a way that is applicable to many specific cases. Concrete examples will also be presented as we introduce two new autonomous instruments in section 7.2. These are based on the tremolo oscillator and discrete summation formula synthesis (both described

in Chapter 3).

These two new instruments and the wave terrain system from Chapter 6 were used to make a set of sound examples for an internet questionnaire. Somewhat flippantly, it was called the "Autonomous Instrument Song Contest" since participants were asked to vote for their favourite sound examples, although the study also contained questions about perceived complexity, simplicity, dislikings and other things (see Section 7.3).

Concatenative synthesis and feature-based synthesis are related to feature-feedback systems by their use of feature extractors. The output of an autonomous instrument may be rearranged by a "cut-up" technique using concatenative synthesis, as discussed in Section 7.4. Such rearrangements reveal how important the temporal organisation is for the character of the output of feature-feedback systems.

The theory of feature-feedback systems, as introduced in the previous chapter, was meant for signal level applications where the feature extractors operate on streams of samples. An important generalisation of the basic principles is to apply feature extractors on larger scale structures, in particular on the note level. What we have in mind when speaking of the note level is somewhat more general than common-practice notation and composition of instrumental music, although those applications follow as special cases. In the end of this chapter (Section 7.5), we hope to shed some light on feature-feedback systems by translating them to the note level.

7.1 Non-stationarity

In Chapter 5, some research on the perception of rhythmic complexity was reviewed. None of that is particularly well suited for characterisations of the temporal unfolding of an autonomous instrument such as those just developed in the previous chapter. Instead, something as simple as non-stationarity can prove to be a useful concept to begin with.

In time series analysis, stationarity is usually a requirement for making valid inferences and predictions (Kantz and Schreiber, 2003). A stationary time series is, roughly speaking, such that its mean and variance do not change over time. Examples include white noise, sinusoids, and coloured noise. It is obvious then, that music in all but some extreme forms breaks the stationarity requirement. Exceptions include La Monte Young's constant sine drone installations, and perhaps some uncompromising noise music.

The concept of non-stationarity as it is used in time series analysis is one thing, but we would need a similar notion that is better suited to music. Such a notion might be the recurrence of musical patterns at a given time scale. Instead of stationarity (in the time series sense) we refer to perceptual *stasis* of music when it is highly similar to itself over a certain time scale. Single notes may be relatively static internally, whereas the patterns of notes that make up a piece is often developing. On the other hand, repetitive minimalist music may be static when considered with the repeating pattern as the unit, but within that repeating pattern, there may be much variability on short time scales. These notions of stasis and change can be related to the informal notions of perceptual complexity discussed in Chapter 5. It is also straightforward to construct feature extractors that can be used to detect stasis, as we will show in this chapter.

Feature-feedback systems often seem to exhibit a typical time scale where there is

much variation, where patterns of some sort can be found. Over longer time scales the behaviour will be reminiscent of something heard in the past output. A way to investigate such scale dependence is to look at the scaling properties of flux, as suggested in Section 2.3.10, or equivalently, the average autocorrelation of the amplitude spectrum could be used since basically this just amounts to taking the complement of flux. The high level audio feature describing structural change on several time scales that was introduced by Mauch and Levy (2011) is related to perceived musical complexity (see Section 5.2.7). It may therefore be suitable for further analysis of feature-feedback systems and other autonomous instruments, although we will restrict ourself to much simpler means of analysis.

Considering deterministic feature-feedback systems, if they reach a fixed point or a repeating cycle, then their dynamics is stationary after the initial transient has died out. Stationarity entails perceptual stasis, but whether the converse is true or not may be less obvious. Consider Brownian noise. It has a fixed spectral profile, and will sound very much the same at any time. However, its mean over limited time windows will fluctuate, hence it is non-stationary.

The criterion of perceptual complexity which has been argued should be applied to autonomous instruments may include the ability of the instrument to take unexpected turns and to evolve in new directions rather than remaining bounded to a restricted range of sounds. A measure of non-stationarity that takes several temporal levels into account, such as the structural change feature, would be useful for the assessment of such a complexity facet. However, as discussed in the previous chapter, a deterministic feature-feedback system will approach an attractor, which necessarily occupies a limited region of the system's state space. This means that only a limited set of parameter values will ever be exploited by the system, and hoping for more variety may be in vain unless further control mechanisms are added to the system.

In Section (6.1.1), the feature-feedback system equation was introduced. As an approximation valid on short time scales, the signal generator $x_n = \mathcal{G}(\pi, n)$ may be considered to have constant parameters π, although its output x_n will be a time-varying signal. Then, the signal generator may be studied on its own as a dynamic system. However, we know that the parameters are not constant, but rather fluctuating. Dynamic systems theory was not conceived for the case of fluctuating parameters, but as Ruelle (1987) argued, some of it remains valid in case the parameters' fluctuation is slow and bounded. Then, the system $x_{n+1} = f(x_n, \lambda_n)$ with a slowly drifting control parameter λ does not have a single attractor A, but a family of attractors A_λ that are time dependent. Taking λ_n as the time-varying parameter vector to the signal generator of a feature-feedback system, it can be seen that Ruelle's idea applies to them. In contrast, in this chapter more abrupt changes of control parameters will be used. In that case, the dynamic systems approach such as estimating Lyapunov exponents or dimensions will most likely be less efficacious. However, the goal of breaking out of too restricted regions of the system's state space and introduce more variety may be reached more easily.

Next, several useful techniques for causing non-stationary behaviour in feature-feedback systems will be introduced, but first we present a useful tool for inspecting aspects related to musical form such as repetition, variation and contrast in audio signals.

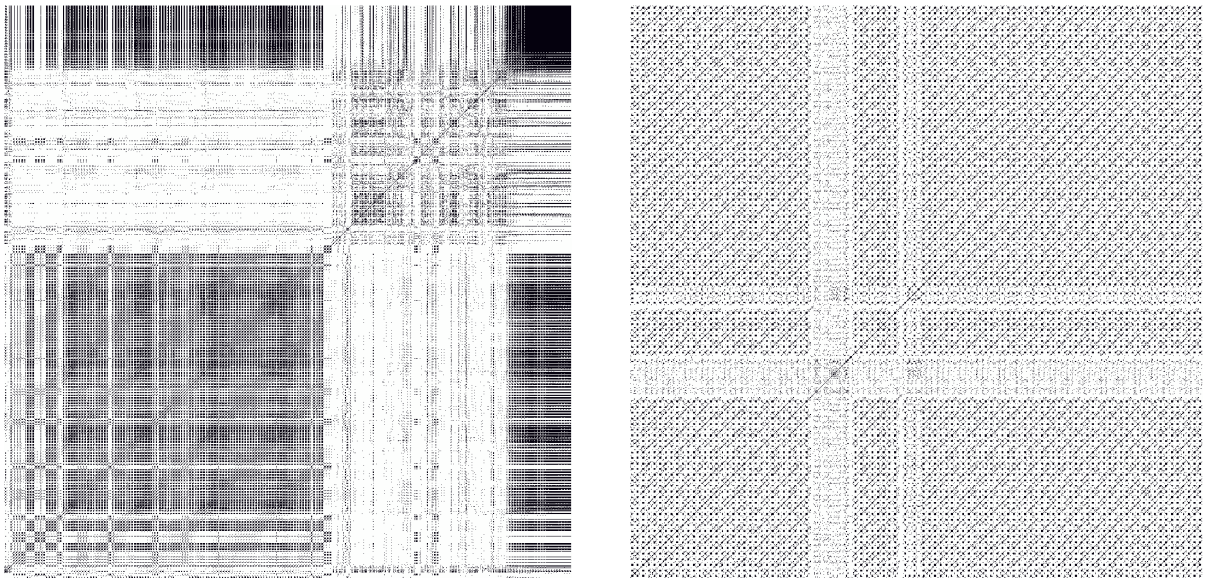

Figure 7.1: Recurrence plots of two sounds from the wave-terrain system, each 90 seconds long. Parts of the two sounds were used in the Autonomous Instrument Song Contest. Left: Ex. B2 showing three qualitatively different regions; right: Ex. B4 which is much more homogenous, except for two brief escapes from monotonicity seen as bands cutting across the plane. Dark areas corresponds to close recurrences.

7.1.1 Recurrence plots

Recurrence plots, a tool introduced by Eckmann et al. (1987) for the analysis of time series from dynamic systems, was later reinvented by Foote (1999) as a versatile analysis method for music that works on the audio signal level. The idea is simple: for each time point in the signal x_n, a distance metric $d(x_m, x_n)$ is calculated, where both m and n range over the entire time series or the portion of interest. If $d(x_m, x_n) < \epsilon$ for some constant $\epsilon > 0$, then a dot is plotted at the coordinate (m, n), otherwise it is left blank. Grayscales proportional to the distance may also be used. The main diagonal from low left to upper right is the distance of a point to itself, which always shows up as a line. Block patterns are common if some temporally contiguous segments are similar to each other. Hence, recurrence plots are useful for the analysis of large scale musical form. Contrasting sections are easily detected by visual inspection, but automated segmentation can also be performed (Foote, 2000); the idea is to use the correlation with a 2-D kernel that has the same form as a typical edge between contrasting sections in the recurrence plot and slide it along the main diagonal.

Figure 7.1 shows two recurrence plots, each of 90 seconds of sound from the wave-terrain system described in the previous chapter. The distance function is taken as the l_1-norm of a set of feature extractors that analyse fixed length segments of the signal. Different segmentations of music with contrasting formal sections may result depending on which feature extractors are used (Paulus and Klapuri, 2008). Thus, one can analyse recurrences with respect to, for example, harmony, rhythm or timbre separately. However, we are mainly interested in having a simple tool for the visualisation of large-scale

aspects of musical form in the output of autonomous instruments. Monotonous repeating patterns, initial transients, equilibrium states and sporadic deviations from otherwise steady patterns are easily detected (compare Figure 7.1).

Recurrence plots originally refer to plots of each sample of the signal or each point in a delay embedding space (Kantz and Schreiber, 2003). For long signals, this results in large matrices that are impractical to plot. The technique used in audio applications of recurrence plots, to use longer segments for each point in the plot, is what is called a "meta-recurrence plot" in the context of nonlinear time series analysis. Thus, recurrence plots can be used at different levels, depending on one's purposes. In music signal analysis, it would usually make little sense to consider the sample-wise recurrence.

Although recurrence plots are a bit harder to interpret than sonograms, they can be useful for displaying the formal contrasts and repetitions in a piece of music. By visual inspection, one can easily locate a behaviour such as the feature-feedback system approaching a steady state, as seen in the upper right corner of the left plot in Figure 7.1. Next, we will introduce a few techniques that can be handy for generating non-stationary behaviour and avoiding those fixed points.

7.1.2 Step sequencers

Sequencers are integrated into many synthesisers and form the backbone of much MIDI-based software. They offer a convenient storage for musical ideas that can be played back perfectly quantised and without error, and at speeds beyond human reach.

> People often have good musical ideas. But to learn an instrument can be really hard and after you've gone through that process, you've sometimes lost your ideas. With sequencers, people who aren't really professional musicians can do their stuff and put it into a good musical form. And not only amateurs. There were all these groups that couldn't play, and sequencers allowed them to do anything. (Manfred Rürup, quoted in Chadabe (1997, p. 203)).

Perfect quantisation being easy with sequencers, they are well suited for musical styles where regular beats are the norm, whereas expressive performance by deviations of timing must be achieved by other means. Another option that better fits our purposes is to control the timing autonomously from within the feature-feedback system.

The general idea of a sequencer is to associate a succession of states (such as MIDI note number, velocity etc.) with a series of time points. It can prove useful to integrate a sequencer into an autonomous instrument, because this is one way to assure that certain specified parameter configurations will actually occur. Otherwise it may be very difficult to design the system so that arbitrarily specified parameter configurations will ever be reached.

Specifically, let there be a sequence of parameter vectors $\{\pi\}_{n=1}^{N} = \pi_1, \pi_2, \ldots, \pi_N$ which are read through cyclically, so that π_N is succeeded by π_1. A traditional sequencer implementation would have fixed durations assigned to these parameter values, but to make it adaptive, we let the time points when the sequencer proceeds to its next step be determined by a feature extractor ϕ. In order to ensure that the next step will eventually be reached, we keep track of an accumulating sum of the feature ϕ_n, updated at each

sample, or rather a properly scaled version using a scaling function $f(\phi)$. Suppose we would like the sequencer to stay in one state over an unspecified number of samples, then jump to the next state. For this to happen, the extracted feature should be scaled so that $0 < f(\phi) \ll 1$. Then, the sequencer's parameter π_{k_n} at time n is picked by the current index

$$k_n = \text{INT}\left[\sum_{i=0}^{n} f(\phi_i)\right] (\text{mod} N). \tag{7.1}$$

If there is any dependence at all of ϕ on the current synthesis parameters π_{k_n}, then the duration of each step will differ. The feature may vary over time for any other reason as well; the result is still a fluctuating duration of each sequencer state.

Technically, (7.1) is just a lookup table oscillator used at low speed. In lookup table synthesis interpolation between adjacent table entries is often used, whereas here it is not. The feature ϕ, here also used as the oscillator's phase, is the argument of a function f that is not necessarily linear. Hence, this is also similar to phase distortion synthesis (compare Section 3.3.5).

The sequencer design can be extended in various ways. Only some hints about useful techniques will be given here, but we leave out the concrete demonstrations. In software sequencers, drum machines and the like, there is often a level of short patterns that can be chained together into longer parts. Similar hierarchical constructions can easily be put to use in autonomous instruments. Another line of extensions leans more towards stochastic algorithmic composition. Instead of moving on one step in the same direction, the next sequencer state may be chosen among all the stored states. Again, this choice can be made a function of the current extracted features, or it could be picked randomly. Further, the stored parameter values of the vector $\{\pi\}_{n=1}^{N}$ may themselves be updated as a function of the dynamics of the system. It is most reasonable that this rewriting of the stored parameter values takes place at a slow time scale, while faster variation may be introduced by the usual method of feature-dependent parameter variation at the sampling frequency or at a somewhat slower control rate. This is analogous to what goes on in Xenakis' GENDYN programme, although we have considered the operation in more general terms here. In GENDYN, the breakpoints of the polygonal waveform are updated for each iteration. If we compare those breakpoints to sequencer cells, the similarity of the two approaches should be evident. The step sequencer concept can be combined with virtually any autonomous instrument.

7.1.3 Permutations on real-valued functions

An alternative (or complement) to the step sequencer is to use an iterated map that gives the next parameter state. As with the step sequencer, this map will be designed so as to cycle through a fixed number of states. The following construct is loosely inspired by a proof concerning the existence of certain period lengths in continuous maps on the real interval (Elaydi, 2008, p. 99). Maps that have, say, orbits of period five but not period three are constructed in the so-called converse of Sharkovskii's theorem (cf. Section 4.3.3). Our construction will however be slightly different and will usually involve discontinuous

functions. Specifically, the iterated function will be based on a permutation σ of some set of integers $J \subset \mathbb{Z}^+$, such that $j_{n+1} = \sigma(j_n),\ j \in J$.

Recall that a permutation is a function on a finite set of elements into the same set of elements; for example, the permutation

$$\sigma : \begin{pmatrix} a & b & c & d & e \\ b & c & e & a & d \end{pmatrix} \tag{7.2}$$

takes the element c to e; elements of the upper row are mapped to those in the lower row. As this permutation is repeated from an initial element, it traces out an orbit. For example, starting from the element a, this orbit can be written in the cycle notation as (a, b, c, e, d). In general, the orbit $j_n = \sigma^n(j_0)$ may have a maximum period equal to the number of elements in the set J, although the period may be shorter.

Let each element in (7.2) be associated with an interval, such that $a < b < c < d < e < f \in \mathbb{R}$, where we have inserted a new end point f. Now, the permutation will be used to construct a real valued connecting function $\Pi : [a, f] \to [a, f]$ such that points that belong to the interval $[a, b)$ will be mapped to the interval that starts at $\sigma(a)$ and so on. For example, using the permutation (7.2), points in $[d, e)$ would be mapped to $[a, b)$, as illustrated in Figure 7.2. As long as the function Π is restricted to lie inside the shaded boxes in the figure, any initial value will traverse the permutation and visit the boxes in the cyclic order determined by the permutation σ. Suppose the orbit of any element j_0 under the permutation σ has period p. Then, an initial point z_0 will return to the same box after p iterations of the map Π. However, in general there will be no exact returns, so that $\Pi^p(z_0) \neq z_0$, because the orbit generated by iterating Π may have a longer period, or it may be chaotic.

Note that the connecting function must have domain $[a, f]$, but does not have to be onto (that is, there may be some points in the interval $[a, f]$ that are not reachable by the connecting function). At the most extreme, the connecting function may just consist of constant segments in each box; that would reduce it to a permutation and nothing else. Thus, one can design a coarse scale cyclic pattern on which a finer scale dynamics will be superimposed. The fine scale dynamics is then determined by the connecting function.

The connecting function is easiest to conceive of as a sum of two maps, such that one map implements the permutation on integers and the other takes care of the small scale variations. Thus, let

$$z_n = j_n + \xi_n$$

be the sum of a permutation of the elements $j \in \{0, 1, \ldots, p-1\}$ and $\xi \in [0, 1)$. Then, the map

$$z_{n+1} = \sigma(j_n) + f_{j_n}(\xi_n) \tag{7.3}$$

uses different functions f_j depending on the current value of the permutation (for simplicity, the same function f might be reused for all j). The idea of simultaneous variation on multiple scales is familiar from fractals. Here, the construction is limited to two scales, but this is sufficient for the purpose of introducing simultaneous variation on a coarse and a fine scale. In any case, it is straightforward to add further scales of variation.

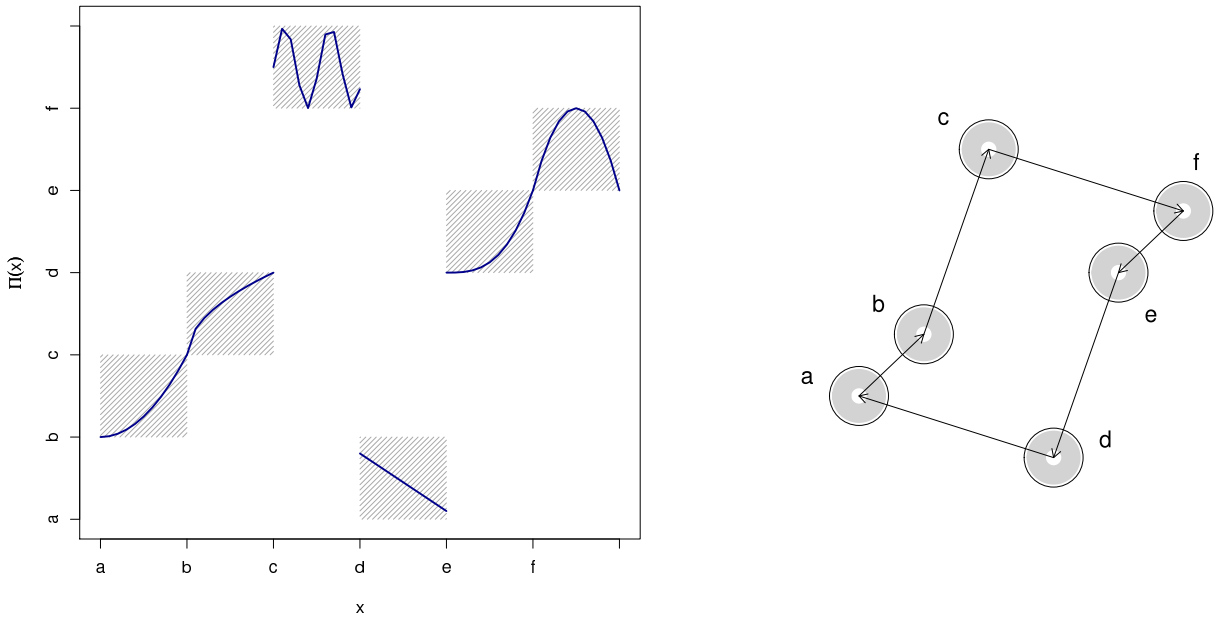

Figure 7.2: Permutations as real valued function (left) and directed graph (right).

The permutation map (7.3) can be combined with the step sequencer (7.1) by using the latter to select time points when the permutation map will be triggered. A simple demonstration of this combination of the permutation map with timings given by the step sequencer can be made using the sinusoidal oscillator introduced at the beginning of the previous chapter (Section 6.1.2). Again, the squared cosine map

$$f_n = F \cos^2(C_n \hat{q}_n)$$

is used for the oscillator's instantaneous frequency f_n, where F is a constant, the estimated frequency is $\hat{q}_n = \mathrm{ZCR}(x_n)$ and $x_n = \mathrm{osc}(f_n)$ is the output of the oscillator, but this time the variable C_n will be updated by the new scheme including the permutation map. From the experiments with the squared cosine mapping in the previous chapter, it was found that the variable C could profitably take quite high values, such as up to approximately 1000. However, if we use a permutation with few elements, the range of the map (7.3) will be correspondingly small—specifically, it will have the range $z \in [0, p)$ for a permutation of p elements. Therefore, we scale up the variable C by multiplying with a constant G:

$$C_n = (j_n + \xi_n)G \tag{7.4}$$

Example 7.1. As an example, we use five elements in the permutation, and the logistic map at parameter values $r \in [3.5, 4]$ for the small scale variation ξ, and let r be a function of j_n. The step sequencer, taking the zero crossing rate $\hat{q}$ as its input feature extractor, determines when the permutation map (7.4) is called.

Much more varied behaviour results with this system than by using the sinusoidal oscillator with C constant as we did in Chapter 6. Still, the window length of the ZCR

extractor is a strong determinant of the resulting dynamics, as is the average duration between parameter updates given by the step sequencer. The shape and amplitude of the function f in (7.1) influence the latter time scale in a straightforward way.

If a chaotic map is added as the ξ component to eq. 7.4, then it may appear unnecessary to use the permutation component, since the range can be suitably scaled up by G. The benefit of adding the permutated integer component j is that it will guarantee that several regions (the gray boxes in figure 7.2) will always be traversed. Cyclic traversal of a limited number of parameter states with exact repetition (as in a simple step sequencer) becomes redundant after a while. Thus, the particular form of the map (7.3) adds deviations to the cyclic periods in a way that may even render the underlying periodicity unnoticeable.

Finally, a comment about the construction of the permutation is in place. Some permutations can be decomposed into disjunct cycles, such as

$$\sigma : \begin{pmatrix} a & b & c & d & e \\ b & a & e & c & d \end{pmatrix}$$

which decomposes into the cycles (a, b) and (c, e, d). This means that the complete set of elements will not be traversed by the orbit σ^n, and that the cycle that will be generated depends on the initial element from which the iteration is started. Usually, this would be a somewhat wasteful construction, although it could be employed in order to let the user select initial conditions among different cycles, which may represent very different outcomes.

7.1.4 Statistical feedback by adaptive thresholds

The next idea has to do with keeping track of the statistics of some property of an autonomous instrument. Instead of merely observing the statistics, we will deliberately control it. There is an interesting predecessor of this strategy that is worth mentioning first.

While some composers wrote serialist music strictly adhering to 12-tone rows, there were other approaches to atonality where the pitches were more freely chosen, but under the constraint of avoiding repetition of recently used pitch classes. James Tenney in particular developed algorithms for this compositional technique, called *dissonant counterpoint* (Polansky et al., 2011). Tenney used randomly generated note sequences, but collected statistics of the distribution of recently generated notes in order to rebalance the observed statistical distribution in favour of notes that had not occurred for some time, or conversely, in disfavour of notes that had recently occurred. Tenney's method of statistical feedback, as described by Polansky et al. (2011), is identical in intent to the one that will be detailed in this and the following two sections; the implementation is however slightly different.

Suppose we have a feature-feedback system that settles on some more or less static pattern after an initial transient; further suppose that there are two different equilibria that may be reached where the synthesis parameters π_n either take the values A_1 or A_2. Now, we would prefer that the system does not settle for one of these states, but keeps switching back and forth between them. Such a perpetual vascillation between two states

is likely to hold a listener's attention for longer than a steady fixed point solution. There is a way to design the system so that this switching behaviour will be persistent. This is easiest to demonstrate in the case of two states, but a similar approach can be taken when there are a higher number of states to alternate between.

Given a feature-feedback system, we consider the case where the mapping function is essentially a switch with a time-variable threshold T_n. Let x_n be the output of the signal generator $x_n = \mathcal{G}(\pi_n)$, where $\pi_n \in A = \{A_1, A_2, \ldots, A_N\}$ is the current parameter vector, and let $\pi_{n+1} = \mathcal{M}(\pi_n, \phi_n, T_n)$ be the mapping function, where ϕ_n is the output of some feature extractor. In its most reduced form, we could write:

$$\mathcal{M}(\phi_n, T_n) = \begin{cases} A_1 & \text{if } \phi_n < T_n \\ A_2 & \text{if } \phi_n \geq T_n. \end{cases} \tag{7.5}$$

Now, the task is to adjust T at every time step so that the balance between A_1 and A_2 gradually approaches the desired proportion. If the feature value lies in an interval, say $\phi \in [0, 1]$, then A_1 will always be chosen if $T_n = 1$ for all n, whereas only A_2 will occur if $T_n = 0$ for all n.

A priori, given a constant T, it would usually be hard to predict the ratio $r = A_1/A_2$. For a given T, it may happen that only one of the A_i ever gets chosen, or their distribution might be highly skewed. If the goal is to eventually reach an even distribution of these two outcomes, their accumulated distribution over time has to be monitored (by "accumulated over time" we mean that the distribution is taken from the start time to the current time, not to be confused with the *cumulative* distribution!). Alternatively, the distribution over a recent limited time segment could be monitored, as we will do in the following section. Thus, there is a probability distribution function $p(A, n)$, evolving over time, which tells how often each of the states A_i has occurred in the past. The distribution for the more general case of an arbitrary number of different states is given by

$$p(A_j, n) = \frac{1}{n} \sum_{k=0}^{n-1} \chi_{A_j}(\pi_k) \tag{7.6}$$

using the characteristic function

$$\chi_E(x) = \begin{cases} 1 & \text{if } x \in E \\ 0 & \text{if } x \notin E. \end{cases}$$

Here, the time-accumulated distribution is taken from the initial time step. The formula (7.6) will be referred to as the *total accumulated distribution.* How this works is best demonstrated with an example.

7.1.5 Sinusoid two-state oscillator

For a concrete illustration of the principle, consider again the sinusoidal oscillator with feedback from a ZCR frequency analyser. The oscillator is given two alternating frequencies, F_1 and F_2. So far, this is also very similar to the tremolo oscillator (see Chapter 3 and below in this chapter), but the mechanism for deciding when to swap frequency differs. Then we have

$$\begin{aligned} x_n &= \mathrm{osc}(f_n) \\ \hat{q}_n &= \mathrm{ZCR}(x_n) \\ f_{n+1} &= \begin{cases} F_1 & \text{if } \hat{q}_n < T_n \\ F_2 & \text{if } \hat{q}_n \geq T_n \end{cases} \end{aligned}$$

and for the threshold, it is updated with

$$T_{m+1} = \begin{cases} T_m + \epsilon & \text{if } p(F_1, m) < Qp(F_2, m) \\ T_m - \epsilon & \text{if } p(F_1, m) > Qp(F_2, m) \end{cases} \tag{7.7}$$

at some suitable control rate such that $m = Kn$ for some $K > 1$, which determines the minimum possible note durations. The small parameter $\epsilon > 0$ influences the speed of the process of reaching the prescribed equilibrium, here parametrised by Q: the oscillator frequency F_2 will occur Q times as often as F_1. This control strategy is a kind of homeostat, similar to the adaptive pitch control described in the previous chapter (Section 6.4). In the present application, there are no stochastic disturbances and the distribution converges very rapidly when using (7.6) as the reference distribution.

The other alternative previously mentioned is to monitor only a recent segment of the past signal, yielding a distribution

$$p_L(A_j, n) = \frac{1}{L} \sum_{k=1}^{L} \chi_{A_j}(\pi_{n-k}) \tag{7.8}$$

over the last L samples. Some tests indicate that this short term distribution easily causes more fluctuating behaviour and, depending on other parameters, the total accumulated distribution (7.6) may not reach its specified balance. For instance, if Q is far from unity the distribution will be highly uneven. Then, to capture the proportions of F_1 and F_2 with some accuracy, L would have to be long enough to observe a few instances of the least often occurring state. As Figure 7.3 demonstrates, the control scheme using short term distribution begins with a transient before the proportion of F_1 and F_2 stabilises. It can be seen that the total accumulated distribution levels out at a lower level than the specified Q. With the total accumulated distribution, however, the proportion would rapidly converge to Q.

Arguably, the total accumulated distribution is the most appropriate measure to use, although from a perceptual standpoint, it can be argued that the most recent temporal segment is what matters most. In other words, we probably do not keep track of the accumulating proportion of the two notes from very far ago, although we may have a sense of their average proportion during the last few seconds. Hence, the short term distribution, if taken over a suitable duration, is motivated for perceptual reasons alone. Another argument in favour of the short term distribution is that when it fails to converge to a limit, it adds extra complexity for free.

Although the sinusoidal oscillator is not the most exciting synthesis model, it serves well as a demonstration of the basic ideas. For this scheme to work, it is of utmost importance that the feature ϕ varies over some range. Actually, it does not have to

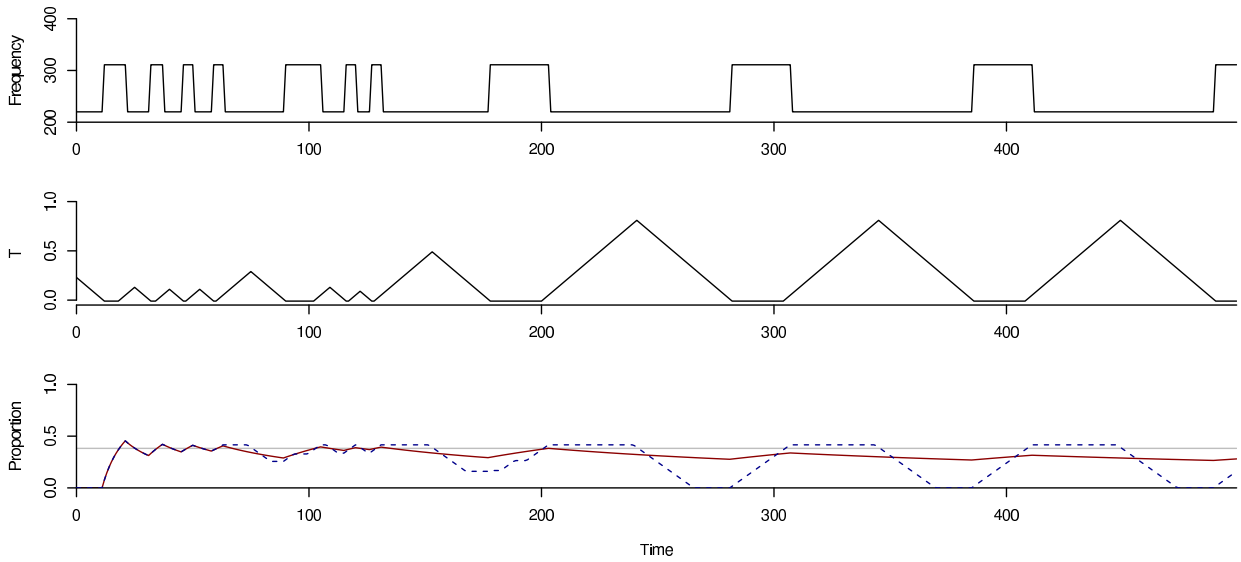

Figure 7.3: Alternating frequencies (top), adaptive threshold using control by short term distribution (middle) and proportions of the two frequencies (bottom), with the solid red curve showing the total accumulated proportion, and the dashed blue curve showing the ever fluctuating short term distribution. The gray line is the specified level of the proportion, Q, which is not quite reached.

capture the variation of the synthesis parameters as long as ϕ keeps changing for whatever reason. Only if the feature used remains constant does the scheme break down. To see this, consider again the above example, but replace the ZCR unit with an RMS amplitude follower. Now, apart from an initial transient, the RMS amplitude should remain constant no matter how much the control parameter (frequency) is cranked up or down. Restoring the control in that case can be as simple as inserting periodic amplitude modulation.

Such a basic model as this sinusoidal oscillator can hardly do full justice to the adaptive thresholding technique. Let us just note that adaptive thresholds become very useful in more complicated feature-feedback systems. In particular, they may be used in a nested fashion, where the generator is itself a feature-feedback system with parameters whose distributions may be rebalanced on a long time scale in order to introduce large-scale variability. Besides, it can be applied to any parameter, not just pitch.

7.1.6 Multi-state generalisation

As we have seen, using the total accumulated distribution (7.6) the parameter values asymptotically approach the prescribed distribution. Now, let us look at the generalisation to the case where the parameter may take on several discrete values A_j, $j = 1, 2, \ldots, N$. We insist on keeping the discussion abstract and applicable to any parameter and set of values whatsoever; for a suggestive example consider the A_j as the twelve chromatic pitch classes, their associated probabilities being equal as in the technique of dissonant counterpoint—just to have something to hinge the ideas on. The following scheme has only been worked out in theory and remains to be tested.

The probability density function $f(\phi)$ of a feature occupying the range $T_0 < \phi < T_N$

is used to formulate the probability that the feature lies in the interval $\phi \in [T_j, T_{j+1}]$ as follows:

$$p(T_{j-1} \leq \phi < T_j) = \int_{T_{j-1}}^{T_j} f(\phi) d\phi \tag{7.9}$$

Now, for each A_j we associate an interval of the feature such that the desired proportion of A_j will be $p(T_{j-1} \leq \phi < T_j)$. Thus, the T_j now act as a series of thresholds that have to be nudged up or down so that each interval contains the specified proportion of ϕ. The mapping function now becomes

$$\mathcal{M}(\phi_n, T) = A_j \text{ if } \phi_n \in [T_{j-1}, T_j), \quad 1 \leq j \leq N. \tag{7.10}$$

Let $\Omega(j),\ j = 1, 2, \ldots, N$ be the desired distribution of parameter states A_j. When the specified distribution is reached, $\Omega(j) = p(A_j)$ holds, either with respect to the short term distribution (7.8), or at least over the total accumulated distribution (7.6). Evidently, the task is to pair the probabilities $\Omega(j)$ with intervals of ϕ over which the integral (7.9) evaluates to $\Omega(j)$. That is, we want to solve

$$\int_{T_{j-1}}^{T_j} f(\phi) d\phi = \Omega(j)$$

for T_{j-1} and T_j and for all the $\Omega(j)$ simultaneously. To this end, the cumulative distribution

$$F(x) = \int_{-\infty}^{x} f(\phi) d\phi$$

of the feature gives the breakpoints of the intervals. In particular, $F(T_0) = 0$ since $T_0 \leq \phi$ is the lower bound for the feature's values. By definition,

$$F(T_1) = \int_{T_0}^{T_1} f(\phi) d\phi \equiv \Omega(1)$$

and generally, it follows that $\Omega(j) = F(T_j) - F(T_{j-1})$. Hence, the interval breakpoints must satisfy

$$F(T_j) = \Omega(j) + F(T_{j-1}) \tag{7.11}$$

which gives a convenient way to estimate them. Thus, the breakpoints T_j will have to be regularly updated by solving (7.11), which plays the same role as the simpler balancing mechanism for two states (7.7).

The parameter states A_j and the rest of the system has not been defined, but this abstract scheme is applicable to any parameter and to autonomous instruments of any kind.

The technique of adaptive thresholds has been shown to work for two parameter states under reasonable conditions, although there may be cases where the scheme breaks down. At the very least, reasonable conditions means that the feature ϕ must vary over some

range over time. As long as this method works, an arbitrary distribution of a set of discrete parameter values can be obtained. It is interesting to note that this way, a desired entropy of the distribution of parameter values can be reached. If we want high entropy, all possible parameter values shall be equiprobable. Inasmuch as surprise has to do with building up and breaking expectations, highly uneven distributions having medium entropy will be the key. Then, there will be one or a few very frequently occurring parameter values, occasionally interrupted by other seldom occurring values.

As the short example of two alternating tones illustrates (Figure 7.3), after a while a steady distribution will tend to appear as a perceptual constant. In that case a limping tremolo resulted, which surely is more varied than just a single constant tone, but in the long run it too gets very predictable. Hence, one may be tempted to add yet another layer of adaptive thresholding to some higher level parameter. We are back to the embedding principle: to get more long range variation, the model's complexity needs to be increased by adding new components.

Tenney's statistical feedback algorithm is different than the method described here. In its simplest form, it can be carried out as follows: Initialise the probabilities $p(A_j)$ to a uniform distribution. (1) Select one element A_j randomly; (2) set the probability $p(A_j)$ to zero for the chosen element; (3) increase the probabilities of all the other elements and repeat from step (1). Further refinements of this algorithm are considered in Polansky et al. (2011). In contrast to Tenney's statistical feedback which assumes randomly generated elements, the method of adaptive thresholds was developed for a deterministic feature-feedback system.

7.1.7 Variable control rates

The rate of parameter updates is one of the most important factors in determining the overall dynamics of a feature-feedback system. There are mainly two types of parameter updating, given by the implementation of the feature extractor. It occurs either at the audio sample rate if it is a sliding feature extractor, or at a much slower block rate if it uses a sample buffer for FFT analysis. In the latter case, L samples are sent to the feature extractor at once, and every L samples it returns a new feature value. Although L may take arbitrary lengths if zero padding is used, we have restricted it to powers of 2, and the hop size is always identical to the window length. However, when the window length is set, it remains constant. This easily produces regular pulsed sounds whose durations are given by the window length. The effect can be clearly heard in several sound examples from the Autonomous Instrument Song Contest (see Section 7.3 below), particularly those in section A, where the window length is in fact the only parameter that is being varied.

Let us now consider the case of a sliding (time domain) feature extractor. Since rapid change is not expected in the output signal from a sliding feature extractor, some redundancy across consecutive output values may be expected. In other words, its output will probably have less than full bandwidth, so it may be downsampled by some amount without loss of information. If we know its bandwidth, say $B = f_s/2K$, we may downsample without prior lowpass filtering by keeping one out of K samples. Hence, this gives some flexibility in the choice of control rate, which may even be time-varying.

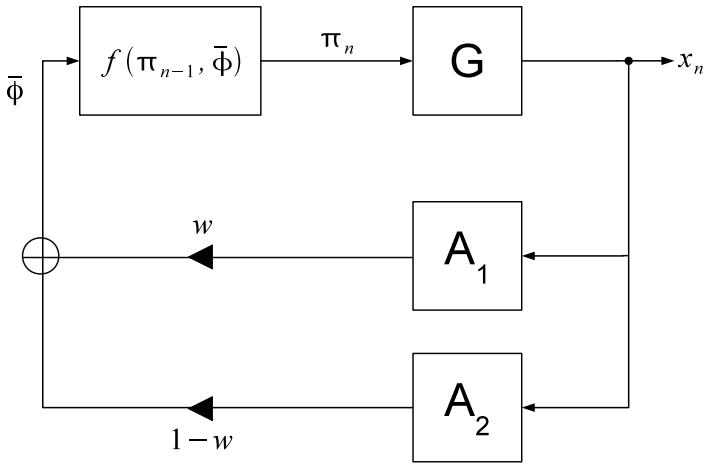

Figure 7.4: Feature-feedback system with two parallel feature extractors with different block rate parameter updates. A weighted sum of their outputs is used by the mapping function.

Even slower control rates are obtainable by lowpass filtering before downsampling. The feature extractor will also have adjustable time constants, so setting its time window long enough will obviate any need for downsampling. However, the feature extractors used here (and introduced in Chapter 2) are not suited for changing window lengths on the fly. Thus, when a window length has been fixed, the feature extractor has its rate set.

As an alternative to fixed length feature extractors, one might consider using two or more instances of the same feature extractor, each with its own window length $L_1 < L_2 < \ldots < L_N$. Suppose we use two different lengths L_1 and L_2 of a feature extractor $\phi_{L_i}(x_n)$. Now, if we want to vary the update rate received at the mapping function over time, we can mix the outputs of ϕ_{L_1} and ϕ_{L_2} by a weighting function $w_n \in [0, 1]$. Then, the weighted feature sent to the mapping component is

$$\bar{\phi}_n = w_n \phi_{L_1}(x_n) + (1 - w_n)\phi_{L_2}(x_n), \tag{7.12}$$

as seen in Figure 7.4. More than two length scales could be interpolated in a similar way by crossfading between different pairs of feature extractors $\phi_{L_k}, \phi_{L_{k+1}}$, although it is admittedly a bit wasteful to compute several feature extractors in parallel.

Assuming stepwise changes are wanted, another technique worth considering is *sample and hold*. As new values of ϕ_n come in, one of them is picked and held for a number of samples, then another newly arrived value is held and so on. Especially with sliding feature extractors, it may be useful to freeze one of the received values for a number of samples before updating the mapping. Thus, if the feature ϕ_n is updated for each audio rate sample, the mapping at time n receives $\bar{\phi}_n = \phi_{n-(n \bmod P)}$, the value from time $n - (n \bmod P)$ where P is the number of samples held constant. Evidently, it is easy to vary P over time as well, so as to change the updating rate of the mapping function. Sample and hold may introduce discontinuities in otherwise smoothly varying time series. If that should be a problem, the use of smoothing filters is an easy remedy.

7.1.8 Generalised mappings

As a simple example of the importance of parameter mappings in sound synthesis and effects, consider the case of a second order IIR filter, where a direct control of its coefficients is not very intuitive at all. Instead, the filter's centre frequency and bandwidth are useful control parameters, which can be mapped to the filter coefficients. Physical controllers are partly determined by their dimensionality, such as one degree of freedom for a slider and two for a mouse that points to a coordinate on a surface. Since the number of synthesis parameters may be different from the controller's degrees of freedom, the mapping from the controller to synthesis parameters may be "one-to-many" if a single dimension of the controller influences several synthesis parameters, or it may be "many-to-one" in the opposite case where several controller dimensions map to the same synthesis parameter (Van Noort et al., 2004). The mapping may also be "one-to-one" where each controller dimension maps to a single parameter, or there may be cross-mappings from several input dimensions to several synthesis parameters. Some interesting examples of the usefulness of such cross-mappings are given by Hunt et al. (2002), who note the shortcomings of simple one-to-one mappings. They also suggest the use of multiple mapping layers. It may be very useful to map perceptual dimensions to synthesis parameters and physical controllers to perceptual effects, insofar as this is possible.

Although the mappings inside feature-feedback systems serve partly different purposes than mappings from physical controllers, the idea of an interface from one kind of signal to another is still valid. These mapping components are not very capable in their simplest form, though. By introducing memory, however, they will be capable of monitoring such things as how long their output has repeated identical values. Consequently, this makes it possible to design the autonomous instrument such that it will escape from attracting fixed points. Obviously "escape from an attracting fixed point" is an oxymoron; if it can be left behind it is not attracting at all—this is akin to a black hole that one could escape from. The idea is rather to perturb the system if it has reached the fixed point π^* enough that it may evolve in another direction. Depending on the dynamics close to π^*, small perturbations may not be sufficient to alter the dynamics more than temporarily. Then, if the perturbed state $\pi^* + \epsilon$ is still within the basin of attraction for π^*, the system will return to the same fixed point again and again.

The terms mapping and map as they occur in different contexts are used in some entirely different meanings, which may cause confusion. In the mathematical sense, a map is a *function* from a domain to a codomain, where each element in the domain is mapped to exactly one element in the codomain. The mapping component of the feature-feedback systems has been assumed to be a map in this sense, but it will prove useful to generalise the mapping concept. Ordinary mappings are functions $f : \mathbb{R}^p \to \mathbb{R}^q$ that map p feature signals to q parameter values at some determined rate. If this happens at the sampling rate, the map is a function of the current feature vector: $\pi_n = f(\phi_n)$. Memory can be included in the mapping component in various ways. Several past samples of the feature signal can be used, such as in a FIR-like or non-recursive structure:

$$\pi_n = f(\phi_n, \phi_{n-1}, \ldots, \phi_{n-M}) \tag{7.13}$$

or we may even feed back the past parameter values, as in an IIR-like recursive form

$$\pi_n = f(\phi_n, \ldots, \phi_{n-M}, \pi_{n-1}, \ldots, \pi_{n-N}), \tag{7.14}$$

further, the mapping may be made non-autonomous, so that it becomes a function of its input as well as of time:

$$\pi_n = f(\phi_n, n). \tag{7.15}$$

The time dependent version may also be combined with the FIR or IIR-like forms. The filter-like mappings (7.13–7.14) are useful for such things as detecting repeated parameter states. The crucial point is that by introducing an internal state in the mapping, one specific input value ϕ_n may be mapped to different parameter values π_n depending on the current internal state.

Finite state machines or automata is another way to formalise the notion of mappings with internal states. A finite state machine is the quintuplet $\mathcal{M} = \{I, O, S, \nu, \mu\}$ where

I is the input alphabet
O is the output alphabet
S are the internal states
$\nu : S \times I \to S$ is the state updating function
$\mu : S \times I \to O$ is the output function.

Although finite state machines have input and output alphabets with a finite number of symbols, here the input and output are vectors of real numbers. Keeping the number of internal states finite, it becomes necessary to somehow coarse-grain the input signal when it is used in the state updating function and the output function. It will also be convenient to have several distinguishable internal variables each with a set of states S.

For a practical application of state machines, consider the following technique for avoiding that the outputted parameter remains the same, implying that a fixed point has been reached. A mapping may yield the same output parameter value for different input feature vectors. Therefore, it may be more effective to monitor the mapping's output than its input, so the strategy will be that if π_n remains fixed for too long, then the map should be changed. In practice, we provide some tolerance for small deviations around the fixed point, so we coarse-grain π to a finite set of states $\{p_1, p_2, \ldots, p_N\}$. Suppose that we want to limit the number of repetitions of an output parameter that belongs to the same category p_i to at most L times. Using the notation $\wedge$ for the logical **and** operator, if

$$\bigwedge_{k=1}^{L} \pi_{n-k} \in p_i \tag{7.16}$$

holds for one and the same i, we have been repeating the output parameter $\pi \in p_i$ (within the margin of tolerance imposed by the coarse graining) for L time steps, and it is about time to modify the mapping function.

As an alternative to checking for fixed points, the rate of change of the features or the synthesis parameters could be monitored. Since the rate of change is the derivative $d\phi/dn$, or $d\pi/dn$, a subsisting close to zero derivative signifies either a slow change or a small fluctuation around a fixed point. In many cases, there is some spontaneous fluctuation or measurement noise in most feature extractors, even if the signal is perfectly stationary. Hence, the magnitude of the derivative can be used for detecting static behaviour.

The formalism of finite state machines is motivated by what can be done with functions in procedural programming languages (as opposed to mathematical functions). In the C language, as well as its descendants and relatives, functions are powerful constructs that can take other functions as arguments, they can call themselves recursively, and they can have internal state variables. The first two aspects, although extremely useful, can be ignored in this context. What is important here is that a program function can have *static variables*, which are variables that store an internal state of the function; hence, its output may be different for the same input if a state variable may influence its output. This is where the finite state machine concept becomes useful.

Repetition detectors can operate on different time scales and with various distance metrics. Furthermore, either the distance $d(\phi_n, \phi_{n-T_0})$ of a feature extractor to its delayed output, or the distance $d(\pi_n, \pi_{n-T_0})$ of the current and past synthesis parameters may be monitored. In any case, repetition detectors should operate with some delay T_0 of at least several milliseconds, or even on the order of several seconds. The motivation for this can be seen when considering a time-domain feature extractor. Locally, that is, on short time scales, the feature will not vary very much, so detecting repetition from one sample to the next is wasted effort. Again, the time scale of the repetition detector will have a huge impact on the general rate of change of the sound. It becomes a new parameter to tweak according to ones needs.

If the system has a tendency to approach a fixed point π^* (implying fixed stable oscillation and constant synthesis parameters), then it may be perturbed away from the fixed point after having spent a certain minimal time T_0 there. In this case, it is sufficient to measure the distance between the current parameter vector and one that is delayed. On the other hand, if the system does not occupy the fixed point over the duration T_0, but keeps returning to a recurrent parameter value π^*, then it is likely that this recurrence will also be detected and the system will be perturbed. Thus, a more sophisticated repetition detector, or more properly, a *stasis* detector, has to take account of all intervening time points in the time interval that is being monitored. Otherwise, it cannot distinguish between fixed points, repeating cycles and chaos.

7.1.9 Generating and detecting glissando

Non-stationarity in pitch can manifest itself as a glissando, or as periodic or irregular modulations. Ordinary differential equations may be used to generate such pitch contours. These pitch modulations may then be superimposed on slower, stepwise changing contours.

Glissandi as flows For an oscillator with instantaneous frequency f_t, a glissando can be made linear in frequency, which means that $f_t = f_0 + gt$ for an initial frequency f_0 and a glissando rate g, whereas for a glissando to be linear in pitch, the frequency variable has to ramp linearly in units of cents, semitones or octaves. Then, the instantaneous frequency is $f_t = f_0 2^{gt}$.

Flow-like mappings are well suited to produce glissandi. Linear frequency glissandi result for $\dot{f} = g$ (where g has units of Hz per second), whereas exponential glissandi are produced with $\dot{f} = gf$. The end-points (or boundary conditions, to use the terminology

proper to ODEs) are of particular concern, since a glissando that continues for too long will soon exhaust the musically useful frequency range.

A perfectly periodic vibrato satisfies $f_t = f_{t+T}$. Again, an ODE formulation may be used to generate vibrato, namely the harmonic oscillator $\ddot{f} = -\omega^2 f$, where ω is the radian frequency of the vibrato.

Now, one reason for introducing sliding pitch changes may be to create variation in a texture of otherwise stable pitches. In that case, another control operating at a slower rate is needed to switch between different amounts and directions of pitch change. A vibrato may be produced by other means than a second order ODE if the direction of glissando is regularly changed. However, the harmonic oscillator generates a sinusoidal vibrato, whereas a repeated switching of glissando direction only produces a triangular shaped vibrato. A general form of a flow-like mapping using $\dot{f} = g_n f$ or $\dot{f} = g_n$ is

$$g_{n+1} = m(g_n, f_n, \phi_n)$$

where ϕ is a pitch follower, and the map is updated at a slow control rate. Using this mapping to generate a vibrato, the modulation rate is given by the control rate if the glissando direction is swapped at each iteration of the map. The current estimated frequency is given as an argument to the mapping because this is useful to prevent excessively high or low pitches.

Other ODEs are interesting to use for frequency control as well, including some of those that were briefly mentioned in Chapter 4, such as nonlinear oscillators and ensembles of coupled oscillators. If the most direct sonification is to use the orbit directly as sound samples, perhaps the second most direct strategy involves frequency control of oscillators, using one oscillator for each variable of the system. An example where a flow is used for pitch control is described below (Section 7.2.1).

Detectors If the autonomous instrument is able to generate both a glissando and a stable pitch, then it will be useful to have a feature extractor that is able to distinguish between different pitch profiles. How can we detect the difference between a melodic profile that jumps between distinct pitches and one that slides continuously? A glissando detector should be able to distinguish profiles with stepwise pitch transitions from smooth glissandi. The pitch needs to be estimated at successive points in time. Two points are sufficient for discriminating between a steady tone and a transition, but a third point is needed for the distinction between sudden and smooth transitions.

Let $\hat{f}_n, \hat{f}_{n-\tau}, \hat{f}_{n-2\tau}$ be the estimated pitch at three points in time separated by a suitable time lag τ. Obviously, τ should be at least as long as the temporal support of the pitch follower. Hence, the glissando detector will be a part of a mapping of the form (7.13) which uses several past input values from the feature extractor. If all three pitch followers have monotonously increasing or decreasing values, then there is a glissando. If two consecutive values are equal and the third is different, then a sudden pitch transition has occurred. The differences $\delta_1 = \hat{f}_n - \hat{f}_{n-\tau}$ and $\delta_2 = \hat{f}_{n-\tau} - \hat{f}_{n-2\tau}$ are more useful than the pitch followers as such. If they are multiplied, then $\sigma = \delta_1\delta_2$ will be zero if any of the differences equals zero, and positive for either a rising or falling glissando. Two further alternatives remain: the contour may have a maximum or minimum somewhere between the two outermost pitch extractors. If so, the differences have opposite sign, and $\sigma < 0$.

The glissando detector should certainly be complemented with a basic pitch profile direction analyser. Such a feature extractor is similar to a differentiator, which can be as simple as taking the difference δ_1/τ as its value. Although we will not demonstrate the practical use of glissando detectors in actual feature-feedback systems, they provide yet another example of the need for considering the generalised mappings that were proposed above. However, in the case of a glissando detector, it may of course be regarded as a basic feature extractor. The division of labour between feature extractors and generalised mappings is somewhat arbitrary.

After these mostly abstract studies of non-stationarity, we now turn to specific implementations that use some of the techniques presented here.

7.2 Case studies

Two autonomous instruments will be developed in this section. First, the tremolo oscillator is reconsidered, then the discrete summation formula synthesis technique is put to use in a somewhat complicated instrument. These two instruments are described in some detail, because they were used for two of the three sections of the Autonomous Instrument Song Contest (see Section 7.3).

7.2.1 The tremolo oscillator revisited

The parameters and functioning of the tremolo oscillator were briefly described in Chapter 3. As argued there, despite potential aliasing problems, it is worthwhile to use a non-bandlimited waveform that allows for smooth transitions of the modulator waveform. Here we recall the ideas, and introduce feature extractors and mappings that will be used in a feature-feedback system. Four sound examples using this system were used in the Autonomous Instrument Song Contest.

The tremolo oscillator is a simple, yet flexible model which consists of an oscillator switching back and forth between two frequencies. Let F_1 and F_2 denote the two frequencies, and let t_1 and t_2 be the duration of each frequency. The frequency changes of the oscillator may be smoothed with a lowpass filter to produce more or less gliding effects. With extremely rapid alternations, the total period $T = t_1 + t_2$ may be short enough to produce a pitch percept of its own at $1/T$ Hz, where the two frequencies F_1 and F_2 act as formant regions. Slower alternations may produce various stream integration or segregation effects, depending on the frequency interval and the pace of the tone sequence (see Holopainen, 2010). Typical telephone call signals are well within the range of idiomatic expression of this instrument—at least from the era just before telephones began to sound like anything at all—but the timbral range is quite rich.

For the feature extractors, fundamental frequency ($\hat{f}$), voicing ($\hat{v}$) and spectral flux ($\hat{\Phi}$) will be used. All three features are measured with the same window length L.

In order to exhibit the most of the attainable variety of this model, some complexity will be required in the form of several control rates. The two frequency parameters F_1 and F_2 are updated at the block-rate f_s/L, that is, as often as the feature extractors are updated. The duration variables (t_1, t_2) are updated by another map at another rate f_s/P that may be chosen arbitrarily, although it does not make much sense to use rates

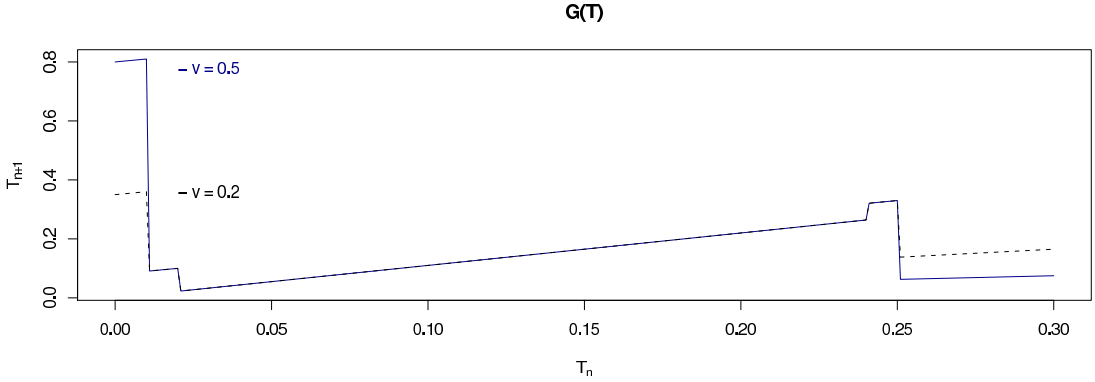

Figure 7.5: Mapping of period length T assuming that $t_1 = t_2$ for $\hat{v} = 0.2$ (dashed curve) and $\hat{v} = 0.5$ (solid).

that are faster than typical period lengths T. In addition to this, a flow-like mapping of the frequency variables will be used, which operates at the audio sample rate.

The mapping of time variables is a function $(t_1, t_2) = g(\hat{v}, t_1, t_2)$ which mainly depends on the period length T, and is split up into several cases depending on T. We refrain from listing the complete function here. Although the graph of g is not easily depicted, a simplified graph (Figure 7.5) showing how total period lengths are mapped for $\hat{v}$ fixed and $t_1 = t_2$ displays the essential behaviour of this function.

For the frequency variables, two different mappings are used interchangeably. First, there is the map

$$(F_1, F_2)_{n+1} = m(\hat{\Phi}, \hat{v}, \hat{f}, F_{2,n})$$

which is primarily given by

$$\begin{pmatrix} F_1 \\ F_2 \end{pmatrix}_{n+1} = \begin{pmatrix} \frac{1}{2}\hat{f} - 100\hat{v} + 800\hat{\Phi} + 200 \\ F_{2,n} + 3(1 - \hat{v} - \hat{\Phi}) \end{pmatrix} \tag{7.17}$$

but in addition, there are reflective boundary conditions that hinders both frequency variables from wandering into sub-audio or too high frequencies (row and column vectors are mixed without respect of mathematical etiquette here; there should be little risk of confusion anyway). The second map is more complicated by incorporating a memory of two past time steps of the frequency variables:

$$(F_1, F_2)_{n+1} = M(\hat{\Phi}, \hat{v}, \hat{f}, (F_1, F_2)_{n-1})$$

This map is given by

$$\begin{pmatrix} F_1 \\ F_2 \end{pmatrix}_{n+1} = \begin{pmatrix} \left|1.1\hat{f} - 0.7\gamma_2\right| + 450\hat{\Phi} \\ 15\sqrt{\gamma_1} + 250(\hat{v} - 0.02) \end{pmatrix} \tag{7.18}$$

in which

$$\gamma_i = k(\tanh(F_{i,n-1}/k - 1) + 1), \quad i = 1, 2$$

are delayed and warped frequency variables, and $k = 5000$ Hz.

Lastly, there is the flow-like sample rate mapping of the frequency variables, which introduces glissandi or other types of modulation. Similar to the Lotka-Volterra model of population dynamics in a system of predators and prey (e.g. Strogatz, 1994), the mapping

$$\begin{pmatrix} \dot{F}_1 \\ \dot{F}_2 \end{pmatrix} = h \begin{pmatrix} aF_1 - bF_1F_2 + \xi_1(\hat{\Phi}) \\ bF_1F_2 - cF2 + \xi_2(\hat{\Phi}) \end{pmatrix} \tag{7.19}$$

lets the frequency variables hold each other in check. Here $h \geq 0$ is an overall scaling parameter, $\xi_i(\hat{\Phi})$ are two different linear functions of spectral flux, both taking very small values, and a, b, c are small positive parameters. If this were an ecological model, then the ξ functions might be thought of as some kind of environmental effects that change stepwise, but very infrequently.

At this point it is warranted to ask whether not simpler functions can be found that could replace these rather complicated mappings. The functions listed here were found by trial and error; a more extensive search would be needed in order to optimise the simplicity of the functions without compromising the level of sonic variation attained with the present mappings.

The tremolo oscillator has two parameter states F_1 and F_2 that it regularly switches between. Idiomatic use of the tremolo oscillator includes audio rate switching as well as slower tremolo as in ornamental trills between two notes. With only two states to alternate between, the tremolo oscillator does not require much memory, although it can easily be extended to store a larger number of parameter states. Then, it would be similar to the step sequencer introduced above in Section 7.1.2.

Furthermore, the variety of behaviour can be increased by introducing "presets", or readymade parameter vectors to be selected by a mapping function. A set of such parameter vectors are stored in an array, and one of the array elements is retrieved by a selection function. Slow-paced variation can then be obtained by updating the array elements sporadically.

7.2.2 Mapping with iterated functions

The following is a presentation of the components of another instrument used in the Autonomous Instrument Song Contest (Section 7.3 below). The instrument is a bit complicated, because it includes many components that operates at different rates. Therefore, its constituent parts are introduced one at a time. It should be noted that all its separate parts have general applicability.

Consider a feature-feedback system with block-rate feature extraction and a mapping of the form $\pi_{n+1} = f(\pi_n, \mu_n)$, where π is a parameter vector as usual and μ, which comes from the feature extractor, is used as a slowly varying parameter of the map f. In the limit where μ is constant (or does not affect the dynamics of f in any way), the system becomes a signal generator driven by an external map. This idea is similar to an application of chaotic maps suggested by Truax (1990): an iterated map used at a control rate yields the frequency of an oscillator.

Loosely inspired from maps suggested by Pressing (1988), the particular map we will use is a variant of the logistic map with reflecting boundaries. For the mapping function, we define it in two steps. First, the function

$$f(x, \mu) = \mu\sqrt{x}(1 - \sqrt{x}) \tag{7.20}$$

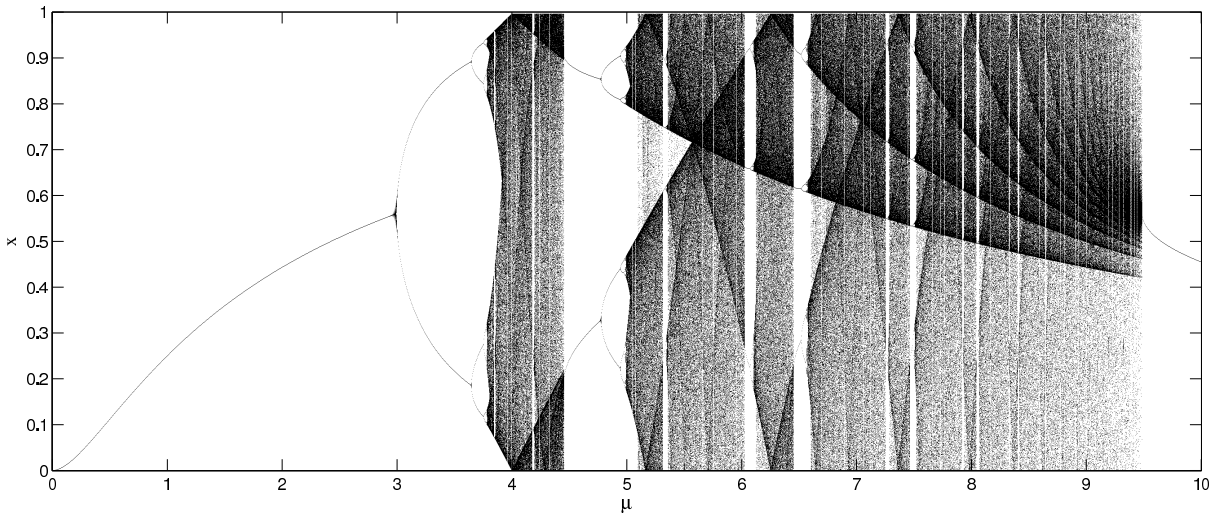

Figure 7.6: Bifurcation plot of the compound mapping $T(x, \mu)$ for $\mu \in [0, 10]$.

can be recognised as a modification of the logistic map, although its hump is more skewed to the left. Then, the mapping

$$T(x,\mu) = \begin{cases} f(x,\mu) & \text{if } f(x) \leq 1 \\ 1/f(x,\mu) & \text{if } f(x) > 1 \end{cases} \tag{7.21}$$

using (7.20) and defined for $x \geq 0$ folds over any values $f(x) > 1$ to the interval $[0, 1]$ with a kind of reflective boundary. With this safety net in place, μ may take any positive values. The bifurcation plot of (7.21) can be seen in Figure 7.6. The map T undergoes a period doubling cascade qualitatively similar to the logistic map, with period two beginning at $\mu = 3.0$ and the onset of chaos roughly at $\mu = 3.8$. At $\mu = 4$, the effect of the folding in (7.21) sets in since $f(x) > 1$ for some x. Up to about $\mu = 9.5$, the map alternates between periodic windows and chaotic bands, but above that point the only stable orbit is a period one solution. Thus, if one would like to restrict the map to ranges where the period is two or higher, its parameter may be restricted to $3.0 < \mu < 9.5$.

The signal generator that will be used for this example is the discrete summation formula oscillator (described in Section 3.3.6). Recall that it takes three parameters with similar names and functions as in FM synthesis: a carrier frequency f_c, a modulator frequency f_m, and a modulation index $I \in (0, 1)$. The oscillator will be notated as $\mathcal{D}(f_c, f_m, I)$.

The iterated map (7.21) is used to drive the oscillator's parameters, but since it takes values in the interval $[0, 1]$ another mapping is needed to adjust the range to suitable intervals for each synthesis parameter. The mapping is a function $m : [0, 1] \to (f_c, f_m, I)$, although this would make all three parameters mutually dependent. It may be useful to allow these parameters to vary independently.

7.2.3 Phrasing by adaptive gain

All feature-feedback systems that have been introduced thus far share a general trait: they all produce uninterrupted streams of sound as long as they do not malfunction. They are, so to speak, left on like a radio receiver that one hopes would broadcast something interesting during the time it stays on. This is very different from the more common use of note events that trigger sounds, typically of relatively short duration, where a piece of music is assembled from sequences of such discrete events. Here, the algorithm runs without any notion of note events or other temporal segmentation (the tremolo oscillator considered above actually is an exception, although it too tends to produce continuous streams of sound). If we want it to stop temporarily, catch its breath as it were, then, either we may search for such dynamics to occur spontaneously—something that appears unlikely—or we have to design a phrasing mechanism.

A simple scheme can be used to good effect if the continuous stream of sound needs to be interrupted once in a while. The idea is to monitor the amplitude A_m at instants $m = 0, 1, \ldots$ separated by some window length L of samples; then, delay the measured amplitude and use it to calculate the current signal gain. Hence, we have the measured output amplitude $A_m = \mathrm{RMS}(gx_n)$ and a gain function $g = f(A_m)$. A sigmoid gain function can be practical, because it will act like an on-off switch; when the past amplitude was loud it turns off, and vice versa.

The sigmoid function may take the appearance

$$f(x) = (\tanh(K(x - \beta)) + 1)/2, \tag{7.22}$$

with parameters $K \gg 1$ controlling the steepness of the transition region and $0 < \beta < 1$ is an offset that allows tuning how the amplitude range will be affected. Using a delay of D steps, the gain function becomes

$$g_m = 1 - f(A_{m-D}) \tag{7.23}$$

for a delay length of $D \times L$ samples, where L is the length and hop size of the amplitude follower. Figure 7.7 shows the sigmoid function (7.22) as well as the effect it has on the delayed gain signal (7.23).

The gain function (7.23) may introduce discontinuous jumps in the gain, so it can be advisable to insert a smoothing lowpass filter on the gain signal. For a one-pole filter with a coefficient $0 < b < 1$, the smoothed gain

$$G_n \;=\; bg_m + (1 - b)G_{n-1} \tag{7.24}$$

is used to multiply the output of the generator $\mathcal{D}$ in the discrete summation formula system whose components we are gradually introducing.

It may appear impossible to get other results than a perfectly periodic on-off switching with this method. This is not quite true if the signal generator causes fluctuating amplitude on its own. The adaptive gain can add extra amplitude modulation if the effective delay length is short or it may introduce a kind of phrasing when using longer delays. Although the phrasing boundaries might not align with positions where it would

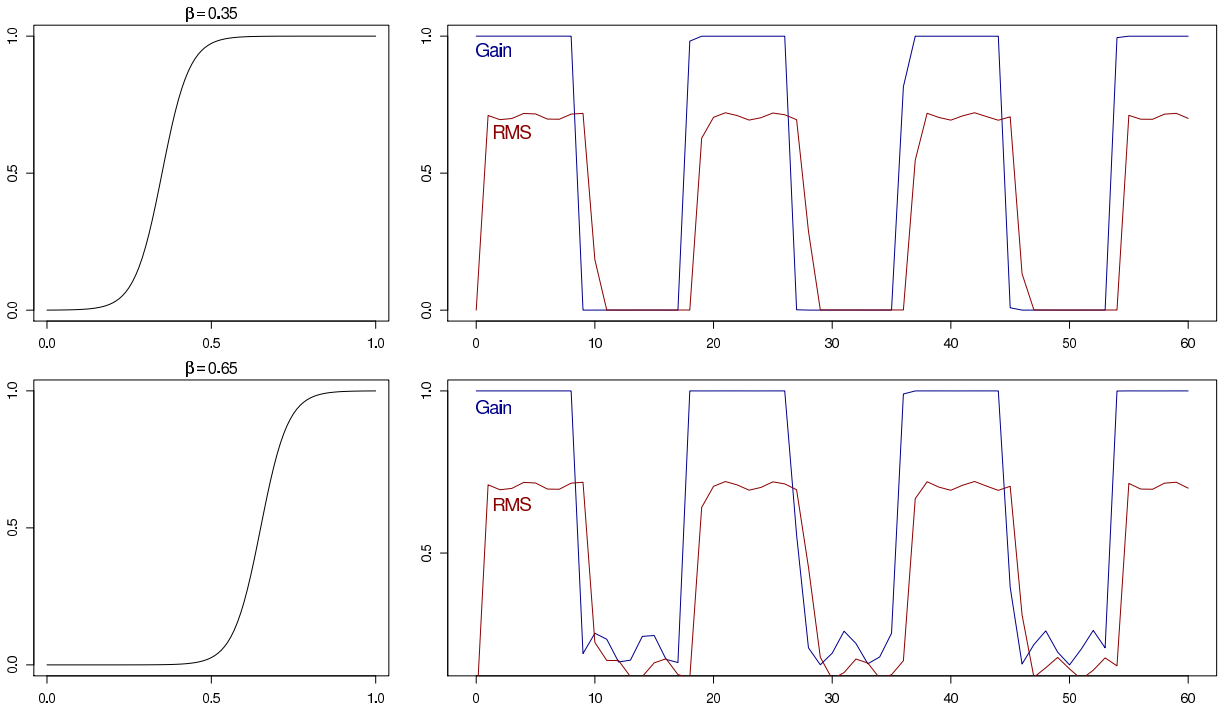

Figure 7.7: Adaptive gain with memory nine steps back. The left panels show the sigmoid function for $\beta = 0.35$ (top) and $\beta = 0.65$ (bottom). The measured RMS amplitudes of full-scale sinusoids and the resulting gain using the corresponding sigmoid functions is shown on the right. The bottom panel illustrates that setting β too high will cause the gain not to reach zero.

otherwise be sensible to break the musical flow, turning the gain down for a brief moment will induce an unquestionable segmentation.

If there are two different signal sources, this scheme can easily be modified so as to accommodate for a turn-taking, such that when one oscillator plays, the other is muted. A similar idea was proposed by Eldridge (2008), where several sound sources are mixed by controlling their gain with an N-species Lotka-Volterra model (see also above, Section 7.2.1).

7.2.4 The discrete summation formula system

Without getting into each and every detail, here is an outline of the functional structure in the feature-feedback system that has been used for some of the sound examples in the Autonomous Instrument Song Contest. The generator is the discrete summation formula, $u_n = \mathcal{D}(f_c, f_m, I)$. This output is amplitude modulated by the adaptive gain function (7.24) to yield $x_n = G_n u_n$ using a delay of four time steps (see eqs. 7.22 and 7.23 above). The output x_n is written to a sound file, and is also sent to feature extractors for estimations of voicing $\hat{v}$, fundamental frequency $\hat{f}$, centroid $\hat{c}$, amplitude $\hat{a}$, and flux $\hat{\Phi}$. All feature extractors operate on a block rate f_s/L. In some cases, the feature from the previous analysis frame is used for comparisons. Thus, the feature extractors and all variables that depend directly on them are updated each L samples. Another periodicity

P is used for updating the generator parameters $\pi_n = \{f_c, f_m, I\}_n$. Then, once every P seconds, a mapping

$$\pi_{n+1} = \mathcal{M}(\hat{v}, \hat{f}, \hat{c}, z_n) \tag{7.25}$$

is applied, where z_n comes from an iterated map. Specifically, the chaotic one-dimensional map

$$z_{n+1} = T(z_n, \mu) \tag{7.26}$$

is used to drive the system, which is the map (7.21), but now its control parameter is in turn a function of the latest flux value, $\mu = g(\hat{\Phi})$.

The idea is that the map T (7.26) is iterated more frequently than its bifurcation parameter μ changes. Then, several iterates of T will use the same bifurcation parameter and its behaviour (periodic or chaotic) will stay the same for all those iterates, until μ is updated. Hence, there may be long stretches of time when the map T is periodic, after which it may receive a new parameter value pushing it into either chaotic behaviour or perhaps another periodicity. However, it is possible to set the time constants P and L such that the map T will receive a new parameter value μ for each iteration.

Despite already being quite complicated, it turns out that this system as described so far is prone to find an equilibrium state. Therefore, a secondary mapping was introduced, which senses whether the distance between the current and previous feature vectors is smaller than some fixed limit ϵ. Thus, a repetition detector

$$d(\hat{\phi}_n, \hat{\phi}_{n-1}) < \epsilon \tag{7.27}$$

is used, where $\hat{\phi}$ consists of $\hat{v}$ and $\hat{f}$. When the expression (7.27) is true, a close recurrence has been found and the secondary mapping is applied. The number of such close recurrences as a function of the feature extractor's window length is indicated below in Table 7.2 on page 300 in the column labelled "resettings". The mapping which is then applied is a function of the form

$$\pi_{n+1} = m(\hat{v}, z_n, I_n). \tag{7.28}$$

Finally, it was found that sending the parameter vector as such to the discrete summation formula generator tended to introduce clicks and other rough edges. That problem was solved by lowpass filtering the parameters, so the generator actually takes a smoothed version of the parameters. The main components of this system are shown in Figure 7.8. Even without giving away all the details of the mapping functions used (eqs. 7.25 and 7.28), the structure of this instrument is quite complex.

After this brief exposition of the tremolo oscillator and the discrete summation formula system, we are ready for an evaluation of these two systems together with the most interesting system developed in the previous chapter, the wave terrain model.

7.3 Evaluations of complexity and preferences

We have argued in Chapter 5 that some kind of complexity should be the most natural criterion to use in the evaluation of autonomous instruments. Moreover, there are uses

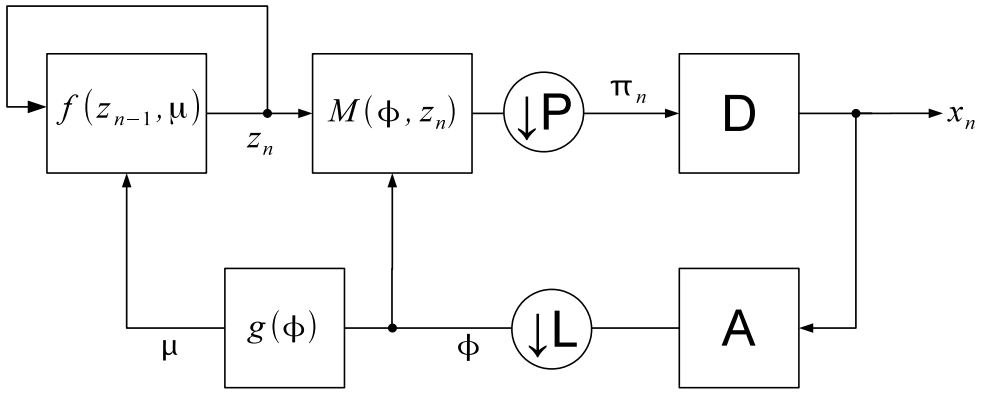

Figure 7.8: Rough outline of the functional structure of the discrete summation formula system, omitting several components. D is the signal generator using the discrete summation formula and A are the collected feature extractors that analyse blocks of L samples. The synthesis parameters are updated every P samples. If P and L are incommensurate, two competing modulation rates influence the dynamics.

for both informal, subjective evaluations of the perceived complexity as well as automated objective measures insofar as they are related to perceived complexity. In case the arguments in favour of complexity-related evaluations should not have convinced the reader, it must be admitted that there are other possible criteria for the success of an autonomous instrument. For instance, if its sound is reminiscent of some other music one likes, a factor of familiarity may affect judgments favourably. We will now concentrate on the complexity of the resulting sound files rather than considering the Kolmogorov complexity or the number of lines of code needed to generate the sounds. In fact, the same autonomous instrument may produce simple repeating patterns for some parameter values and complex irregular patterns at others.

As Streich (2006) has proposed, the complexity evaluations of music may adequately be divided into several facets. Among the facets introduced by Streich, those that appear to be best suited to characterise our feature-feedback systems are the timbral and rhythmic complexities. For timbral complexity, we will introduce a simple measure which captures the degree of variability among a set of feature extractors; this measure will then be compared to complexity evaluations from an informal listening test which will serve as ground truth. The listening test (the already mentioned Autonomous Instrument Song Contest) also asked about preferences and gave the participants plenty of opportunity to comment on each sound example. It should be emphasised that this study was of an exploratory character, thus the quantitative results should be seen as indications in need of further confirmation. However, the sound examples apparently evoked rather vivid imagery among some test participants, which will be related below in Section 7.3.7.

7.3.1 Quantifiable complexity

If we turn to the output signal as generated by an autonomous instrument, it may be analysed by a collection of feature extractors. For feature-feedback systems in particular, this is an eminently well motivated method, since the system's internal state also depends on feature extractors. Then, one could simply use the internal feature extractors of the system for a description of its dynamics. However, if one would like to compare the dynamics across several different feature-feedback systems, then the same set of descriptors must be used, and it will be better to analyse the instrument's output signal.

The multidimensional time series of descriptors can then be further studied with respect to fractal dimension, entropy and other measures. This approach, although not using the available knowledge of the instrument's internal dynamics, nonetheless allows for a direct comparison between different parameter settings as well as across different autonomous instruments. Indeed, the analysis of feature vectors is the most general approach, applicable to any style of music and any type of audio signal.

Analysis using feature vectors also allows comparisons to be made with user evaluations of complexity. At this stage we can test if there are correlations between user preferences, the experienced level of complexity and objective complexity measures. The results of such a comparison will be reported below.

As discussed in Chapter 5, Shannon entropy has often been related to complexity, but it has also been criticised as an inadequate complexity measure. After all, maximum entropy corresponds to equiprobability of all possible events, whereas objects that have a high perceived complexity tend to have a medium entropy. Shannon entropy is originally defined on discrete sets of symbols such as the letters of the alphabet or the different pitch classes in music. Real valued feature extractors cause problems here: in order to define the probability of a feature value being observed in an interval, a partition must be imposed on the range of the feature extractor. The problem is that the entropy estimates depend critically on the partition. Apart from Shannon entropy, many other entropy measures have been proposed, some of which might be used in similar situations. One of them is the Kolmogorov-Sinai entropy, which is defined as the supremum of entropy taken over all conceivable partitions (Ebeling et al., 2001). The permutation entropy introduced by Bandt and Pompe (2002) does not require arbitrary partitions to be made, and would be interesting to apply at the level of feature extractors. The measures of structural change on multiple time scales by Mauch and Levy (2011) also seem promising for a quantification of complexity, but we have to leave them for future studies. Several potentially useful complexity facets were also introduced by Streich (2006). The essential idea worth retaining is to split the perceptual complexity evaluation into several dimensions. A monolithic notion of complexity is hardly justified when applied to perception. Therefore, in the Autonomous Instrument Song Contest, we have asked the test participants to judge two facets of perceptual complexity related to timbre and rhythm.

It would have been nice to have a really simple objective complexity measure that nonetheless faithfully reflected the perceived complexity of music. This is probably not a realistic hope, as evidenced in most of the above mentioned methods as well as the refined method of rhythmic complexity estimations by Shmulevich and Povel (2000), discussed

in Section 5.2.7. In light of these attempts, we now propose a simple, albeit rather naïve method for diagnosing timbral complexity.

7.3.2 Spread of feature extractors

The range of a feature extractor as it varies over time indicates how much the sound changes. We introduce a measure of the spread of a collection of feature extractors during a time segment (spread of features, or *SOF* for short) as follows: Outliers of the distribution will not be counted, because we would like to consider typical behaviour rather than extreme events. If the evolution of the sound is characterised by an initial transient followed by more static behaviour, most of the outliers are likely to be found in the beginning. Therefore, the measured range of each feature F_j is taken as the α and $(1-\alpha)$-quantiles of its distribution, written as $Q_\alpha(F_j)$ and so on. With $\alpha = 0.025$, the lowest and highest 2.5% of the distribution will not count; this is the α-value that will be used in the following. The measure of spread

$$SOF_\alpha = \frac{1}{N}\sum_{j=1}^{N} w_j \left(Q_{1-\alpha}(F_j) - Q_\alpha(F_j)\right) \tag{7.29}$$

is just the average spread over all the feature extractors weighted by coefficients w. It appears to be a bad idea to use standard deviation as the measure of spread, because the feature values may have irregular distributions, often they even deviate from single mode distributions. Quantiles are a more robust statistic in that respect.

The feature extractors that will be used are: ZCR, voicing, spectral entropy, inverse crest factor, flux, and centroid. (Recall from Section 2.3.1 that the crest factor is the peak amplitude divided by RMS amplitude. Taking its inverse ensures that it is normalised to the unit interval.) These features are already normalised to $[0, 1]$, so the weights will all be set to $w_j = 1$. The purpose of the weights is to set the balance if any feature with a different range should be added, or if one wants to emphasise certain features over others.

Variations in pitch or loudness contribute to the general amount of variation in a sound. Then, amplitude and fundamental frequency extractors might be used to complement the other feature extractors in the SOF measure. If so, the frequency is scaled to the interval $[0, 1]$ by a suitable choice of its weighting coefficient in order to have the same range as the other dimensions. This measure including pitch and amplitude will be called SOF+.

Using one and the same window length, hop factor and length of sound files to analyse, the relative amount of timbral variation can be estimated across different sound files. That is what we will do with the twelve sound examples in the Autonomous Instrument Song Contest. For reference, the SOF and SOF+ measures of a few signals are given in Table 7.1. The SOF+ measure is usually smaller than SOF because the measure is scaled by the number of feature extractors. A constant sinusoid results in close to zero values. The chirp signal spans the entire range of ZCR and centroid, which explains why it takes relatively high values whether or not the frequency extractor is included. Steve Reich's Pendulum Music (discussed in Chapter 5) is a good example of a piece that begins with

	Sinusoid	Pink noise	Chirp	S.709	Reich (a)	Reich (b)	Speech
SOF+	< 0.001	0.037	0.316	0.292	0.364	0.255	0.354
SOF	< 0.001	0.041	0.350	0.338	0.419	0.288	0.417

Table 7.1: SOF values of some one minute signals with $\alpha = 0.025$, feature extractor window length 8192 samples. The chirp signal is a linear, constant amplitude chirp from 20 to 0 kHz over the entire duration. S.709 is the first minute of Xenakis' piece; Reich (a) and (b) refers to the first and the last minutes respectively of Reich's *Pendulum Music*, performed by Sonic Youth. The upper row uses amplitude and frequency extractors, whereas the lower row does not.

a high degree of timbral variation, but ends when it has reached stasis. The decrease in spread of features from the first to the last minute of the piece confirms this impression.

The SOF measure tries to indicate the amount of timbral variety in the sound. There are other plausible ways to measure timbral variety (Streich, 2006), but SOF has the benefit of being easy to calculate. The underlying assumption is that feature dimensions are independent and that it makes sense to add them together. Alternatively, the collection of features $F = \cup_{j=1}^{N} F_j$ could be treated as points in an N-dimensional space, and the spread could be taken to be the greatest distance between those points.

If timbre is regarded as an aspect of sound independent of loudness and pitch, then it is arguably more correct to measure plain SOF without using RMS amplitude and fundamental frequency extractors. However, the SOF measures with or without amplitude and frequency are correlated, although using fewer dimensions tends to yield higher values. Henceforth SOF will be calculated *without* fundamental frequency and RMS amplitude extractors. Zero-amplitude analysis frames must however be skipped lest the SOF measure be overestimated. Silence, after all, has no timbre. This has been done in the analyses, but for signals containing much quiet but not entirely silent parts such as speech, it may be better to set a positive amplitude threshold for which analysis frames to include.

7.3.3 The Autonomous Instrument Song Contest

An internet survey was designed where participants were invited to listen to several sound examples made with feature-feedback systems. Questions were asked about positive and negative preferences as well as the perceived complexity and simplicity of the sound examples. The results were compared with the SOF measure of each sound example in order to test its validity.

When working with synthesis models and with complicated autonomous instruments in particular, there is a risk of acquiring a myopic view of the sounds they produce. At that point it can be healthy to get a second opinion about the sounds. Are they interesting or dull, pleasant or ugly, simple or complex? Descriptions of each sound example were given by several participants. Some of these descriptions are cited in Section 7.3.5; other responses will be further interpreted in the next chapter.

This is an exploratory study, where the respondent's free associations to the sounds are no less interesting than their evaluations of preference and complexity. Unlike studies in

music psychology, this survey did not primarily address the way judgements of complexity and preference are correlated in the population; rather one of its purposes was to gather ground truth about the perceived complexity of various sounds generated by autonomous instruments. Due to the way data were collected (see below), responses are not available in a convenient format for studies of the correlation between preference and complexity ratings on an individual basis.

From the collected data, rankings of preference and complexity were obtained for the sound examples. Such rankings assume that the respondents agree to a certain degree about what is complex and what is not. There are indications that there is a sufficient agreement about complexity and simplicity across the respondents, as will be discussed below in Section 7.3.6. The ranking of timbral complexity will then be used as ground truth in comparisons with the SOF measure.

Designing sound examples with feature-feedback systems so that they vary systematically in a single perceptual dimension while other dimensions are kept neutral is virtually impossible. For each of the three test sections, some synthesis parameters or even mapping functions were varied across the examples. However, parameter variations do not yield easily controllable changes in perceptual aspects of the sounds. Hence, one example may be perceived as more complex than another with respect to several facets of complexity; likewise, there may be several reasons for preferring one example rather than another. That is yet another way this study differs from reasonably well designed psychological experiments; the stimuli are not easily controllable and vary in several ways at once, thus making it hard to draw conclusions as to what aspect of the stimuli were likely to yield high or low perceptual complexity. Nevertheless, the participants provided valuable responses that will be analysed in the rest of this section as well as in the next chapter.

7.3.4 Method

Participants The test was carried out as an anonymous internet questionnaire. Invitations with links to the test were sent out to the cec-conference and music-dsp mailing lists, to the staff at the Department of Musicology at the University of Oslo, and to several other contacts. Typical subscribers to the mailing lists can be expected to be well versed in electronic music, but the test was open to any participant regardless of background. A total number of 35 replies came in, of which five replies were empty. Furthermore, two pairs of replies were obviously duplicates of the same reply at different stages of completion. This makes for a total of 28 usable replies.

No questions were asked about the participant's musical background or previous experience in electronic music. However, in many cases one can get an idea of each respondent's level of relevant background knowledge from their qualitative assessments of the sound examples. Even if not asked directly about it, some participants described what they assumed to be the underlying sound generating mechanism; moreover, some of these descriptions were remarkably accurate despite that virtually no information was given about the synthesis models. However, it cannot be ruled out that curious participants might have found related information by searching the internet for a term such as "autonomous instruments". Such technical descriptions will be further commented on in the

final chapter (see Section 8.1).

Unfortunately, some potential respondents reported having trouble with the media players on their web browsers, which may explain the relatively low number of participants. Another reason for the small number of replies may be that the questionnaire was only open for two weeks in the end of April 2011, including the Easter holiday.

Questionnaire design A readymade solution was chosen for the questionnaire based on a form developed and used at the University of Oslo. In the introduction to the test there was a "warm up" sound file. Participants were urged to play it and adjust the volume to a comfortable level. This test sound file also served the purpose of introducing examples of the sounds to be encountered later in the test. It consisted of three segments of more or less typical sounds from the three different synthesis models that were used in the test, though not identical with any of the examples used. The three segments of the test sound file were spliced together to a duration of 32 seconds. All sound examples were encoded as 48 kHz, 320 kbps mono mp3 files. Each sound example in the main part of the questionnaire had a duration of 40 seconds.

The test had three sections (A, B, C) with four sound examples each. Under each sound example there was some empty space for comments, and the following instruction was given: "As you listen to the following sound examples (in any order), you may write brief comments as an aid to your memory." Although the questions were given in English, the questionnaire encouraged answers also in French or the Scandinavian languages.

Each section ended with five forced choice questions, where four radio buttons appeared whose alternatives were the sound examples of the present section. In fact, it was possible to skip these questions since there were no mandatory questions, although participants were urged to try to answer as many questions as possible.

The following questions were posed:

1. Which of these sound examples do you prefer? Vote for your favourite.

2. Which sound example did you like least? This counts as a negative vote.

3. Which sound appears to be most rhythmically complex? This may mean most unpredictable, or difficult to tap along to.

4. Which sound example is timbrally most complex? That is, which one is most varied in timbre?

5. Which of these sound examples is simplest? Mark the one that seems most redundant or predictable.

At the end of the test participants were asked to reflect on all of the sound examples and questions in the test. The following forced choice questions were posed:

1. How hard was it to decide which examples you liked most? Was it easy or difficult to pick your favourite sound in the three sections above?

2. How difficult was it to decide which example was most complex? Were the questions about timbral and rhythmical complexity easy or hard to answer in general?

The alternatives were: No problem at all; Quite easy; Somewhat difficult; and Almost impossible.

Then some open questions were posed about the test as a whole:

1. Did you have an overall favourite sound? Which one, and why?

2. Surprise. Were you ever surprised by any of the sound examples? Which one, and why?

3. Are there any similarities? Think about all the sound examples you have listened to. Is there any general similarity between them? If so, how would you describe it?

4. Odd examples. Are there sound examples that stand out as different from the others in a section? If so, which ones? In what sense are they different?

5. Other comments?

7.3.5 Description of sound examples

In the design phase of this study, the sound examples were adjusted in amplitude so as to avoid great loudness differences between sections. Within sections, the loudness was judged not to vary too much. After a preliminary test, it was noted that the sound examples in group B (from the wave terrain model) had a much brighter timbre than the others, making them somewhat unpleasant to listen to. Therefore, some lowpass filtering was applied to all the examples in section B.

Section A (see Table 7.2) used the discrete summation formula instrument developed in this chapter (Section 7.2.4). Only the feature extractor window length parameter was varied. For the two shortest window lengths (512 and 1024 samples or 10.7 and 21.3 ms, used in A3 and A1, respectively), a grainy texture emerges with occasional tonal bleeps. With longer windows, quite different patterns result. There are almost repeating note patterns, where the rhythmic pattern is less varied and the pitches and timbral qualities change more. Examples A2 and A4 have relatively long silences between sounds.

In section B, the wave terrain instrument from Chapter 6 was used. It had the same parameter constants as given there (eq. 6.57 on page 261), except that the two parameters υ and η were also varied. The parameters were as given in Table 7.3.

Both examples B1 and B3 are irregular (this irregularity persists for at least 10 minutes). Example B2 starts with a periodic short pattern that is repeated with subtle variations for the first 52 seconds (from time zero, which is not included in the sound file). The sound file begins 45 seconds into this process, while it is still periodic. Then, a wildly irregular phase takes over, after which the sound enters the decaying vibrato phase (cf. Figure 6.23 on page 264 and the left part of Figure 7.1 on page 270).

Section C (Table 7.4) uses the tremolo oscillator and two independent mappings for the pitch-related variables ($F_{1,2}$) and the duration-related variables ($t_{1,2}$) as described above in Section 7.2.1. The internal feature extractors in the tremolo oscillator (flux, voicing and fundamental frequency) are used with one window length L, and another time constant P is used for periodic resetting of the duration variables. Incommensurate

Sound example	Window length (ms)	Resettings	SOF	Comments
A1	21.3	423	0.277	Fast pulse, slightly irregular, slightly noisy
A2	85.3	0	0.382	Varied groups of 3-5 tones, fast
A3	10.7	220	0.217	Bubbling granular texture, quite noisy
A4	341.3	41	0.457	Similar to A2, but slower

Table 7.2: Sound examples in section A using the discrete summation formula system. Resettings refers to the number of times the repetition detector (eq. 7.27) found a close recurrence.

Sound example	L_a (ms)	L_c (ms)	h	υ	η	SOF	Comments
B1	500	309	4.5	0.7	0.8	0.372	Irregular
B2	400	400	9.5	0.7	0.8	0.428	Starting from 0'45
B3	130	210	9.5	0.8	0.7	0.240	Irregular
B4	650	80	4.0	0.7	0.7	0.322	Almost periodic

Table 7.3: Sound examples section B with the wave-terrain instrument. Ex. B2 was used from 45 seconds into the sound.

lengths of L and P ensure that there will be no exactly repeating periods. Two different mappings were used for the frequency variables.

Example C1 soon approaches a repeating pattern after an initial transient has settled. The periodicity arises in part because of the window lengths since in this case P was set to exactly three times as long as L. C2 and C3 use the m map (7.17), whereas C1 and C4 use the M map (7.18) and a second mapping (flow), which runs at the audio sample rate and is a kind of Lotka-Volterra system applied to the frequency variables, as described above.

7.3.6 Results

As mentioned, a total of 28 different answers came in. The preference and complexity ratings are shown in Figure 7.9. Since participants only rated their most preferred example within each of the three sections, one cannot compare the results directly across sections. Apart from that, it is easy to see that some ratings were more unanimous than others.

One may wonder how reliable the respondent's evaluations are. As a way to estimate their confidence, they were asked how easy or hard it was to decide on their favourite sound and the most complex sounds. Slightly more than half of the test subjects found

Sound example	L (ms)	P (ms)	map	SOF	Comments
C1	341.3	1,024	M + flow	0.265	Periodic, triple metre
C2	10.7	1,200	m	0.244	Timbrally varied, rising glissando
C3	341.3	1,200	m	0.341	Cycles of tremolo rates speeding up
C4	10.7	1,200	M + flow	0.168	Periodic, quadruple metre

Table 7.4: Section C uses the tremolo oscillator. Window lengths in milliseconds. Different mappings were used. Note: in C1, $P = 3L$ which causes periodicity.

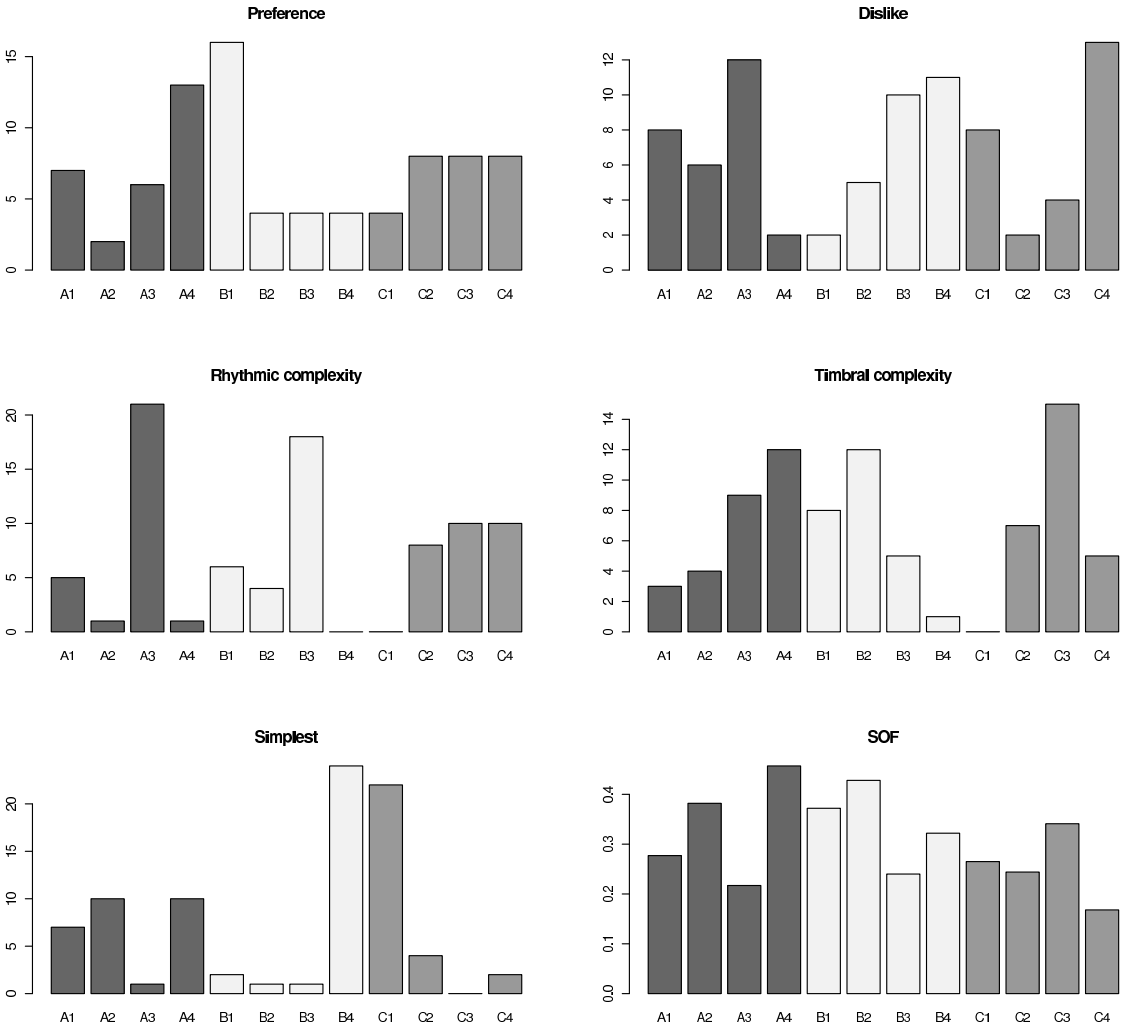

Figure 7.9: Ratings (from top left to bottom right) of positive and negative preference, rhythmical and timbral complexity, simplicity and the SOF measure. Notice that the ratings are always made within each section, hence one cannot directly compare the number of votes across sections A, B, and C.

Rating confidence	No problem	Quite easy	Somewhat difficult	Almost impossible
Preference	1	12	13	2
Complexity	1	5	20	2

Table 7.5: Rating confidence (self evaluated). Complexity appears to be more difficult to assess than preference.

it hard to rate their preferences (15 against 13), whereas the complexity ratings were deemed more difficult by a majority of respondents (22 against 6; compare Table 7.5).

Another way to estimate to what degree respondents agreed on preference and complexity judgements is to study the correlations between the positive and negative ratings. Thus, the correlation between preference and dislikings gives an expected negative correlation of $r = -0.529$, with statistics $df = 10$ (note that the degrees of freedom is given by the number of sound examples, which is twelve), $p = 0.0767$, which is not significant at the $p = 0.05$ level. So, although there is a negative correlation of liking and disliking as one might expect, it is not highly significant. This indicates that the group as a whole differed somewhat in their preferences.

For the complexity evaluation, it is reasonable to merge the ratings of timbral and rhythmic complexity by adding the variables. Then, the combined complexity ratings are negatively correlated to simplicity with $r = -0.842$ ($p = 0.0006$). Thus, there is good mutual agreement on the complexity evaluations. Timbral complexity alone versus simplicity, as well as rhythmic complexity versus simplicity, are correlated with $r = -0.67$ and $r = -0.68$ respectively, both significant ($p < 0.05$).

The questionnaire gave the instruction that the example you disliked most would count as a negative vote. Hence, we combine positive preference and disliking ratings into a single preference variable, and likewise add up the two complexity facets and subtract the simplicity ratings:

$$\begin{aligned} P &= \text{preference} - \text{disliking} \\ C &= \text{timbral} + \text{rhythmic complexity} - \text{simplicity} \end{aligned} \tag{7.30}$$

It turns out that there is no linear correlation between the two variables, P and C ($r = 0.185$, $p = 0.564$). However, as previous studies of the relation between aesthetic preference and complexity have shown (Heyduk, 1975), there is often an inverted U curve with a peak preference for stimuli that are moderately complex. Since we have not studied the correlations for each individual, a test of this hypothesis cannot be performed on the data of this study without manual recoding. What can be shown presently, however, is the corresponding scatter plot as it looks from the perspective of individual sound examples. The result shown in Figure 7.10 must thus be interpreted with some caution; in particular, it is not warranted to draw any conclusions about possible relations of complexity and liking in the test population. As can be seen, the lower and upper extremes of complexity (examples B4 and C1 on the low end of complexity, example A3 on the high end) receive low scores of combined preference, whereas some examples with medium complexity get the highest preference scores (ex. A4 and B1). However, it would be an exaggeration

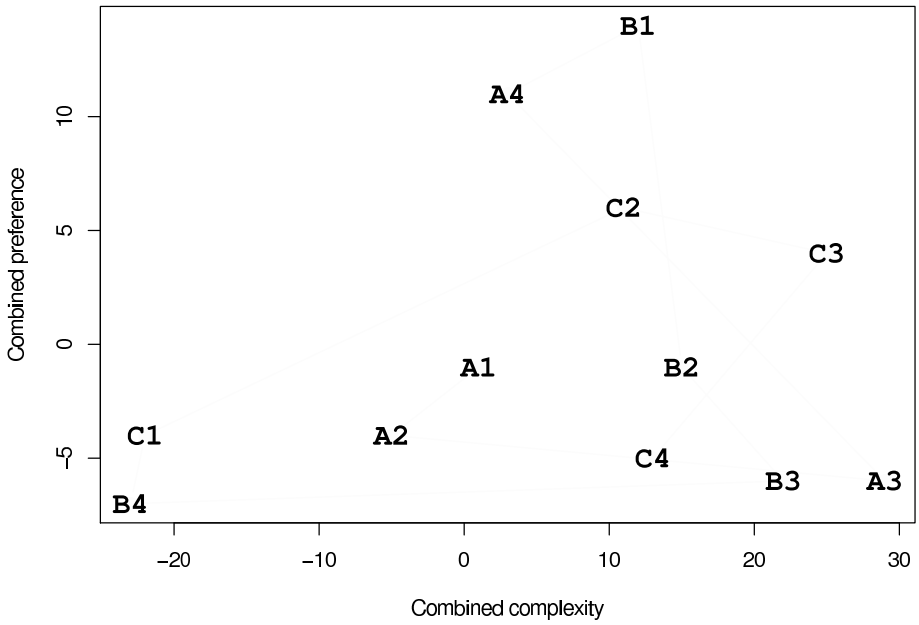

Figure 7.10: Combined complexity evaluations versus combined preferences with dislikings subtracted (eq. 7.30) of the twelve sound examples.

to claim that the figure demonstrates a nice inverted U curve. The main problem here appears to be the lack of consensus on preference among test participants. Consequently, there is much spread in preference for sound examples of medium rated complexity.

Finally, the proposed SOF complexity measure turns out not to fare very well. The SOF measure and the timbral complexity evaluations are only correlated with $r = 0.448$, which is not significant ($p = 0.1439$). Hence, the SOF measure cannot be used to predict listener's evaluations of timbral complexity, at least not in the current set of examples. The SOF measure is better suited for distinguishing between simple test signals and anything slightly more timbrally varied. There were no such extremely simple examples in the test.

7.3.7 Qualitative descriptions

Many respondents used the opportunity to comment on individual sound examples. In some cases, there was a striking coherence in their descriptions and metaphor use. Descriptions ranged from onomatopoetic to technical or music theoretical qualifications, but here we list mainly those with more vivid or imaginative associations. The respondents' spelling and language have been retained. As usual, the numbered examples refer to correspondingly numbered sound files on the web.

Example 7.2. Ex. A1

Examples A1 and A3 are the most noisy ones, and have the shortest feature extractor window lengths. Water related descriptions often occurred for this group: "haché court désagréable" [short chopped unpleasant], "pizzicato-ish, raindrops", "regular, bubbly",

"bulles, low-mid, disgracieux" [bubbles, disgraceful], "motorboat", "fatty bullfrog, speeds up subtly", "blubb blubb dipp", "glooping squawk", "beeting heart, low freq", and some also mentioned birds or twitter.

Example 7.3. Ex. A2

Associations to robots were common in these replies: "haché plus long" [chopped, longer], "robotblipp i fast takt" [robot blip in steady pulse], "Sounds like electric shocks being delivered, not particularly musical", "fault-ridden downward microtonal arpeggios", "radiotubes", "robot does compute", "glib robot", "robotic, future, annoying, low key".

Example 7.4. Ex. A3

As some participant noted, this example was qualitatively similar to Ex. A1, hence there are again some mentions of bubbling: "sounds like bubbling porridge, not very musical", "mostly distorted bass with blips", "broken record player", "teeming, flocking", "spennande, kaffikjele, kokande vatn, drar kjenning på lyd :)" [exciting, coffee kettle, boiling water, recognise sounds], "Rumbling", "flappy squawk", "bouillonnement, clics, affreux" [boiling, clicks, horrid], "maskingevær med mute og liten frosk" [machine gun with mute and a tiny frog], "boulders, irritating", "påhengsmotor" [outboard motor].

Example 7.5. Ex. A4

This example is similar to A2, but slower. The inharmonic quality of the tones was captured in some of the descriptions: "robot/telephon", "sounds like telephone dialtones", "microtonal step sequencer reminiscient of Balinese gamelan", "singing tubes", "Photo copier", "atonal, bad keyboardist, rythmic".

Example 7.6. Ex. B1

Apparently, some elastic or bouncing quality of the sounds was noted in this section, with the use of characterisations such as: "nice jumpy flings on the verge of exploding were it not for fading out", "happy heart", "FM scratching", "wow, like a bouncing metal fly", "bounce squeeze", "gliss, attaques, ear fatigue", "springs, not rythmic, bad, high pitch", "e saw + scratch", "sprettball" [bouncing ball].

Example 7.7. Ex. B2

Starting at 45 seconds, this example finally reaches a decaying vibrato. Some comments relate to its peculiar form: "monotone au début" [monotone in the beginning], "Alarm", "a skipped CD version of B1, with a nice ending though", "sick growling heart", "bouncing metal, alternating in longer phases", "photo high", "gliss, laid, désordonné, fatigant" [gliss, ugly, disordered, tiring], "snapping, bouncing soundball(s) - rar slutt" [weird ending]. Yes, the end of this example actually differs from the others in the study by being the only one that approaches a steady state.

Example 7.8. Ex. B3

This example is one of the irregular kind, somewhat similar to B1: "more arhythmic than the B1 and B2, but seems more human at times", "haché" [chopped], "Full-on chicken attack", "crazy gnomes", "chaotic", "madcap, some arrhythmia", "squirrel", "chahut, cassé, rires, fatigant", [racket, broken, laughs, tiring], "trippetrippesnerre" [trip-trip-snarl], "not rythmic – most complex, worst", "dipp dipp dipp wawawa (nervous)".

Example 7.9. Ex. B4

This example begins with a very strictly repeating pattern with a deviation before the end, as some seem to allude to: "sinus heart goes wrong for a while", "Assembly line robot", "repetitive downward arpeggios as in A2", "windscreen wipers (crap)", "ratés, répétions, fatigant" [failure, repetitions, tiring], "gummiloop, tynn strikk" [rubber loop, thin strip].

Example 7.10. Ex. C1

Here the associations go in many directions: "fire alarm", "houseparty på 90-tallet!" [house party in the '90s], "some frequency beat bubbles, lockstep", "alien drop", "sinus, gargouillis régulier", [sinus, regular bubbling], "laserpistolaktig" [laser gun-like], "cool, rythmic, monotone, water", "droll droll troll droll (monotonous)".

Example 7.11. Ex. C2

A common trait in several of the following associations seems to be sounds with a certain granular quality, as in motors or snoring: "moteur qui démarre" [starting motor], "boblekvitter, repetitiv" [bubble twitter, repetitive], "froggy bubbles, that is all", "stockhausensk", "Edward Woodward", "reniflements, cynique, très laid" [snuffling, cynical, very ugly], "traktor i cyberspace, jevne hjul" [tractor in cyber space, smooth wheels], "mountain bird sound, snoring", "worst, gutsound, not rythmic", "blipp blip troooaa (almost repetitive)".

Example 7.12. Ex. C3

This example is described as: "problematic but musical circuit bending", "baklengs" [backwards], "melodi og snork" [melody and snore], "ah, radio på jenterommet tidlig 80 tal :)" [ah, radio on the girl's room in the early '80s], "modem song", "flerstemt" [polyphonous], "lots of tones, bad rythm, irritating".

Example 7.13. Ex. C4

Again the birds (and frogs, for some reason) are frequently mentioned here: "boblekvitter, mindre repetitiv" [bubble twitter, less repetitive], "Terminator laser riffle battle", "faster froggy bubbles in the season of mating", "radio, mellombølge, intessat for ein gammal radiovenn :) Takk!" [radio, medium frequency, interesting to an old radio enthusiast], "dispute d'oiseaux, horrible" [bird's quarrel, horrible], "kvittrar ubåtseko" [twitters submarine echo].

As can be gathered from this sampling of descriptions, the sounds in the Autonomous Instrument Song Contest evoked vivid images of the most varied sorts. Not all respondents used the opportunity to comment on the sounds, but most did (about 20 of 28

respondents, differing somewhat between sections). Some comments dealt more with aspects of musical form or synthesis techniques, as will be discussed in the next chapter.

7.3.8 Discussion

As already said, the study does not allow for any conclusions on an individual basis about any possible relationship between perceptual complexity and preference. Furthermore, the ready-made form used for the questionnaire led to some compromises regarding the design. It would have been preferable to present the sound examples in randomised order, which was not feasible in the present questionnaire. A more controlled study of the experience of surprise would necessitate a randomised order of presentation, since the first sound example will always be less familiar than those that follow, given that they are generated by similar means. The experience of surprise will be further discussed in the next chapter.

When composing music, using autonomous instruments or any other means, it is not common to present sketches to listeners and ask for their opinions before finishing the composition. This is not to say that such evaluations cannot be made, and examples may be found where the music has to serve a function in a more commercial setting, such as film music or advertising. In the present context, however, we would expect that autonomous instruments are mostly of interest to more experimentally oriented composers and if they choose to compose music with an autonomous instrument, they will evaluate the results themselves. Therefore, a listening test such as the present one is not intended to replace the composer's evaluations, even though it was presented as a song contest. That said, let us see what some respondents liked about the sound examples.

The open question about an overall favourite sound received varied replies. The most often mentioned example was B1 (six times). Since example B1 also got more votes than any other example as well as the highest score when disliking was subtracted from preference, it is the clear winner of the song contest. Next came example A4 (three mentions), which is perhaps different from the rest by its almost melodic character and slow pace. Examples A1, C2 and C4 were selected two times each as the overall favourite. Some participants listed multiple favourite sounds, but many did not have any clear favourite.

Although the Autonomous Instrument Song Contest asked about preferences, no general conclusions can be drawn about what aspects of the sound examples was liked or disliked because this differed very much among the participants. Comments ranged from "Nope, actually I liked them all" to "Nope, hated them all... I don't like synths".

As can be seen from the following replies, there can be many reasons for liking a particular example:

"Ex B1, it was highly novel and varied in timbre."

"B1 has a nice pace of variation versus repetition."

"Ex B.1, because of the physicality, agility and elasticity in the spectral deviations."

"I did not have a specific favourite, but I tended to prefer the ones that emulated some kind of warmth, suggesting an acoustic source, although they were all clearly synthesised. Perhaps A4, because I find it easier to relate to."

In conclusion, participants found it harder to assess the complexity of the examples

than their preferences, but were more in agreement on the complexity than in their preferences. The SOF measure is too simplistic to be used for predicting the perceived complexity of the sounds in the study, but may distinguish simple signals from those that have a higher degree of timbral variety.

7.4 Sampling based synthesis

In this section and the next, we suggest two ways to expand the range of applications of feature-feedback systems. The first relates to the use of sampled sounds, and the other is an extension to note-level algorithmic composition.

Autonomous instruments may use sampled sounds from any source that are triggered and possibly modified. The signal generator might then be implemented as a lookup table reading from stored and looped waveforms. Much of the theory developed so far, viewing autonomous instruments in terms of dynamic systems, is less applicable in that case. It is also a familiar problem that the most basic implementation of sampling synthesis does not lead to malleable sounds in the way that parametric synthesis models do, including additive synthesis and abstract models (Jaffe, 1995; Risset, 1991). Sampling a vocal sound and transposing it more than about a semi-tone will alter its character quite noticeably, an effect that was used and abused in some popular music from the early 1980s when samplers began to reach the market. On the other hand, concatenative synthesis using fragments sampled from high quality recordings has the benefit of allowing realistic renditions of sounds.

7.4.1 Matching targets to sources

Various synthesis techniques that use feature extraction for mimicking or adapting to an input sound were considered in Section 3.1. Concatenative synthesis in particular can easily be combined with any autonomous instrument, whether its output is used as the corpus or as the target to be matched. Sampling synthesis may also be used as the signal generator in a feature-feedback syste. Thus, there are at least the following three ways to relate sampling synthesis and feature-feedback systems:

- Sampling synthesis with stored wavetables is used as the signal generator in a feature-feedback system;
- The output of a feature-feedback system makes up the corpus in concatenative synthesis;
- The output of a feature-feedback system is used as the target, which is resynthesised by concatenative synthesis using another collection of sounds for the corpus.

In the first case, the sampler may have control parameters related to pitch, amplitude and perhaps various timbral aspects of the stored sample. Then, the sampler may be used similarly to the other synthesis techniques that have hitherto been used as signal generators. Another alternative is to read in the sound in a lookup table, then loop it and transform the samples in the table for each iteration. This is structurally similar to

Karplus-Strong synthesis (Karplus and Strong, 1983) or the cellular automata introduced by Chareyron (1990). The first alternative, using sampling synthesis with parametric controls in a feature-feedback system, is something we have not tried; the transforming sample loop, however, has been tried. The technique will not be described in detail here; suffice it to say that the period length of the loop is the most important parameter, although the character of the resulting process can be fine-tuned by the design of the transformation that is applied to the sample table. As the input sound is transformed again and again, it will soon be degraded and lose all of its original character. The process is directly comparable to that of Lucier's *I am sitting in a room* (see Section 5.1.4). Feature extractors can be useful in the loop to extract meaningful descriptions of the current state of the sample buffer and to decide what to do to it next.

Another idea is to generate databases of sounds from the output of an autonomous instrument. This database is then used as the corpus from which sound fragments are stringed together according to specification. There is nothing novel in this idea as such, although using the sounds from an autonomous instrument as the corpus for concatenative synthesis makes it easy to rearrange them in new constellations.

Concatenative synthesis needs a huge database of differentiated sounds for high quality resynthesis. If, instead, the corpus is taken from a synthesis model, there is actually no need to store all the sounds; all that needs to be stored is the synthesis parameters and the feature vector corresponding to the audio signal generated at those parameters. We will give some hints as to what will be needed to accomplish such a database. Using feature-based synthesis, the reconstruction of a target sound file should be easier if the synthesis technique is predictable in how it maps synthesis parameters to sound. This assumption fails spectacularly with feature-feedback systems, not least because hysteresis must be expected. This means that not only the current synthesis parameters decide the current sound, but also the complete history of the past values of the synthesis parameters. Therefore, we suggest to use granular synthesis with the grains taken from a corpus of sounds collected from a feature-feedback system and taken from different points in time, parameter space and initial conditions.

Michael Casey's recent developments of live processing using concatenative synthesis or "soundspotting" deserve to be mentioned here. In soundspotting, the live audio input from a performer is matched to a database of sounds which may consist of material recently played by the musician (Casey, 2009). The result is what Casey calls an "associative memory canon". The soundspotter application has some interesting features such as the option to select matches at a given distance from the target, including as distant as possible, motivated by the fact that the closest match is not necessarily the one that is musically most satisfying. Another detail that apparently turns soundspotting into some kind of (semi-autonomous) feature-feedback system is the optional use of feedback from the current matched output, which is inserted as the next target input. According to Casey, the use of feedback together with a restriction on the matching which eliminates recently chosen matches from consideration for a few cycles may lead to processes that have the characteristics of deterministic chaotic systems.

In the following, we will describe a simple implementation of concatenative synthesis and how to mash up the output of feature-feedback systems with it.

7.4.2 Search scheme

We have implemented a programme for concatenative synthesis, divided into two parts. First, the target sound file is analysed by a collection of feature extractors and the analysis is written to a text file. Next, a second programme reads the text file and opens a corpus sound file from which segments are retrieved to match the descriptors in the text file. The best matching segments are windowed, overlapped and written to an output sound file.

Extensive search for the best match in a large corpus database can be prohibitive, but there are shortcuts. Instead of comparing the distances between the target and each data point of the corpus, a more limited set can be considered by first sorting the corpus according to each of its descriptors taken one at a time, and finding a number of close matches in each descriptor considered by itself.

Suppose there are M different feature extractors $\phi_i, i = 1, \ldots, M$. Let $\phi^T(n)$ be the M-dimensional feature vector of the target at time n, and let $\phi^C(m)$ be the feature vector of the corpus at time m. First, we look for the N closest matches between the target at time n and all corpus segments with respect to each of the features in turn. The N closest matches with respect to each of $\phi_1, \phi_2, \ldots, \phi_M$ then result in a collection $\{\phi_i^C(m)\}$ of at most $M \times N$ segments. Next, we calculate the distances $d(\phi^T(n), \phi^C(m))$ for each m in the collection and find the entry that minimises the distance. The recurrence plots shown in Figure 7.1 actually use the same distance function as this search scheme.

If the features have different ranges, scaling may have to be applied. This applies to pitch extraction in particular, as we noted while discussing the SOF measure in Section 7.3.2. Different weights for each feature dimension may be useful in the distance function. Another option is to collect a different number of potential matches for each feature extractor. We have used the l_1-norm of a collection of feature extractors as the distance measure. The feature extractors currently used are: fundamental frequency and voicing, ZCR, spectral entropy and flux. Amplitude is not used in the matching phase, because it is easy to rescale the amplitude after a good match has been selected.

7.4.3 The catalogue method

In the previous chapter, we mentioned the technique of random sampling of the parameter space of the wave terrain system. With its large number of parameters, a fine-grained systematic search through this space would not be feasible. Now, the idea is again to sample the parameter space of a feature-feedback system. Let us therefore consider what information about the system needs to be stored in order to recreate a sound that it has produced.

Suppose that we create a catalogue of feature vectors related to the sound generated at a large number of parameter values of an autonomous instrument. This catalogue may then be used for concatenative synthesis by splicing together grains synthesised with the autonomous instrument. The target may be any audio signal or a sequence of sound descriptor values. Although the following scheme has not been implemented, a simpler method which yields an identical final result has been tried. The benefit of the proposed catalogue method is that no corpus sound file needs to be stored. The

quality of concatenative synthesis depends crucially on the size of the corpus and on how differentiated its material is. There may be situations where memory storage can be traded for computation time. In those cases, the proposed method could be useful.

If the autonomous instrument has been designed according to the proposed criteria, then it will generate non-stationary signals. Thus, the sound will vary over time, and so will the feature extractors that are applied to its output. Furthermore, different initial conditions may lead to fundamentally different behaviour. Hence, a full description of how to recreate a particular sound segment contains the initial parameter vector given to the signal generator, any other constant parameters and the time location or the offset into the sound from where the grain is to be taken. If all the specifiable constant parameters and initial conditions are lumped together into a vector π_0 and the time position is τ, then the needed data is contained in the vector $\Psi = \{\pi_0, \tau\}$.

Let $\mathcal{F}(\pi_0, \tau, L)$ be a feature-feedback system $\mathcal{F}$ with initial parameters π_0, started at $t = 0$ and run until $t = \tau$. The output signal segment of length L is then taken from that position and onwards. Thus, the catalogue will consist of a list of pairs of vectors. The first vector is the address to the synthesis parameters and time points Ψ, and the second is the feature vector $\phi(\Psi)$ related to that address. The matching algorithm takes a target descriptor $\phi^T(n)$ at time n and searches the catalogue for the closest possible match in $\phi(\Psi)$. This close match Ψ_i is then given as initial conditions and parameters to $\mathcal{F}$, which is run for the prescribed number of samples, after which a grain is overlapped and added, and written to the output sound file.

By dispensing with the corpus sound file, a huge amount of storage space can be saved since the catalogue file only contains feature vectors, lists of parameter values and time indices. Moreover, it may happen that several different parameter addresses map to roughly the same feature values. This is in fact very likely for static sounds, where little change occurs after an initial transient has died out, but may happen also in more varied sounds. Such a redundancy can be exploited. Parameter addresses that produce similar sounds may be pruned, keeping only the first occurrence.

As said, we have not implemented this catalogue searching method. Instead, an equivalent way to generate the same output is simply to search the autonomous instrument's parameter space and store its output in a corpus sound file. Then, standard concatenative synthesis techniques can be applied.

7.4.4 Assessment of concatenative resynthesis

With any reasonable implementation of the matching algorithm, the quality of concatenative synthesis depends crucially on the corpus. Perceptually close matches to the target are only possible if the corpus already contains material that is sufficiently similar. An often used strategy to extend the range of the corpus is to apply some simple transformations to each segment for better matches. Amplitude rescaling is straightforward and very useful, resampling may be used for simple pitch transpositions, and filtering may be applied. All of these transformations were used by Coleman et al. (2010), where the matching takes place against predicted values of the descriptors as if the transformations had been carried out; furthermore, they considered mixing more than a single grain from the corpus at each instant. Such mixing necessitates search by matching pursuit or

similar algorithms.

Our present implementation of concatenative synthesis is very basic, with no other transformations than amplitude rescaling. The output of the wave terrain system as it was searched by random sampling of its parameter space was used for the corpus in a series of sound examples with various targets. Further experiments were performed with material from the discrete summation formula system. In both cases, the material is too homogenous and restricted for verisimilar reconstruction of arbitrary sounds. In particular, the pitch ranges were too small. Steady tones in the target may not find any match in the corpus, leading to unstable wavering compromise solutions. It is a bit like hearing a tone-deaf person trying to sing.

Example 7.14. Unstable pitches caused by lacking matches can be heard in the concatenative resynthesis of Mumma's Hornpipe , sped up to five times its original tempo and resynthesised using sounds from the wave terrain system as corpus.

These limitations in the corpus material can partly be blamed upon the synthesis technique itself, that is, on the limited character of the sounds in the corpus. Nevertheless, both the wave terrain and the discrete summation formula systems are capable of generating any pitched sound and several inharmonic sounds as well. Thus, it is also a question of searching the synthesis model's parameter space systematically—instead of haphazardly, as is actually the case with random search.

Concatenative resynthesis using a corpus of sounds taken from an autonomous instrument may be instructive for the evaluation of the instrument's timbral qualities. The timbre remains restricted as the output is mashed up and recombined into new sequences. Only the characteristic temporal development of the autonomous instrument is destroyed. By granulation, juxtapositions of disparate elements may break up the smooth flow of the feature-feedback system.

7.5 Note level applications

So far we have only considered feature-feedback systems that generate a continuous sound stream. If the result is a musical composition, then this is a composition over a single long note so to speak, albeit an evolving and highly complex note. It will be illuminating to consider what the process in a feature-feedback system appears like when it is translated to the higher level of symbolic note representation. If the symbolic level is regarded as synonymous to common practice notation, then there is the vast field of traditional music theory to take into account, including theories of tonal and atonal music. Here, we will instead refer very selectively to traditional music theory and keep the ideas on a slightly more abstract plane. Our procedure, then, will resemble the though experiment of Xenakis (1992, p. 155), pretending to suffer from a sudden amnesia, and reconstructing music theory from scratch. For example, the existence of scales of fixed pitches or the division of time into bars and beats will not be assumed. Neither will there be any clear-cut separation of melodic line and chordal textures; there will only be notes given in something similar to a piano-roll representation.

In signal level feature-feedback systems, the synthesis algorithm generates a sequence of audio samples which is not divided into any separate events such as notes. The general principle of feature-feedback systems is that an output object is generated, which is concurrently analysed and mapped to the control parameters that decide how the next object is to be generated. So far, the object in question has been audio samples, but clearly the same principle can be applied to objects of other kinds, particularly to collections of notes.

Standard notation could be used to represent the notes; then this procedure becomes a variant of algorithmic composition in the traditional sense, where an algorithm produces an output of data that is translated into a standard notation score, which can be handed over to musicians for interpretation. Another option is to use a sound synthesis language such as Csound, although MIDI representations would also be adequate for our purposes. We will stick to Csound for terminology; translations to other representations are straightforward.

In Csound, a score file is specified with a list of note events that feed parameter values to an instrument which is specified in an orchestra file (Boulanger, 2000). A useful aspect of the score file is that its note events may be specified in other than chronological order, because the Csound interpreter will take care of sorting the notes before generating audio output. Each note minimally has three parameters (called *p-fields*): the instrument number, its starting time and duration. Other optional parameters may be specified, corresponding to user-defined parameters in the instrument. We will consider the case of an instrument with five parameters, where the fourth corresponds to amplitude and the fifth to frequency. No assumptions are made about the timbre of the instrument. This is at once the weakness and strength of working abstractly with symbolic note representations. Musical structures may be realised with any instrumental timbre one wishes, and several timbral variants can easily be compared. The drawback is that no information about concrete timbral aspects is available that applies in general to each and every instrument. For instance, the degree of sensory dissonance of a specific chord depends not only on the constituent pitches of the chord, but also on the spectral balance and exact tuning of partials in the sound (see Chapters 2 and 3, as well as Sethares (2005)). In practice, this is not a serious objection since once the instrument has been chosen, all details about its timbre will also be available to inspection—and, not least, to design.

In interactive music making, the note level and MIDI representations were explored extensively before signal level interactive music became common. As Robert Rowe (2009) has pointed out, despite the serious limitations of the MIDI standard such as its low data rate, it provides access to many high level features of music that are presently not within reach for signal analysis. Multipitch analysis of audio signals is not perfect even with the best current algorithms (Klapuri, 2008), and may still be too slow for effective realtime use, although this is an active field of research where improvements are regularly being made. On the other hand, Rowe argues that perhaps the note level information accessible through MIDI representations is not the most salient description of music, mentioning the difficulty humans have in transcribing a four-part harmony. However, we will be concerned with algorithmic composition at the note level, in contrast to interactive music, but in a way that will be in keeping with our philosophy of autonomous instruments.

7.5.1 Motifs and their descriptors

To begin with, we introduce some new terminology and a few descriptors related to collections of several notes. The descriptors will play a role analogous to that of signal level feature extractors. Much work in music information retrieval has been directed at similar descriptors, particularly in the context of tonal music. Our purpose of using motif descriptors is however different, firstly because the music we will have in mind is not constrained to lattices of pitch and time, and secondly because here, we are not going to apply them to the analysis of existing music.

Let $X = \{x_1, x_2, \ldots, x_N\}$ be a collection of N Csound notes, called a *motif* (the individual notes will always be written in lower case, whereas the motifs will be written in capitals). Note that "motif" is used here as a technical term different from its ordinary sense of phrase or melodic pattern. In particular, the motif is treated as a unit by the algorithm, but may consist of one or several perceptual units or Gestalts; or conversely, several motifs may combine into a single perceived Gestalt. The new note-level algorithm takes this collection as its unit, which is analysed for appropriate attributes and modified. Continuing this way, a composition is generated as a concatenation of these motifs X_k, $k = 1, 2, \ldots, K$. Actually, the motifs could be allowed to overlap in time, although the approach will be less confusing if we require that motifs be non-overlapping.

The procedure is similar to the compositional technique of writing variations on a theme. In fact, it resembles a metamorphosis technique even more, since the original motif (the theme) is forgotten by the algorithm already after the first iteration. For, similarly to the signal level autonomous instruments we have previously considered, the algorithm stores a recent past state of the system, but not its complete history. It is thus a Markov chain with the motifs X_k as its elements. In the following, we will focus on first order Markov chains, where the next motif depends on the previous motif but not on any motifs further back in history.

Assuming that no further information is stored about the collection of notes than the parameters of the individual notes, it turns out that several important musical facts about the motif as a whole are not immediately known. Unless the notes are sorted in chronological order (as they need not be), the motif's starting and ending times are unknown. It will be useful to introduce two time coordinates. First, a global time coordinate system is used on the level of the entire piece. Each motif X_k will have a start time with reference to this global time coordinate. Second, inside each motif there will be a local time coordinate for the notes x_n which always starts at 0 with the first note event, as shown in Figure 7.11. Under the assumption that motifs do not overlap and are laid out in chronological succession over the duration of the piece, we will not need to refer to the global time coordinate at all.

The start of a motif is found by sorting all the notes according to their start times. The ending time of the motif is the moment when its last sounding note ends. From this, the motif's total duration can be found.

The following notation will be used. Superscripts indicate the parameters of notes and motifs, where x_n^s is the start time of the note x_n and X_k^s is the start time of the motif X_k. Similarly, the superscript e denotes ending time; d is duration; a stands for amplitude and f for frequency. Then, the motif's density

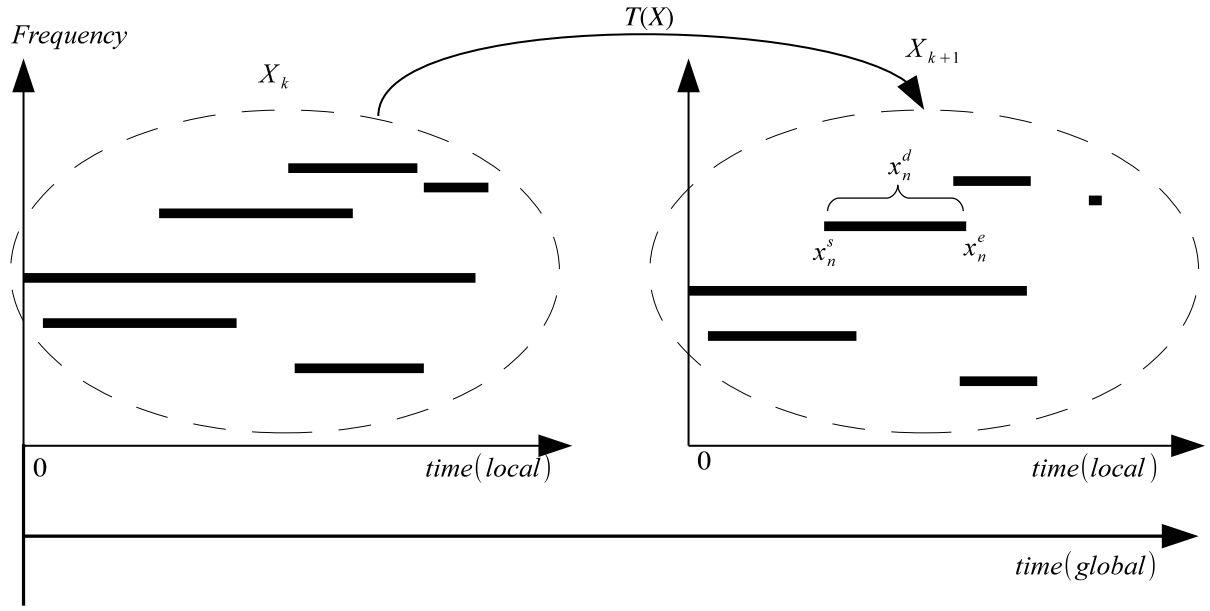

Figure 7.11: A transformation from one motif to another. Time points within motifs are transformed with respect to the local time coordinate system. Likewise, each note's start (x_n^s) and end times (x_n^e) are given with respect to local coordinates.

$$D_k = \frac{1}{X_k^e - X_k^s} \sum_{n=1}^{N} x_n^d \tag{7.31}$$

is the average overlap of notes, or the sum of all the notes' durations divided by the motif's duration. Staccato style motifs with a single note at a time will have $D_k < 1$, but if the notes always overlap, then $D_k > 1$. Combined occurrences of staccato and chords will provide contradictory influences on the density.

For the formula (7.31) to be valid, the motifs must not overlap. If, in addition, it is to be a sound estimate of the resulting density on the audio level, then the instrument has to be taken into account. Perhaps it is a simulation of a dry pizzicato; then, even if a note is specified with a duration of several seconds in the Csound score file, its effective duration may be only a fraction of a second.

If the notes of the motif are sorted by ascending start time, an average inter-onset interval

$$I_k = \frac{1}{N-1} \sum_{n=1}^{N-1} x_{n+1}^s - x_n^s \tag{7.32}$$

corresponds to an average temporal density if these intervals are more or less equal. However, it may be appropriate to treat multiple simultaneous onsets as a single note, even if they are not perfectly synchronised. The occurrence of almost simultaneous events, such that $x_{n+1}^s \approx x_n^s$, would bias I_k towards too short durations unless multiple events beginning more or less at the same moment are filtered out. Therefore, an estimate of

average tempo should only count note events farther apart than some threshold time interval ϵ, that is, those that satisfy $\left|x^s_{n+1} - x^s_n\right| > \epsilon$. Still, this simple descriptor is not intended to replace a proper beat tracker.

For the amplitudes, the average and standard deviation may be useful measures. If durations vary much, a weighted amplitude average

$$\bar{A}_k = \frac{\frac{1}{N}\sum_{n=1}^{N} x^d_n x^a_n}{\sum_{n=1}^{N} x^d_n} \tag{7.33}$$

may be more interesting. Steady increase or decrease in note amplitudes over time indicates a crescendo or diminuendo. A trend line could be fitted to the notes regarded as points in the onset-amplitude coordinate plane; positive slope would indicate a crescendo and negative slope a diminuendo. The same operations are relevant for analysis of the note's frequencies x^f. Together with the average, the span of the pitches $(\min x^f, \max x^f)$ describes the tessitura. At this point, no assumptions should be made about any restrictions on the pitches that may occur, in particular, they do not have to be restricted to any specific scale.

Under the assumption of octave equivalence, which holds for tones with harmonic spectrum (Sethares, 2005), the pitches may be reduced to the same octave by a modulo operation. From this representation, an histogram of pitch occurrences may be generated and then scales may be inferred. For this procedure to be reliable, sufficiently many notes have to be analysed, and perhaps more than are given in a single motif. However, we have not assumed the pitches to be restricted to any scale, so the histogram may provide misleading information. Imagine that the pitches are selected at random from an interval $x^f \in [f_0, f_1]$. Then, there cannot be a scale in the sense of a finite collection of pitches, and if some scale is nevertheless found, it must be a measurement artefact.

Depending on how the motif is distributed over time, it may be a single melodic line, a single chord, or any combination of these. Contour analysis can be applied to monodic textures. Chords, on the other hand, are characterised by such attributes as density, irregularity of intervals and degree of dissonance. The first two of these attributes are not dependent on any tuning system or scale. If the notes are fixed to equal temperament (12 tones per octave or 12-TET), then the whole apparatus developed for traditional music analysis becomes available, including pitch class (PC) sets.

7.5.2 Pitch set correlations

So far, neither the timing nor the pitches of the notes in a motif have been assumed to be quantised. Nevertheless, scales have certain advantages such as making it feasible for a musician to learn a limited collection of pitch intervals. For analysis and feature extraction on the symbolic note level, there are a few things that can be done with discrete pitch spaces that would be more complicated, if possible at all, in continuous pitch. Measuring the correlation between two motifs is one such thing.

Suppose we have a motif in 12-TET and would like to find what tonal key it belongs to, if any. Then, one could use a template of typical pitch distributions known to be associated with a certain key, such as those found by Krumhansl and Kessler (1982) and further elaborated by Temperley (2001), and match transpositions of those templates

against the motif. The transposition with highest scoring match would then correspond to the key of the motif. However, this technique generalises to finding matches to any pitch collection. Thus, one can calculate the correlation between the pitches in two motifs, where the occurrences of each pitch is treated as a probability and the motif is described by a vector, in this case with 12 elements representing the probability of each PC.

Since the vectors represent probabilities, their entries sum to 1; hence, we know that the mean entry in any vector is $1/N = 1/12$. In the calculation of the Pearson correlation of the two PC vectors, the mean would be subtracted from the probability values and the correlation can take values in the range $[-1, 1]$. It turns out that this leads to a somewhat counterintuitive correlation measure when applied to PC sets. For example, two disjoint sets of pitches would not have a zero correlation as one might expect. Preferably, a similarity measure should take the value one if the two PC vectors are identical and zero if they are disjoint. The *cosine similarity* of two vectors u and v,

$$r = \frac{u \cdot v}{\|u\| \, \|v\|} \tag{7.34}$$

is precisely such a measure. Then, we get a correlation $r \in [0, 1]$, where zero correlation means totally unrelated or maximally contrasting pitch vectors and the higher the correlation, the more similar the two sets of pitches. An application of the PC correlation to motif mappings is to check for similarity or contrasts in the previous two motifs. Taking $u = X_k$ and $v = X_{k-1}$ in (7.34) and comparing only the pitch classes, a measure is obtained that can be used to detect whether there is any harmonic change from one motif to the next. This can be seen as analogous to spectral flux or the various repetition detectors introduced in this chapter where two adjacent segments of a signal are also compared.

A simplification of this approach is to ignore the weights of the pitch classes, and only count occurrences and absences of pitch classes. Instead, we shall consider the complication that arises when no scale is given and there is just a set of pitches along the continuum of frequency. There are several ways this can happen, for instance there may be distinct notes of steady pitch, where the pitch can be chosen freely from an interval. Or the pitch might fluctuate continuously over time as in a yoik, and it would be possible to obtain a continuous pitch probability distribution. In the latter case, the correlation of two continuous pitch probability distributions is straightforward, it is simply obtained by replacing the summations with integrals in the appropriate way. For the case where the notes are steady and distinct, there are two practical alternatives. Note that this case can be described as comparing two continuous distributions which both consist of delta functions—chances are they will not coincide.

First, one can partition the pitch distribution, in effect introducing a scale that the pitches are quantised to. Then, the correlation proceeds as above. Second, the two distributions may be convoluted by a smoothing kernel before calculating their correlation. Each spike is then replaced with a Gaussian curve or similar function, which acts as a blurring filter that makes it more likely that the filtered spikes from the two distributions will partly overlap. The second approach makes no assumptions about what underlying scale the pitch collections belong to.

Pitch classes, as they are used in PC-set theory, assume octave equivalence. The modulo operation that reduces all pitches to one octave is quite a drastic transformation, which would render many musical textures unrecognisable. Thus, although the correlation operation can be used to infer such things as tonal relatedness, it ignores aspects pertaining to pitch register. It could be redefined to operate on the entire pitch register in use. Then it might be more efficient to compare broader registral regions, such as octaves. Instead of just considering the pitch dimension, correlations may be generalised to an overall similarity measure of motifs. If an excessive repetition of motifs should become a problem, repetition detectors based on an overall similarity measure will be able to detect it.

7.5.3 Motif transformations

In the simplest case, the number of notes in a motif always remains the same. Musical phrases do not come in standardised formats though, so it would be preferable to have variable length motifs. However, let us first list a few operations that are often used in common practice music.

- Inversion of pitches requires a pivot pitch around which all pitches are mirrored. This should normally be carried out on a logarithmic frequency scale such as cents from the pivot frequency. Another option would be to mirror frequency values.
- Transposition of pitch. This is simply the multiplication of all frequencies by the same constant, but any kind of warping function might be used.
- Retrograde or time reversal, and retrograde inversion.
- Augmentation and diminution of durations with or without change of tempo. Diminution with note onset times unaltered is illustrated in Figure 7.11.
- Permutations. Serialism had a predilection for this operation, which works by swapping onset times of the notes belonging to a motif.
- Sorting or partial sorting. Any single dimension of the motif may be sorted in ascending or descending order. All other dimensions remaining unchanged, sorting amounts to a permutation. Since sorting algorithms scan through the list of elements to be sorted in several passes, partial sorting can be used by running the sorting algorithm only for a limited number of steps.
- If the data for different dimensions are normalised to the same interval, then exchange of dimension may be used. Amplitudes may be swapped with durations, etc. This is an operation encountered in integral serialism. Obviously it is a very drastic transformation that completely alters the motif's appearance.

All the above operations keep the size of the motif, its note count, constant. Other operations insert or delete notes:

- Interpolation: between any two notes occurring sequentially in time, insert a note that is in-between with respect to pitch and/or other parameters.

- Free ornamentation: insert embellishing notes between two adjacent notes.
- Add a parallel line on a fixed or variable pitch interval but temporally synchronised to the first (mixture).
- Add a heterophonous line or other counterpoint.
- Delete notes, either with no other modification, resulting in silent gaps, or as if cutting out a segment (of time) and splicing the ends together.

Further restrictions on the motif can be useful in some situations. Imposing the constraint that motifs will be a single melodic line, there will be only one sounding note at any time, or no note at all if there is a rest. Then, the melodic contour is an interesting object to operate on. Contour analysis (Friedmann, 1985) provides several useful representations that may be used for motif descriptors as well as for transformations. For instance, the transformations may be subjected to constraints so as to retain the contour. There are also many ways to measure the similarity of melodic phrases (Grachten et al., 2004), such as the edit-distance, which counts the number of operations required for transforming one phrase into another. These operations may include any of the motif transformations listed above.

Let us again consider transformations that leave the note count invariant, this time in terms of mathematical functions. As said, each motif consists of N notes, $X_k = \{x_1, x_2, \ldots, x_N\}$, each carrying data for five musical dimensions, $x = \{x^i, x^s, x^d, x^a, x^f\}$, where x^i is the instrument number. Having several instruments in the Csound orchestra provides an opportunity to compose with contrasting timbres. Most of the motif transformations considered above map a single dimension to itself; furthermore, most of them operate in a uniform manner on all notes. For example, if pitch transposition is used, it applies equally to each note. Such limitations are by no means necessary.

The parameter space of each note (excluding instrument number) is

$$\mathcal{P} = (s, d, a, f) \in \mathbb{R}^4,$$

of which there are N instances, one for each note of the motif. Thus, the most general transforms of motifs that keep the note count invariant are functions $T : \mathcal{P}^N \rightarrow \mathcal{P}^N$. Seen in this unusual perspective, the transformations may include such exotic things as rotations by α degrees in the (s, f) subspace; i.e., the starting times and frequencies of the motif are rotated around some centre point in the coordinate plane. A 180 degrees rotation corresponds to a retrograde inversion, whereas a 90 degrees rotation means that points on the time coordinate will be interpreted as frequencies and frequency as time (running backwards). In the terminology of iterated maps, this rotation technique generates periodic orbits of period $360/\alpha$, if the rotation angle α is a rational number.

Compared to signal level feature-feedback systems, there is an important difference in the time scale at which motifs are transformed. Nevertheless, the principle that fixed point and periodic behaviour represent superfluous outcomes that might better be generated by simpler means remains valid. It is interesting to note that transformations such as inversion, retrograde, and retrograde inversion, if iterated, all generate period two orbits in the $\mathcal{P}^N$ space. Suppose for instance that we invert the pitches of X_k and

store that as motif X_{k+1}, then again invert the pitches and store the result in X_{k+2}, then $X_k = X_{k+2}$ for all k.

Transformations with other periodicities may be introduced; for example, as discussed above in Section (7.1.3), the maximum period of an iterated permutation is as long as the number of elements that are permutated. For high variability between iterations of the motif transformation, chaotic maps could be used. There are several ways to apply maps, such as using one separate 1-D map for each dimension, or using a single 4-D map on each note $x_n \in \mathcal{P}$, or even using a general map of the form $T : \mathcal{P}^N \to \mathcal{P}^N$ that cannot be reduced to a simpler form. In fact, Pressing (1988) already considered the benefits of using a single map of higher dimension rather than several independent maps for each musical dimension and found that the result can become too unpredictable and information-laden when all musical dimensions vary independently.

7.5.4 Example motif transformations

As we just saw, some familiar musical transformations such as retrograde and inversion, if they are iterated, lead to period two orbits in the motif space $\mathcal{P}^N$. Unless that is the desired result, we either have to avoid using transformations that have period two or find other ways to work around the problem. If several transformations $X_{k+1} = T_j(X_k),\ j = 1, 2, \ldots, J$ are used interchangeably, then some of the transformations may have a period two solution without the entire sequence X_k being of period two. To decide which transformation to use, the motif descriptors may be useful. This is in perfect analogy with the signal level feature-feedback systems.

Motif descriptors are also useful in case the consequences of iterating the transformations are not easily seen in advance. Then, they can be used as a check of the musical soundness of the resulting motifs. For instance, if one of the transformations diminishes note durations by a fixed amount of time, then we will want to guarantee that the durations never become negative. Much more so than in feature-feedback systems (on the signal level), here the motif descriptors capture high level properties that are musically meaningful. It becomes easier to imagine the effect of applying some particular transformation based on the status of the current motif. Perhaps this again has to do with the slow time scale at which the motif transformations are iterated. It should be said that very interesting processes can be created also without using motif descriptors, but it is easier to avoid senseless transformations if the available information about the motif is taken into account.

Example 7.15. Some tests were made with motifs utilising about a dozen different transformations chosen randomly, but no motif descriptors of the kind listed above. The motifs were melodic sequences of notes , so that simultaneous notes never occurred. Motifs were represented by a C++ deque structure, which allows easy insertion and deletion of elements in the beginning and end of the array. Most of the transformations changed the number of notes in the motif by deleting or inserting new notes. Safeguards were used to avoid deleting notes if there were too few in the motif, or against inserting new notes if there were already too many. Sometimes very short notes got organised in ascending frequency, thus producing emergent glissandi.

Some tests with certain deterministic and stochastic procedures for the choice of transformation indicates that the process becomes easier to influence in the desired direction when the transformations are chosen randomly. This may appear counter-intuitive, but often it turns out that stochastic processes are easier to design so that their outcome is more or less what one wants than complicated deterministic chaotic processes are.

All the transformations considered so far do something to the motif, but some transformations may impose a character of their own on the new motif. This can be understood by a comparison with constant functions, such as $f(x) = 1$. A constant function applied to a motif X_k ignores the properties of X_k and instead always generates some particular motif $T(X_k) = X_*$ for all X_k. This is not very useful if there is only a single transformation T, but if there are a range of transformations T_j to choose from, then a constant transformation serves the purpose of generating a specific motif that one can return to any time. Moreover, transformations that significantly reduce the range of variation in the notes' dimensions may be conceived of as "almost-constant" functions that will impose a recognisable character on the new motif. For example, multiplying all onset times with a small positive scale factor focuses the note entries to a single point in time. If the durations are similarly made more equal, any input motif will end up more like a chord with simultaneous onsets and equal duration of all notes.

Let us mention a few transformations that operate selectively on the notes by using available information on the motif. If there is a long note, it can be chopped up into several short notes on the same pitch. Conversely, if there are several consecutive short notes that are close in pitch, they may be merged into a single long note. The latter operation requires a motif descriptor made for detecting such repeated notes. The density of notes can be used to decide if the next motif should contain fewer or more notes. The composer can impose any constraints on the motifs. Indeed, the strategy of *constraint programming* (Anders and Miranda, 2009) as applied to computer assisted composition is very closely related to such context-aware operations on motifs. As Anders and Miranda argue, constraint programming is a very versatile method that can be used as a middle way between strictly algorithmic composition and manual composition. The constant functions that always yield the same motif are tantamount to manual specification of the musical material, whereas repeated application of one or more transformations that are automatically chosen leads to pure algorithmic composition. The *generate and test* method which has often been applied in algorithmic composition can also be implemented within the motif framework, although as it has been described here, the method is rather to test first, then generate the next motif.

7.5.5 Evaluation of motif mappings

Feature extractors in autonomous instruments introduce a substantial time window of memory of the recent past signal, as we have many times pointed out. Motif descriptors, on the other hand, typically operate on a single motif. In feature-feedback systems, the feature extractors have a temporal extension lasting for several hundreds or thousands of samples, and the next sample is essentially given by a function of this past window of samples. In motif mappings the descriptor contains only one single step of the object to be iterated, yet the duration of a typical motif extends well beyond common feature

extractor window lengths. This has several consequences. There is not the same kind of smoothing of past output in motif descriptors, although many of them use averages of dimensions inside the motif. Further, the vast difference in time scale makes it hard to compare the dynamics of the two types of systems directly. Motif mappings may need such a small number of iterations for a complete composition that the problem of transients leading to fixed points may not be observed in that short time.

New design problems and opportunities arise with motif mappings. The number of notes in a motif is a strongly influential parameter, especially if it is kept constant. Much can be done with sound design of the instrument that will be used to play the motifs. As noted, if common practice notation is used, it is easy enough to extend the output to instrumental music performable by human musicians. Another extension that might be interesting to explore is to use autonomous (signal level) instruments to play the structures generated by motif mappings. Conceivably, one could even set up communication channels between their signal level feature extractors and the motif-level transformations. Using several nested but independent levels may however be a more efficient strategy for generating complex results with autonomous instruments, as argued by Eldridge (2008). Insofar as the levels are not coupled, the problems of designing overall system behaviour is more tractable. Tying low and high levels together into very complicated feedback systems may however lead to frustrating experiences with undebuggable malfunctionings.

When used to implement some variant of constraint programming, the motif mappings may bridge the gap between algorithmic and computer assisted composition. It is relatively easy to design functions that work predictably on the motifs. Custom-made motif descriptors can be used to enforce specific constraints when the result of the transformation is hard to predict.

As a note level compositional technique, motif mappings are in principle amenable to be carried out by hand. It is even suitable for not so rigorous composition practice using a method that Morton Feldman has been said to practice. Feldman's extremely long pieces are often made of patches of repeating patterns, sometimes with subtle variations from one instance to the next. These variations were sometimes introduced by accident, by erroneously copying the notes from the previous sheet.

The idea of motif mappings could be investigated as deeply as any of the other systems we have described, so surely there remains much to do for anyone who finds the concept appealing. Here, however, we must be content with having pointed out how the principle of feedback from generated output to control parameters generalises to other settings.

7.6 Summary

We began this chapter by considering the important role of non-stationarity in music. Feature-feedback systems may or may not exhibit non-stationary dynamics after an initial transient has settled, but what matters more is whether this corresponds to perceptual stasis or not. Recurrence plots are convenient tools for visualisation of the temporal processes of feature-feedback systems.

Then, we considered practical methods to ensure temporal variation in the gener-

ated signal such as the use of a step sequencer to gradually traverse a set of predefined parameter values. The method of adaptive thresholds may be applied to any control parameter to enforce on it some prescribed statistical distribution of values. Being able to dynamically vary the window length of the feature extractor is also a very useful control method that helps expanding the range of sonic behaviour. Various repetition detectors were introduced, which may monitor the system and perturb its parameters such that it hopefully enters a region of more varied behaviour afterwards.

The two case studies of the tremolo oscillator and the discrete summation formula system then utilised many of the techniques of self-regulation away from stasis. Both of these systems also illustrate the emergence of a higher level of perceived organisation than what is directly specified in the algorithm.

The results from the Autonomous Instrument Song Contest indicate that the respondents generally agreed on their complexity ratings, but not on preferences, although many found it harder to evaluate the complexity. Whereas the simplistic SOF measure was not found to be a good predictor of the perceived timbral complexity, it might be useful as a stasis detector in feature-feedback systems. Because of the exploratory nature of this study, no further conclusions can be drawn from the complexity and preference evaluations at this point. However, a substantial part of the study consisted in qualitative sound descriptions, which will be further interpreted in the next chapter. The sounds, whether the participants liked them or not, evoked a wealth of associations despite issuing from abstract synthesis models not intended to model or mimic any real-world phenomena.

The use of external material in the form of sampled sounds is arguably a foreign element in autonomous instruments, or at least this undermines the ideal of creating self-organised sound by introducing ready-made material from outside the system. Nevertheless, Casey's soundspotting application may, in certain modes of operation, be seen as an instance of a semi-autonomous feature-feedback system. Since feature extractors are essential parts both of concatenative synthesis and of feature-feedback systems, it is only natural to consider possible relations.

Lastly, we saw how feature-feedback systems generalise to the note level. Although signal-level systems may emergently generate a higher level of musical organisation, admittedly, it is much easier to control the high level behaviour by operating directly on units at that level. Thus, the motif mappings can be seen as a particular form of algorithmic composition where the use of motif descriptors contribute to making it easier to acheive the musical goals one would like. In the next chapter we will discuss the composer's role in algorithmic composition and return to a few of the questions posed in the Autonomous Instrument Song Contest.

Chapter 8

Open Problems

Several issues concerning autonomous instruments remain to be addressed. Problems for the musicologist arise in the analysis of works that we suspect have been made with autonomous instruments. Other problems face the composer working with autonomous instruments. Parts of the ensuing discussion can be understood as the psychology of the composer of algorithmic music, a field in which studies are scarce. There are also aesthetic problems associated with how to make music with autonomous instruments, and how to present it.

The purist approach to autonomous instruments clearly is not attractive to everyone. It either means accepting whatever results come out of the algorithm, or it takes an endless chain of programming and tweaking of the algorithm until its output begins to sound acceptable. This is hardly spontaneous music making. It is no wonder then that semi-autonomous instruments offer a more attractive alternative to many musicians. Even with the slightest degree of interaction, the almost-autonomous instrument may be nudged into more interesting behaviour whenever needed. As long as the instrument retains some degree of autonomy, it will not just respond predictably to every input. Thus, improvisation will be the preferred mode of interaction. At least there is no point in trying to fix every aspect of the work if it depends on unpredictable responses from the semi-autonomous instrument. Improvisation within strict limits is still an option. This leads to the idea of *open form*, where some aspects of the work are fixed and prescribed in a score, and others are left to chance, the performer's choice, or the whims of the machinery.

The divide between music made by autonomous instruments and that made by other more interactive means does not necessarily materialise in the musical surface; instead, one has to know about the process of construction that led to the piece of music. Despite a prevailing understanding that many twentieth-century composers documented their compositional processes in some detail, there is often a lack of sufficiently comprehensive technical descriptions of compositions that might qualify as being made by autonomous instruments. Thus, we are left to speculate about the compositional methods. Works in open form pose other problems for the musicologist who wants to analyse the music from a single realisation when there is an endless number of potential forms. One solution here might be to ignore the conception of the work, open form or not, and to only discuss particular performances or recordings.

Generative music, being a strategy of automatically making new versions of a piece or constructing algorithms that produce ever-changing output, is a worthwhile form of music making by autonomous instruments. A more traditional output format is fixed media composition, or "tape music" as it used to be called in the days of analogue technology. There are aesthetic concerns about the choice of form in which the music is eventually presented, which will be discussed throughout this chapter.

8.1 Listening to autonomous instruments

The responses to the Autonomous Instrument Song Contest raise some intriguing questions as to what a listener can deduce about the generating principles from listening to sound files only, without having access to any further information. This situation is similar to the concept of *acousmatic listening* (Schaeffer, 1966), where the sound source is hidden from the view of the listener. Recorded sounds transmitted by loudspeakers always lead to an acousmatic listening condition, but the point is not merely that one is unable to see the sound source. Rather it is the absence of cues about the causes of the sound that is supposed to make it easier to focus on the sound as such. Lack of knowledge about the generating mechanism that caused the sound strengthens the acousmatic listening; but as we will see, the participants in the internet survey often speculated about the nature of the sound-producing mechanisms behind the autonomous instruments, and they were often on the right track.

After reviewing some of Schaeffer's listening theories and mentioning implications for the analysis of music made by autonomous instruments, more responses to the Autonomous Instrument Song Contest will be presented in the following sections. In particular, these responses are about the techniques that seem to have been used, as well as reactions of surprise or otherwise.

8.1.1 Analysis and listening

When listening to a musical work for the first time, say an acousmatic piece, certain curiosities direct the listener's awareness. Questions such as the following may arise: Whose music is this? When was it made, and how? What was the composer's intention? On the other hand, the listener may reflect upon how it feels to experience this music: What mood does it induce? What memories does it evoke? Both these objectively and subjectively poised curiosities may coexist, but neither is necessarily focused on the music as such.

To restate these listening curiosities in Schaeffer's terminology (as shown in the figure called *bilan final des intentions d'écoute*), there is *écouter*, turned toward the indexical function of sound, *comprendre*, to understand the meaning of the message, and *écoute reduite*, or reduced listening (Schaeffer, 1966, p. 154). The questions of what produced this particular sound in this piece, who made it and what technology was involved, all deal with the indexical side, whereas questions about the composer's intention and the associations it evokes in the listener belong to its meaning. By means of a phenomenological *epoché*, wherein the curiosity about the sound's provenance, causality and meaning are bracketed out, reduced listening is attained. However, reduced listening does not get

rid of identification (of sound sources) and qualification (of messages); rather it reorients the listening focus toward the *objet sonore*, that object which remains stable through ever-changing modes of listening.

To some, the concept of reduced listening may have a somewhat mysterious and intimidating aura. After all, it is often assumed to be an unnatural way of listening, one which goes against the grain of our natural inclination of identifying and understanding the events and objects of our environment. How can we know that we are really practising reduced listening? After all, the sound object will not suddenly sound like a different object when we do. Nevertheless, everyone has had the experience of uttering a word repeatedly until the relation to its meaning begins to feel unnatural and it turns into pure sound. Likewise, Schaeffer noted that they had been practising phenomenology in the studio, long in advance of actually knowing that they were doing so (Schaeffer, 1966, p. 262). Both the acousmatic listening condition and the possibility of an indefinite repetition of a sound fragment are procedures that facilitate reduced listening. With composers working in the studio, reduced listening actually becomes an ingrained habit; so much so that Denis Smalley even found it appropriate to warn against its excessive use: "... many composers regard reduced listening as an ultimate mode of perceptual contemplation. But it is as dangerous as it is useful" (Smalley, 1997, p. 111). The dangers are that while listening intensely to the sound materials of a composition, one may focus too much on "spectromorphological" features of secondary importance or small details, on background instead of foreground, on intrinsic qualities instead of referential aspects.

Another, albeit similar, peril of too close listening is what Smalley calls *technological listening*.

> Technological listening occurs when a listener "perceives" the technology or technique behind the music rather than the music itself, perhaps to such an extent that true musical meaning is blocked. Many methods and devices easily impose their own spectromorphological character and clichés on the music. Ideally the technology should be transparent, or at least the music needs to be composed in such a way that the qualities of its invention overrides any tendency to listen primarily in a technological manner (Smalley, 1997, p. 109).

What Smalley formulates here is an aesthetic position which is allied with acousmatic music but which is far from universally valid. The ideal of transparent technique is not shared by everyone (Cascone, 2000). Anyway, even if one agrees with the notion that technique should not be the primary content of the music, it may be hard to escape a certain emphasis on the technical side in music made with abstract sound synthesis and algorithms, as in autonomous instruments. To be able to perceive the technology used, the listener must be familiar with the typical procedures used. In this respect, it is interesting to note that so many of the replies to the Autonomous Instrument Song Contest contained more or less detailed, and always plausible, speculations about what techniques were used, as we will see in the next section.

The analysis of musical works has often been synonymous with the analysis of written music. Despite the availability of recordings, only recently have there been attempts to

analyse the interpretations, for instance by the extraction of tempo curves from several performers playing the same piece (e.g. Beran, 2004). Many analyses seem to have the objective (if not always explicitly stated) of uncovering the way the piece was written, as a kind of reverse engineering of the compositional technique. Forte's pitch class analysis is a case in point. The very opposite is the approach of Schaeffer and his followers. Here, it would be appropriate to speak of an *ideology* of reduced listening, arguing that what really counts is the sonic appearance of a piece of music, in contrast to any information as to how or why it was made, the social context in which it was made, the feelings it evokes in a listener and so forth. Evidence for such an ideology can at least be found in Smalley's remark just quoted, that "many composers regard reduced listening as an ultimate mode of perceptual contemplation."

Whether one agrees or not with the philosophical standpoint of reduced listening, it must be remembered that it emerged in a situation where "musique à priori" as Schaeffer called it was a norm amongst many European composers. In practice, the notion of *musique à priori* more or less coincided with serialism, but clearly it applies to any music that is planned in advance without much evaluation by listening to the results. Schaeffer's emphasis on the primacy of the ear was a reaction against those composers and analysts who had developed advanced ways of structuring the symbols of notation. This is lucidly expressed in *Solfège de l'objet sonore* (Schaeffer and Reibel, 1998, CD 1, tracks 2–3), where the electronic sounds of early music from the Cologne studio are juxtaposed with a phrase on a Jew's harp—the timbral similarity is striking; it is a "strange return to the sources" as Schaeffer puts it. See also the harsh critique of Stockhausen's *Studie II* (Schaeffer, 1966, pp. 613–623) for an example of Schaeffer's hostility towards serialism.

Is algorithmic composition with autonomous instruments a kind of *musique à priori*, then? The process of composing with autonomous instruments bears a certain conceptual resemblance to the strictly applied rules of integral serialism in being based on algorithms, albeit of a very different nature. What really matters is the care for the end result. It is perfectly possible to compose *musique à priori* with autonomous instruments by building up interesting systems without going through cycles of critical evaluation of the output and further refinements of the algorithm. On the other hand, what really made *musique à priori* a viable strategy for a period was probably the fact that those compositions were mainly notated in scores, where the composer could display the formal elegance of his compositional techniques whether or not it corresponded to anything the audience might be able to discern in the sounding music. With electroacoustic music, the extra communication channel of the score is removed; thus one cannot show off any technical accomplishments in that format, although exceptions certainly were made, such as Stockhausen's score for *Studie II*. Programme notes of course have remained a channel for giving away the compositional secrets. In the recent practice of live coding, some practitioners have felt compelled to give the audience access to the algorithmic process as it happens by projecting the computer screen with the written code for direct inspection (Collins, 2003; Brown and Sorensen, 2009). Such projections are of little avail if the audience does not understand the programming language. Live coding as such cannot be accused of exemplifying *musique à priori*, not least because it happens directly and with pressing cognitive loads on the performers (Nilson, 2007).

In the analysis of works or performances with semi-autonomous instruments, the

analysis may become problematic if one insists on the reduced listening strategy. Open works make the reduced listening ideology highly problematic as a point of departure for analysis, since it easily misses the point that the work is not necessarily the same twice. And here it is most appropriate to yield to the wish to find out how the music is constructed. Unfortunately, there is no way to find out directly from a recording of the music, except perhaps in a few rare cases such as Alvin Lucier's self-explaining *I am sitting in a room.*

There should preferably be several recordings available for comparison of different runs of the same system, but the availability of more than one recording is exceptional. The computer network band The Hub have actually released duplicate versions of a few of their pieces, judging from some recurring titles, at least. Since there is no score for this type of music, we will have to look at the testimony from the composers themselves in the form of their written descriptions.

In summary, the purpose of the analysis that it would be interesting to undertake in the context of autonomous instruments is to try to understand how a piece of music was made, and in the case of an open work, in what other ways it might potentially have sounded. This is not to say that other routes of investigation are less significant. Reduced listening, in particular forgetfulness regarding causality, remains important for assessing qualities of a sound fragment or an entire work where influences of extraneous knowledge may be disturbing. Imagine listening to what sounds like coloured noise with a periodic modulation, but being told that this is a sonification of some time series, such as the number of sunspots over time. Although a listener practising reduced listening will not easily forget this added information, its importance will diminish and the sound may be evaluated in relation to other similar sounds. In the context of complete compositions, this amounts to a certain irreverence for information that is sometimes communicated in programme notes or papers about the construction of the work.

Besides analysis of existing musical works, there is the strategy of simulation. Given sufficient knowledge of the mechanisms required to produce a certain piece of algorithmic music, new variant works of the same genre may be constructed. We have previously noted that Xenakis' GENDYN programme lends itself to such procedures. In principle, this may then offer insight into how certain compositions can be simulated. Nothing of course guarantees that these simulations based on the underlying functioning of a generating mechanism will strongly resemble the original. Conversely, a recording may be mimicked, or recreated by entirely different means and without any understanding at all of its generative principles. There have not been many studies of this kind so far, perhaps because the field of computer-assisted algorithmic composition is not yet very old.

8.1.2 Technical speculations in the Autonomous Instrument Song Contest

The respondents to the Autonomous Instrument Song Contest made many observations about the technical nature of the sounds presented. Given that no information was provided as to the sound's provenance and that no questions were asked about the apparent technique that had been used, it is quite remarkable that many responses often

mentioned standard signal processing and synthesis techniques. These labellings of the assumed techniques were often on the right track, if not entirely correct.

The character of this study being exploratory, there was no hypothesis as to how respondents were supposed to comment on the sound examples. A few examples follows (we have retained the original respondents' orthography):

Section A. Ex. A1 was described as "Filter with square LFO, resonance chock", and "bass pulses with clicks and low amplitude higher-frequency blips".

Ex. A2: "S&H, clicks" probably referring to sample and hold, "metallic ring modulation".

Ex. A3: "mostly distorted bass with blips", "like A1 extremely rapidly, digital distortion".

Ex. A4: "sounds like telephone dialtones", "ring modulating effect, LFO pulse on amplitude, sequenced", "slow sample and hold, filtering, regular", "4-part filtered sinus". The telephone dialtones referred to consists of two inharmonically related sinus tones, an effect that can be mimicked by the discrete summation formula system. Similar sounding effects may be produced with ring modulation of sinusoids.

Section B. Ex. B1: "Ring mod, shuffle", "FM scratching". The latter is an interesting term; scratching of course involves speeding up and slowing down a record, playing alternatingly forwards and backwards, FM being a side effect of this.

Ex. B3: "chaotic". Little is known about how good listeners are at distinguishing the sounds of chaotic from regular or stochastic time series. Of course the designation as "chaotic" may have been made in a colloquial sense of being "messy".

Overall, there were more qualitative comments about the sounds in section B and few technical descriptions, as if the synthesis technique used here was harder to guess than in the other sections. No comments pertaining to technical aspects were made regarding either Ex. B2 or Ex. B4.

Section C. Ex. C1: "sinus sequence, repetitive", "downward arppegiated sinc pulses in the low-middle range of hearing", "pitchgrainy pulse. Inverted filter. Different tempi of LFO and OSC. But even". The observation that there were different tempi (in the "LFO and OSC") seems to reflect the fact that different time scales of parameter updates were used in this section, as explained in the previous chapter.

As many respondents remarked, Ex. C2 consisted of a repeating pattern, but there were no comments related to the synthesis technique as such.

Ex. C3: "Random faster random tones (pulse LFO) for each turn", "sinus, basses, ringmod". The tremolo oscillator uses a sinusoid waveform, as is perhaps alluded to here.

Ex. C4: "more frequency beating". Most comments in section C consisted of either free associations or descriptions related to patterns, repetition, rhythm and timbre.

General remarks. The following comments about general similarities across all sound examples also touched upon technical matters:

"Ils semblent tous monophonique, c'est à dire issus d'un seul oscillateur" [They all appears to be monophonic, that is, issuing from a single oscillator], which is true.

"All were synthetic, algorithmic, oscillator-based", a very concise and correct description.

"Only one sound at a time, and sounds like there is some sort of ring modulation going on, making tones less pure. Also, the rhythmic and melodic portions of the sound mix in a strange way, of which I'm not sure if I like it. There is some rhythmic repetition, but there is randomness that tries to obscure it. But it sounds most of the time more like random stuff rather than planned progress." Randomness obscuring rhythmic repetition—this is a good way of putting it. As discussed in Chapter 6, near-recurrence to previously visited points of phase space is a hallmark of chaos. There are reasons to believe that such recurrences are actually made audible in the feature-feedback systems used here.

"Most are obviously algorithmic, except B3 sounds like it was performed by a human." We will have more to say about the speculation of a human performer later.

"Electronically generated, seemingly algorithmically controlled on a high-order level. Assuming some chaotic elements, where small variation in parameter value can yield surprisingly strong changes in the generated sound." An unusually perspicacious observation, except for the speculation of a high-order control which is debatable.

"Some sounded very FM, others used varying amounts of ring modulation. In general they mostly sounded like variations of 3-4 different synths." Ring modulation is a recurrent theme. Although not used as such, both the wave terrain system and the discrete summation formula technique can be formally related to RM. As discussed in Chapter 3, the tremolo oscillator is closely related to FM. The respondents might have guessed from the three sections that there were also three different synthesis models, but it is not clear why this respondent thought there might have been three or four different synthesis models.

"There was definitely a lot of similarity inside each group, almost as if each group was a synthesizer with the instances having different parameters. At any rate, the action seems like it happens on different time scales in the variations." Indeed, each group, here probably referring to the three sections, used its particular synthesiser.

"The A and B groups sound like there is some chaotic feedback through cross-coupled oscillators, the C group sounds like modulation of sine tones without feedback, so it's not as dynamic and interesting." Of course there is a feedback process going on also in the sounds of section C, but it would be interesting to try to explain why they are perceived as differing from the rest.

"Other than the fact that they are all computer synthesised, nothing obvious springs to mind."

"It all sounds quite 'synthesiser'. Most sounds appear to be based on simple waveforms. Most of the timbral complexity appears to result from rhythmic density and pitch alterations." This comment can perhaps be related to the fact that it is the control functions, not the oscillators as such, that do the job in these systems.

"Ils sont tous mono, extrêmement laids, bruyants et mal produits, extrêmement prévisibles et redondant." [They are all mono, extremely ugly, noisy and ill produced, extremely predictable and redundant.] Yes, they were in mono, and not "produced" or post-processed very much, except for some lowpass filtering for one of the sections.

"låter som samma syntesteknik använts, så de befinner sig i samma ljudvärld. nästan alla har ett rytmiskt mönster eller frasering som de håller kvar vid - några byter abrupt,

men de är i minoritet." [Sounds as if the same synthesis technique has been used, so they are in the same sphere of sounds. Almost all cling to a rhythmic pattern or phrasing—some change abruptly, but they are a minority.]

"Most seem based on some kind of algorithm. Often, there are small variations between repetitive themes. Reminds me of minimal music." Is it merely a cautious reservation that *most* sounds seem to be based on an algorithm, or do some sounds appear to be made by other means?

"Alla är baserade på syntetiska toner utom A3 som är lite brusig också, men det kan vara en effekt av extremt snabba tonskiften." [All are based on synthetic tones except A3 which is a bit noisy too, but that may be caused by extremely rapid shifts of tones.] Different window lengths were used in section A, of which the shortest one was for Ex. A3.

"Alla följer en sorts puls, ofta med oregelbundna accentueringar. Alla bygger upp lite mer komplexa klanger av tonerna, ofta genom ringmodulering eller väldigt snabba tonglissandon som har klanglig eller perkussiv effekt" [All follow a kind of pulse, often with irregular accents. All of them build up more complex sounds of the tones, often through ring modulation or very fast tone glissandi that have a timbral or percussive effect.] Ring modulation mentioned again!

One respondent gave this final comment:

> Interesting work, I've explored similar territory through mixer feedback, "chaotic" synth programming and software algorithmic composition. It's obviously fairly difficult to characterize because a sound's "function" may change dramatically with only a small change in parameters, and this sort of thing must also address human perceptions of information versus noise and preference for an appropriate balance between stasis and change, expectation and surprise, etc. It's not intuitive at all. Good luck.

These insightful comments raise the question of how much an informed listener can really appreciate of what is going on in a system that generates these kinds of sounds. The feature-feedback systems used in the questionnaire are definitely original systems, albeit composed of common signal processing and synthesis parts. Other practitioners of electronic music might thus recognise timbral cues such as ring modulation, or perhaps rhythmic and formal cues that may indicate chaotic feedback systems.

8.1.3 Surprise, surprise!

We have pointed out the central importance of surprised reactions to the result of automated composition processes and the problematic relation to the concept of emergence (cf. Section 5.3.1). Expectations about what will happen is the necessary condition for the experience of surprise. If a large number of alternative events are about as likely to occur at any given time, then none of them will be more unexpected than any other. Inasmuch as this principle is valid, entropy can also be related to the perception of complexity.

With a melody sung or played on an acoustic instrument, the listener will probably have a certain sense beforehand about what approximate range the melody is likely to

occupy. We know that it is uncomfortable to sing very high or low tones. So when listening to a singer, we might expect tones in extreme ranges to be exceptional, and when one of those tones has occurred, we might expect the singer to return to a more relaxed medium tessitura.

It is thus quite natural that certain expectations follow listening to melodies. When a large skip has occurred, a gap-filling motion in the opposite direction can be expected, either immediately or after a few more notes. This is not so much a matter of convention as a necessary consequence of limited instrumental and vocal ranges. For if a large skip up in pitch were followed by several small steps or skips in the same direction, the instrument's range would sooner or later be exhausted. And even if the tones were electronically generated and the register were theoretically unlimited, the limits of human perception would soon put a stop to the experience of an infinitely rising pitch contour.

Stéphane Roy (2003) has adapted several of the most important analysis methods for instrumental music to electroacoustic music, amongst them the theory of Leonard Meyer. As Roy observes, the situation often occurs in electroacoustic music that a skip in some musical dimension occurs, later to be filled in with contrary motion.

> En musique électroacoustique, le saut d'espacement (*gap-fill*), lorsqu'il concerne non seulement la dimension mélodique mais aussi les autres dimensions musicales, est un phénomène fréquemment observable : une allure dynamique peut augmenter instantanément l'amplitude ou l'ambitus de son oscillation pour ensuite rejoindre en *diminuendo* son état initial; une unité qui présente soudainement une grande intensité dynamique se résout peu après par un decrescendo; un son inharmonique succédant subitement à un son tonique peut, par procédé de filtrage, recouvrir sa forme initiale de son harmonique; un espace fermé peut tout à coup s'ouvrir sur un vaste volume spatial pour lentement se refermer sur lui-même; et ainsi de suite. Tous ces changements soudains peuvent dans certains contextes stylistiques générer un certain niveau de tension et devenir implicatifs (Roy, 2003, pp. 503–504).
>
> [In electroacoustic music, the gap-fill, since it not only applies to the melody but also to the other musical dimensions, is a frequently observable phenomenon: an allure of dynamics may suddenly increase in amplitude or range of oscillation, later to return by diminuendo to its initial state; a segment that suddenly presents a great dynamic intensity resolves soon after by a decrescendo; an inharmonic sound suddenly succeeding a tonal (harmonic) sound may, by filtering, recover its initial harmonic form; a closed space may all of a sudden open up to a vast spatial volume and calmly close around itself again; and so on. All these sudden changes may in certain stylistic contexts create a certain level of tension, and generate implications.]

This idea of gap-fill translates well to some cases of sounds generated by feature-feedback systems. One such case is the example given at the end of Chapter 6 (see Figure 6.22 on page 262). After tens of seconds, up to several minutes, the texture may change to something quite different. A normal listening expectation in such circumstances might be to suspect that any of the textures that have been presented so far might return later on. Of course some textures may have a very anonymous character and not lend themselves

well to being memorised, but when the contrast is well marked, one may suspect that the texture in the former section will occur again after a while.

A different kind of expectation follows when experimenting with the parameters of a feature-feedback system, in trying out various parameter values and building a knowledge base of experiences of parameter-to-sound relations. In that case, one may get a feeling for "typical" system behaviour, and one may notice unexpected deviations from that norm. For obvious reasons, the audience of a piece of music composed by such algorithms can only experience the first kind of surprise, that is, a surprise that comes about as expectations of the musical continuation are thwarted.

8.1.4 From astonishment to indifference

Let us now review the responses to the question about surprise in the Autonomous Instrument Song Contest. The question was: *Were you ever surprised by any of the sound examples? Which one, and why?* Here are some of the answers, divided into three groups.

> "A1 – sounds like there's some clipping on my speakers"
>
> "I was unpleasantly surprised by A1 and A3, but was acclimatised by later on. I was expecting more traditional music, and this would be far more experimental."
>
> "Vil nevne samme eksempel [A4], som skilte seg fra de øvrige." [Would mention the same example (A4), that differed from the rest.]
>
> "kanskje av a3 fordi det minnet om japansk støymusikk og visste ikke at risto lagde det, menjeg likte det veldig godt. kanskje litt overrasket over at lydene i eksemplene var så like." [Maybe by A3 because it reminded me of Japanese noise and I did not know Risto made such music, but I liked it very much. Perhaps a bit surprised about the similarity of the examples.]

The only variation between the four examples of section A was the window length of the feature extractors, and two of these examples (A1 and A3) were quite noisy because of short window lengths. This may be the reason why the first example sounded like there was some clipping. All the above replies are related to contextual expectations, concerning musical style and deviations from what were deemed to be typical examples, in contrast to the internal or form-related expectations that were decisive in the next section.

> "B2 started very repetitive, but then it changed"
>
> "B4 makes an interesting variation toward the end, but is otherwise mostly static."
>
> "Mildly surprised by B.4, as it changes towards the end, when one expects it to continue as it was."
>
> "B1, fikk assosasjoner til afrikansk strengeinstrument (1 streng)" [B1, had me associate to an African string instrument (one string)]

> "B4, eftersom den flippar ur en liten stund efter ett tag, efter att den verkade syfta till att vara monoton mesta tiden." [B4, since it freaks out for a while, after appearing to aim towards being monotonous most of the time.]

Section B, using the wave terrain model, contained some examples with contrasting variations amidst almost repeating patterns, as several respondents point out. It is easy to explain the surprise effect of example B4, which has a deviating episode (at 0'28–0'35) and then "returns to normal". Here, an almost identically repeating pattern has built up the expectation that it will continue indefinitely. Ex. B2 is perhaps the one that is most varied, and might be analysed as containing three form sections. First there is the repetitive start, as one respondent points out, then there is a long transition, and finally a steady tone is approached by a decaying vibrato. It is less obvious why two respondents mention B1, which is of the irregular type. Perhaps there is an effect of novelty; if one listens to the sounds in the given order, the first sound is bound to be more surprising than the rest, which inevitably will resemble other previously heard examples. That is why there was a priming example in the beginning which consisted of representative examples from all three synthesis models. As indicated by one respondent, it fulfilled its function to some extent:

> "Den første, for å stille inn volumet. Jeg viste ikke hva slags lyder jeg skulle forvente i utgangspunktet. Dessuten opplevdes den som svært variert, springende." [The first one, to adjust the volume. I did not know what kind of sounds to suspect to begin with. Also, it appeared as very varied, running.]

Ideally, perhaps there should have been more priming examples, since all surprise-causing examples were drawn from sections A and B and no one mentioned any example from section C. In a more formal study, the order of presentation should have been randomised, but this was not feasible in the current experiment.

Finally, there was a group of more or less blasé responses.

> "No sound has ever surprised me. Haunted me, yes; but not surprised."

> "Not really, the were all a bit strange but some changed timbre halfway through..."

> "Non, tout ça a été entendu des milliers de fois." [No, all of this has been heard thousands of times before.]

Of course each participant will have their own interpretation of how much it takes for something to be surprising, from causing shock to barely noticeable changes. The building up of expectations by repetition or static characters is essential for the surprise effect that is most interesting to try to achieve with autonomous instruments. As demonstrated by the responses that pointed out examples in section B, this effect does occur to a certain extent, although the other replies indicate a wide range of reasons for being at least mildly surprised on first hearing these sounds.

8.2 Open works

Interaction with semi-autonomous instruments will never be entirely predictable. Fixing a single version of a piece then becomes pointless, and instead one will have to accept a range of outcomes. This leads either to works in open form—that is, works that are flexible with respect to the order of sections or the exact nature of the material it contains—or it leads to some form of improvisation.

Next, we turn to the problems of music analysis that occur when little is known of the compositional process. The examples are drawn from generative music, tape composition, and some works that involve semi-autonomous instruments.

8.2.1 Aleatoric tape music

All performed music will be played slightly differently on each performance, even if there were a hypothetical ideal performance to strive towards. Open works take the variability of performance to its extreme, in effect challenging the identity of the work. When a recording of an open work exists, or an improvisation for that matter, it may be given the status of a documentation displaying one of several possible realisations. The analysis of open works has to deal with this problem. Although analysis of the actual sounding documentation may be revealing and useful as a listening guide, it cannot tell the whole story. It does not explain why the piece sounds as it does, and it gives no clue as to how it might sound on another occasion. Additional information thus becomes crucial. A score, if available, should provide necessary clues, otherwise a composer's commentary or a diagram of the setup used may prove helpful.

John Dack, in an introduction to Pousseur's *Scambi* (1957), talks about how the techniques of tape music made open form unpractical to realise in electroacoustic music (Dack, 2005). Open form was used at least occasionally by many composers in instrumental music at the time, beginning in the 1950s. The openness of form may appear at different levels: either the piece is constituted by fixed sections that may be played in an arbitrary order, or there are choices to be made on the small scale of phrases, such as the order of notes in a phrase. *Scambi* has the former kind of openness, and for obvious reasons. Within the tape medium, it would be unfeasible to allow the restructuring of small constituent parts while the overall form remains intact.

As Carl Dahlhaus discussed in one of his few texts that deals directly with electroacoustic music (*Ästhetische Probleme der elektronischen Musik*, from 1970 (see Dahlhaus, 2005)), aleatoric procedures were introduced in instrumental music for the sake of giving the performers an opportunity to regain a certain liberty at a time when the notation was becoming increasingly detailed. Ironically, many performers preferred to fix one version instead of improvising. Dahlhaus clearly did not appreciate the strategy of delegating the choice of the final realisation to the interpreter, who then had to complete the unfinished job of the composer. Also, the use of aleatoric form in electroacoustic music appeared contradictory to Dahlhaus, who noted that the interpreter was dispensed from electroacoustic music, yet was necessary for the realisation of aleatoric compositions. These are surely dated remarks, with a bearing on the historical situation when fixed media (tape) composition was dominant and live-electronics still unusual. For one thing,

the composer/performer divide is no longer taken for granted, as current electronic music practice is full of composer/instrument builder/programmer/performers all in one person.

Today, with digital audio workstations and algorithmic composition software, open form of any kind is easily attained. Yet it appears to be uncommon today to make works that can be assembled from short formal segments into any desired order. Free or restricted improvisation flourishes, and there are many examples of generative music that is made to reveal different facets on each presentation, but these are other approaches than that of *Scambi*.

Pousseur himself gave a detailed explanation of the construction process that led to *Scambi*, that is, to the set of 32 sections that can be assembled into a version of *Scambi* (Pousseur, 1959). One important technique was to use gated noise with various gating thresholds (this was called an "amplitude selector" or "dynamic filter" in those days). This material was further processed by varying the tape speed, feeding the sound into an echo chamber, and other processes. Finally, Pousseur reasoned that beginning to cut the tape would imply taking new directions that he wanted to avoid, so instead he decided to use the gating technique once again on the material as a way to choose the parts that would be retained independently of his own will or predilections. In that sense, the generation of material has an autonomous touch to it. There were certainly many choices made by Pousseur at all levels, but the gating technique is in its simple way an example of signal-adaptive processing that leads to results that the composer cannot decide in every detail. What, then, could be more fitting than to make an open form piece of the material?

8.2.2 Interpretation and the open work

Umberto Eco's book *The Open Work* (Eco, 1989) introduces the idea of openness in artworks with a few compositions that are notated, but whose parts can be performed in a number of different ways. These include such compositions as *Klavierstück XI* by Karlheinz Stockhausen, the third sonata for piano by Boulez, and the previously mentioned *Scambi* by Henri Pousseur. Such works are open and flexible in an obvious way. Eco then gives several other examples of visual art, music and poetry that have a fixed form but are nevertheless open to interpretation, although he claims that some works are more open than others. Joyce's *Finnegans Wake* is one example; although the text is fixed and objectively remains the same each time one opens the book, it is so loaded with possible and mutually conflicting interpretations that each time one returns to any particular sentence, one is likely to discover a new meaning in it.

After quoting a passage full of vivid pictorial associations by an art critic trying to describe a painting, Eco contends that "half of his reactions have nothing to do with an aesthetic effect, and are merely personal divagations induced by the view of certain signs", the viewer being "more involved with the games of his own imagination than with the work," which prompts Eco to ask, "is it a limitation of the work that it should play a role similar to that of mescaline?" (Eco, 1989, p. 93).

We need to differentiate between open form and openness to interpretation. Eco does not do his best to keep those two concepts apart, which may be confusing, although he might argue for the continuity and interrelatedness between the two variants of openness.

Raymond Queneau's book *Cent Mille Milliards de Poèmes* is a collection of ten pages of stripes with a single line of text on each that can be combined in any way with thirteen other lines, each chosen among the several "pages", to form a complete sonnet. This is surely an open work and also a work in open form that is not fixed to one single form of appearance. Dack (2005) mentions as literary examples of open form Queneau's poems, as well as Mallarmé's unfinished *le Livre*, an ambitious project containing a collection of loose sheets of paper that could be assembled in any order by the reader. Dack then reminds us of the inherent openness in any performed music, and notes the absence thereof in acousmatic music. Although live spatialisation brings in an element of interpretation, it cannot do anything about timing. There have been attempts to accommodate flexible timing controlled by a performer, using the conductor metaphor (Mathews, 1995). Conducting an electroacoustic piece is still essentially adapted to works that exist in an almost fixed form. A more radical alternative is to make the performer directly in charge of every aspect of the music (like a violinist in full control over minute inflexions of the tone), or various intermediate forms of interaction with semi-autonomous instruments.

Suppose we were to modify one of the feature-feedback systems of the previous chapters by giving it some amount of direct input controllable by a performer. Let us say that we would like to control the tempo of some sequence of events. Although we have not tried this, it appears likely that the conducting type of interaction would not be effective, insofar as it assumes a predetermined course of musical events whose timing (and perhaps dynamics and timbre) remain flexible. The hysteresis that pervades these feature-feedback systems would make it utterly impracticable to map input parameter values to specific sonic characters; what actually unfolds will depend on things such as the velocity of parameter change. Nevertheless, if one renounces total controllability of a timbral or morphological kind, it may be possible to design a feature-feedback system that responds relatively predictably to external control. Again, this has not been tried, so the effectiveness and musical interest of such amendments remain to be investigated.

Another intriguing question is what does it take for an open work to remain recognisable as one and the same, despite appearing in various highly differing manifestations. After all, tunes exist in orally transmitted music cultures, such as jazz and various folk music, where there may be a much greater flexibility in the way the piece is performed than in Western classical music.

8.2.3 Ambiguity, complexity

Openness of the kind that Eco describes in fixed works, that is, works that are not themselves undergoing any major transformation from one presentation to the next, is related to perceptual ambiguity. It is also related to complexity, although arguably ambiguity is just a means to achieve perceptually complex results. Ambiguity can be exploited in music, as has been done quite deliberately by Steve Reich in the choices of rhythmic patterns that may be grouped in either triple or duple meter, and as these simple patterns are repeated, one interpretation is never settled upon as more correct than the other (Reich, 2002). It would be hard to argue the case that Reich has made complex music, at least when compared to composers such as Boulez or Ferneyhough; but on the other hand, it is

harder to demonstrate how similar perceptual ambiguities are exploited in more complex music, perhaps because such ambiguities exist in an overwhelming number. Semantic ambiguities in texts are perhaps more familiar than musical ambiguities. Well-known examples of visual ambiguity (or bistable perception) can be found in the Necker cube or in figures that may be interpreted alternatively as a young or an old woman, or a rabbit versus a duck. Shepard tones and the various perceptual "paradoxes" they give rise to are perhaps the best musical counterpart of such ambiguous phenomena. Apart from that, the grouping into streams in auditory scene analysis (Bregman, 1990) may be open to varying interpretations. Grouping is probably the most general aspect where ambiguity may be deliberately exploited in music, including the less researched genres of contemporary instrumental or electroacoustic music.

In fact, Temperley raises similar ideas when discussing listening strategies, albeit with a focus on tonal music: "In a complex, well-written common-practice piece, the search for the optimal encoding is a dynamic process, requiring constant attention and providing a significant (though not insurmountable) perceptual challenge for the listener. [...] The listener must also be constantly alert for patterns of repetition which may allow a more efficient encoding" (Temperley, 2001, pp. 333–334).

Complexity and its relation to encoding and compressibility was discussed in Section 5.2.6. Temperley's reasoning vaguely echoes that of Schmidhuber, who argues that the experience of beauty has to do with patterns and efficient cognitive encoding strategies. A measure of timbral complexity using the relation between compressibility and entropy was proposed by Streich (2006). Measuring (and even defining) timbral complexity is nontrivial. In Streich's implementation, timbre has to be discretised first by making some partition of feature values. There always remains room for doubt as to the adequacy of the chosen partition, in contrast to an already discrete domain such as pitch classes. Electroacoustic music which abstains from fixing pitches and events onto lattices is inherently harder to relate to any complexity criterion that presupposes discrete sets of pitches, inter-onset intervals etc. In the Autonomous Instrument Song Contest, the questions about complexity were split into two facets that seemed to be the most prominent ones, i.e. those of timbral and rhythmical complexity. Perhaps formal complexity could have been added because some sound examples, short though they were, possessed contrasts between different patterns allowing segmentation into different sections.

Jay Dowling (1989) has noted the problem that in psychological experiments on music perception, the stimuli are often "skimpy" as compared to real music. Instead of real acoustic instruments, one might use sine tones; instead of letting a musician perform a phrase, it is rendered perfectly quantised on a MIDI sequencer; instead of considering full-length compositions, one isolates short segments. All this is done in the name of testability. There is nothing wrong with this reductionist strategy as such—indeed it is necessary to gain an understanding of all perceptual mechanisms at work as we listen to music—but one may reasonably doubt the extent to which the findings generalise to actual music. The nagging doubt as to the ecological validity of such psychological research always remains (see also the discussion in Section 2.1.2).

The Autonomous Instrument Song Contest, although asking questions about perceived complexity, purposely did not follow the reductionist strategy. Anyway, as we have noted before, generating stimuli that differ in some controllable way with respect

to perceived qualities with feature-feedback systems is not easy. The stimuli (the sound examples) do not change in a predictable way as the synthesis parameters are varied. Shorter sound examples could have been used in order to reduce the influence of formal processes on complexity evaluations, but cutting them shorter would have amounted to presenting skimpy stimuli. Even the 40-second length that was chosen is far too short to fully demonstrate the difference between those sound examples that undergo complicated transient processes finally to land on a static behaviour, and those that are persistently irregular and perhaps exhibit intermittent chaos. And, if we take seriously the goal of rendering full-scale compositions by autonomous instruments, then miniatures of less than a minute are on the verge of cheating. Only with sufficiently long pieces, say a few minutes in duration, do the shortcomings of autonomous instruments come to the fore. The only excuse for not presenting such long sound examples in the test was the concern that the participants would not have the patience to listen to them.

8.2.4 Poietic analysis and reception

Gerald Bennett (1995) once lamented the lack of tools for the analysis of electroacoustic music. This, according to Bennett, has consequences for music education, the quality of electroacoustic compositions, and the status of electroacoustic music. The wealth of analysis methods at the disposal of the analyst who wants to take apart a piece of notated music is formidable. When it comes to orally transmitted music the alternatives are fewer, unless it is possible to transcribe the music into common practice notation. For music that eschews being lattice-based altogether, as much electroacoustic music does, the analysis methods are even fewer. Why would notation be a sine qua non for analysis? Maybe because it permits the analyst to look over the composer's shoulder and to glean insight into the processes that went into assembling a piece. However, analysis may also be about what a listener experiences, in which case there are no greater difficulties in the analysis of electroacoustic music than instrumental music. Using sonograms is one option for fixing the intangible flux of musical sound in a visual format, and recurrence plots is another. Neither sonograms nor recurrence plots, nor any other means of visualisation for that matter, are very helpful when it comes to gaining an understanding of the compositional process that leads to the final piece. The only exceptions are the rare cases when a composer has translated a visual image into a sonogram, such as can be done with Xenakis' UPIC software (Xenakis, 1992). Other examples include a piece by Aphex Twin whose sonogram yields the image of a face at one point (Adams, 2006), and David Dunn's piece *Gradients* (Dunn, 2007).

Stéphane Roy, as mentioned before, has done much to introduce some of the best contributions from the analysis of instrumental music to electroacoustic music (Roy, 2003). In particular, he borrows the concepts *"analyse du niveau neutre"* (analysis at the neutral level), *"analyse poïétique"* (poietic analysis) and *"analyse ésthesique"* (analysis of reception) from Nattiez. In instrumental music, the neutral level consists of the score. The poietic level is that which concerns the making of the composition, everything that can be classified as compositional techniques, whereas the analysis of reception deals with how the music is perceived by an audience. The neutral level is a problematic concept already when applied to scores; for example, it is not clear how to categorise all the

detailed knowledge of performance practice that cannot be found in the score but which guides any performance of it. Applying the concept of a neutral level to an electroacoustic composition seems even more fraught with difficulties. Roy proposes that a preliminary segmentation of the work is what constitutes this analysis at the neutral level. However, must one not then begin with what is perceived when listening from beginning to end, that is, by performing an analysis of reception? Alternatively, if one knows anything about the generative paths that have led to the final form of the piece, the analysis could be informed by these known facts about the poietic level.

These difficulties with the concept of analysis at the neutral level are acknowledged by Roy (2003, pp. 80–82). Quoting from Nattiez, we are told that "neutral" signifies that the poietic and receptive dimensions have been neutralised; analysis at the neutral level is obtained by pursuing to its ultimate consequences any chosen analysis method (Roy, 2003, p. 80–81).

Although Roy is most concerned with developing a theory of reception analysis, he makes some interesting remarks about the poietic aspects as well. It must be remembered that analysis of music that exists as written scores has a privileged access to the symbolic level of notes, and that a vast range of compositional techniques have been developed to manipulate this symbolic level. Although operations such as the temporal augmentation, transposition, retrograde or inversion of a theme involves an implicit translation from notes to integer numbers and back, this operation is transparent to any music analyst and sometimes to listeners as well. Electroacoustic music in general does not possess a comparable symbolic level, even though there are some notable exceptions such as Wendy Carlos' Bach interpretations and later experiments with various unconventional tuning systems (but as Landy (2002) has noted, most specialists of electroacoustic music would not consider *Switched on Bach* by Carlos as genuinely belonging to any category of electroacoustic music). In those cases, the lack of scores still causes practical problems for the analysis of note level structures, although advanced multipitch extraction methods could come to the rescue.

As Roy (2003, p. 329 ff.) notes, operations such as transposition, augmentation and inversion are all invertible. Precisely because of their invertibility a listener may be able to hear the transformed phrase as related to the original. This is not to say that non-invertible operations do not occur in notated music. However, the fact that an operation is invertible is no guarantee that the transformed phrase will be recognised as being related to its original—cf. some of the more drastic but invertible motif transformations considered in Section 7.5.3. Roy contends that it is the *systematic* character of such transformations in notated music that makes it possible to demonstrate that a transformed motif has a certain kinship with the original object.

> En effet, de nombreuses transformations mélodiques, rhythmiques et de durées en musique tonale font appel à des algorithmes, des opérations systématiques à caractère mathématique dont le principe de fonctionnement, une fois découvert, peut être appliqué de manière inverse et systématique à l'unité transformée afin de retrouver l'unité originale; c'est précicément ce que nous entendons par réversibilité. Or, la musique électroacoustique est apparemment dépourvue de ce type de transformations, si l'on se place du point de vue de

> l'esthésique (Roy, 2003, pp. 329–330).
>
> [In effect, numerous melodic, rhythmic, and temporal transformations in tonal music requires algorithms, systematic mathematical operations whose function, once discovered, may be applied systematically in reverse to the transformed unit in order to recover the original unit; that is precisely what we mean by reversibility. However, if one takes the esthesic viewpoint, electroacoustic music is apparently lacking such transformations.]

No matter how fascinating the algorithmic procedures that the composer has set up for controlling various synthesis or effect parameters may be, if they are not audible to the listener as structures perceptually related to each other, their relationship will be lost. Musical examples can be drawn from the early days of electronische Musik from the Cologne studio, where serialist principles were applied to the organisation of sonic parameters. In fact, Dahlhause had criticised these composers for trying to apply serialist techniques to the the composition of sound, which led to unified percepts as timbre instead of as composite chords, as one would suppose (Dahlhaus, 2005).

The fact that it is so hard to trace the composerly procedures in the reverse direction in acousmatic or electroacoustic music does not mean that compositional techniques resembling those of instrumental music cannot be used or are not used.

> Bien entendu, il est possible et même fréquent qu'au niveau des opérations poïétiques, le compositeur de musique électroacoustique ait recours à des algorithmes de transformation que favorisent la quantification, par exemple, des paramètres d'un filtre (en Hertz) ou d'un contrôleur d'intensité (en décibel). Or, si de telles transformations algorithmiques ne sont pas audibles (pour l'analyste comme pour l'auditeur), elles ne peuvent constituer une donnée valable pour l'ANN [l'analyse de niveau neutre] ni une donnée pertinente pour l'analyse esthésique portant sur les relations d'équivalence (Roy, 2003, footnote, p. 337).
>
> [It is quite possible, and even common, that on the level of poietic operations, the electroacoustic composer takes recourse to algorithmic transformations that favour quantification, for example of filter parameters (in Herz) or the control of intensity (in decibel). Nevertheless, if such algorithmic transformations are not perceptible (to the analyst as well as to the listener), they can constitute neither a valid given for the analysis at the neutral level nor a pertinent given for the analysis of reception bearing on the equivalence relations.]

In conclusion, the poietic analysis of electroacoustic music is often difficult, because it presupposes access to information that is usually not extractable from listening to the piece alone. If we are lucky, there are composer's sketches, notebooks, programme notes and other documents to consult. Another point worth mentioning is that the pieces considered by Roy (most of his book is devoted to F. Dhomont's *Points de Fuite*, but a few other works get a brief mention) might be classified as typical examples of acousmatic music. Indeed, there is no analysis of live-electronic works, or anything that resembles self-organisation. Nevertheless, if one wanted, any piece of music could be analysed by

following some of the strategies outlined by Roy, including music made with autonomous or semi-autonomous systems. As long as the analysis is about reception, this should be unproblematic. As for the poietic analysis of self-organised sound, one cannot generally rely on traditional notions of compositional techniques. Here, the computer programme that generated the music is the text to consult. To quote Trevor Wishart, even though he has always stressed the importance of listening as a means of evaluation or analysis:

> We might in fact argue that the truly potent texts of our times are certainly not texts like this book, or even true scientific theories, but computer programs themselves. [...] For the text of a computer program can act on the world through associated electronic and mechanical hardware, to make the world anew, and in particular to create new and unheard sonic experiences (Wishart, 1994, p. 3).

Actually, there are examples of computer music published as source code. The Csound book (Boulanger, 2000) came with a few compositions that anyone who has a computer with Csound installed can render. More recent examples are the call for SuperCollider users to write pieces specified by at most 140 characters of code[1] and the previously mentioned one-liners of bytebeat (see Section 5.2.4). Obviously such short programmes draw heavily on included subroutines; counting all of them would render the programmes considerably longer. In these cases, the poietic level is even more accessible than in common practice notated music. Instead of having to figure out what the motives are and what transformations have been applied to them (as one would in an analysis of thematic development or when following permutations of twelve-tone series) one has direct access to the operations that produce the signal. The situation is so transparent that it is almost an exaggeration to speak of analysis; all the tricks are given away for free, and if there is anything hidden it resides not in the code but somewhere else. Or perhaps we should rather say that the analysis becomes an act of "programme language hermeneutics". Understanding what someone else's code does (or understanding one's own for that matter) is not always easy. The situation where the source code is available for inspection is what Collins (2008a) calls "white box testing", in opposition to black box testing where several different outputs of a generative music programme are available for analysis but not the source code itself. There are some difficult challenges in the analysis of generative music that bear some similarity to the problem of investigating parameter spaces of feature-feedback systems.

8.2.5 Generative music

To summarise the main ideas of generative music, it is used for making large-scale compositions by simple means or compositions of limited duration that manifest themselves differently each time they are realised. For the most part, generative music is algorithmically generated, thus there is a link to autonomous instruments, although generative music may sometimes involve live interaction.

[1] http://supercollider.sourceforge.net/sc140/

Many conceptions of generative music exist. For Brian Eno, one of its proponents, generative music seems to involve a certain economy on the production side that nevertheless leads to a wealth of musical output (Eno, 1996). Eno mentions Terry Riley's *In C* and Steve Reich's *It's Gonna Rain* as two early sources of inspiration. *In C* with its flexible instrumentation (for any number of musicians) consists of 52 bars that may be repeated by each individual performer as many times as he or she wants before proceeding to the next bar. This freedom of interpretation yields unpredictable combinations of musical patterns, especially in the middle of a performance where the musicians are likely to play from different parts of the score. *It's Gonna Rain*, on the other hand, is simply made of tape loops that phase out against each other. Here, the economy of material stands in contrast to the ever-changing combinations of the tape loops, and in addition the listener's attention inevitably wanders around as it gradually discovers new aspects in the short looped sound fragment. Eno's *Music for Airports* (1978) builds on similar principles of tape loops that phase out. Although the pieces released on record are short enough to fit on one side of an LP, the loops are of incommensurate lengths and may go on endlessly without repeating.

Eno also acknowledges two other, very different sources of inspiration for his generative music. One of them is the two-dimensional cellular automaton known as the *Game of Life*, and the other is a computer screen saver. Drawing on these new inspirations, Eno has released generative music of another kind, one that exists in the form of a computer programme that always outputs different versions of the music. Thus, some of the most important aspects of generative music can be found in Eno's work: using incommensurate length loops that phase out to generate ever-changing constellations, with a high degree of economy of the material in relation to the potential length of the music; and using computer programmes that generate new versions of the same composition each time they are run.

Generative music differs from thoroughly-composed music in that one cannot know in detail how it will appear in a given performance or at any particular moment; it also differs from improvisation by being determined by the generative mechanism. There is, however, some conceptual overlapping with music in open form with a set of instructions for the performer to follow. Now, let us consider how the ideas of generative music might apply to autonomous instruments, and to feature-feedback systems in particular.

In fact, autonomous instruments are germane to generative music systems. Their use may vary from setting up a system that will run as a sound installation for a very long time to realising short compositions. In the latter case, the autonomous instrument might just be used as a one-off composed instrument for the generation of a single fixed form composition, but then the generative music aspect is totally lost. However, why fix a single version of the piece, if a whole set of related pieces may be as easy to generate? The idea of generative music saves us the trouble of deciding in favour of one particular take, or of one parameter setting for an autonomous instrument. This decision, being somewhat arbitrary, is replaced with a family of potential realisations of the piece, each activated a single time.

Generative music systems that create new realisations of the piece on each run may depend only on a random seed that provides an initial condition or parameter value for the system (Collins, 2008a). A feature-feedback system could conceivably be started from

a randomly chosen initial point in its state space. This, however, usually means that the dynamics approaches some attractor; it is even quite likely to approach one and the same attractor for a range of initial conditions. Usually this will imply that the generated sound will be rather similar, except for an initial transient which may differ depending on the initial condition. A second alternative would be to randomise some parameter constant, which could have a more profound impact on the dynamics. These two ways of producing a novel output of the same feature-feedback system appear to be worth trying, although neither come without problems.

According to Nick Collins, "... the design of generative music is of an order of magnitude harder than making fixed products" (Collins, 2003, p. 71). This is so if one considers all the possible outcomes that may ever occur when running a generative music algorithm. Only with very simple generative systems will one be able to appreciate the range of potential output without actually listening to it all.

The audience may not experience more than a single performance of the generative piece, or if it runs uninterrupted for a very long time they may hear only a brief moment of it. What, then, is the rationale for engaging with the uncertainties of generative music making? Collins (2003, p. 72) lists three situations where the generative approach is essential:

- There is not time or space to store the entire composition, as may be the case in installation work or if a large number of alternative pieces should be made.
- There is a "tenuous hidden conceptual thrill that the system is running during listening".
- The generative system allows some kind of interaction, or involves time- and site-dependence.

Live performances with musicians are in a very obvious way different from listening to the same music in recorded form. If the thrill and the focus on the brittleness of live performance is considered important, then that may also be an argument for running generative music live. When the audience is supposed to engage interactively with the generative system, it must be equipped with the necessary means for reacting selectively to different actions. In such cases, the music cannot be fixed but must adapt to circumstances. The first point, however, may appear to be the least well-founded. Given the huge storage space on current hard disks, a lot of audio can be stored. Of course there are limits that may be reached if an installation is supposed to run for weeks or even years with ever-changing output. The old (analogue era) solution to this problem was to run several tape loops of different lengths that phased out against each other, thus producing constant variations of sound combinations. Certainly there are conceptual motives for wanting to generate such constantly varying, never repeating music; nonetheless, one may wonder what impact it has on the single time listener who only hears a brief snapshot of the infinite process.

In the phasing tape loop paradigm of generative music, one may reasonably get a sense of all possible clashes and collisions between the juxtaposed sound layers. With algorithmically generated material it may be much harder to predict the future output if the algorithm is stochastic or deterministically chaotic and sufficiently complicated.

This is the case for our class of autonomous instruments, the feature-feedback systems, of which one can only know for sure what they will do if they are deterministic and have already been tested with the same parameters. In this sense, Collins is absolutely right about the difficulty of composing generative music. It is much harder than writing thorough-composed music existing in one single version, at least as long as one considers every potential output of the generative system and takes the trouble to ensure that all are acceptable versions. However, are not the ambitions simultaneously slacked? In single-version composition, the piece is usually written to stand repeated listenings, despite the sad fact that most contemporary music gets at most one performance—one is supposedly not going to get tired of it after hearing it a few times. Arguably, the compositional strategy necessary for generative music detracts attention from the details and form of a single realisation. Perfectioning a miniature is the opposite extreme; a similar devotion to fine detail seems unidiomatic in generative music.

In a curious way, generative music harks back to an era before fixed recordings became the usual way to experience music. The same work will never be the same when repeated. Part of the reason why our centuries old literature of Western classical and early music still survives is probably that it remains open to new interpretations. Thus, it is not surprising to see continued interest in novel interpretations of open electroacoustic works, such as *Scambi* by Pousseur.

8.2.6 Devil's Music

As an example of generative music, let us consider *Devil's Music* by Nicolas Collins (not to be confused with Nick Collins, whom we have quoted above). *Devil's Music* uses radio broadcasts as its input source. The current version is basically a sampler with three tracks and some simple processing capabilities such as changing playback speed. An interactive version comes with the CD (Collins, 2009) in the form of a MAX/MSP patch that can be run on a computer with a radio receiver plugged into its audio input. The three tracks may be sampled to by holding down a button for up to ten seconds. As soon as the tracks contain audio material, they start playing. Either they play the material in a loop, or in a stuttering fashion, or forwards and backwards similar to scratching; these playback modes can be set by the performer.

Devil's Music strikes an interesting balance between recognisable form and novelty, between being fixed and yielding to the performer's expressive style and the contingencies of the situation. There are strong constraints that ensure that the output will be recognisable as a version of *Devil's Music*, despite the fact that the input from the radio will never be the same, and despite the possible variations of performing styles. Collins specifically gives the instruction that the input be taken from FM, AM, or shortwave radio; sources other than radio should thus not be used. Using radio broadcasts, the performer may premonitor the radio channels and choose what seems fit to the mix. One might suppose that this opens up the possible versions so much that they might sound almost any way; but using radio input means sampling from an ever-changing stream of sound, and as a result *Devil's Music* always has a peculiar formal property of going from one sampled fragment to the next, never returning to material that was present earlier in the piece, as the buffers are erased and supplanted by new material. An important factor

in shaping the sound of the piece is the length of samples. One can sample extremely brief segments that result in pitched drones, and up to ten seconds which lets the material come through clearly recognisable.

A simple adaptive mechanism is used for controlling when to reset the sample playback. The incoming radio signal is analysed by a peak tracker, and whenever the peak strength is sufficiently high a resetting is triggered. Thus, there is a very simple feature extractor in the system. The piece is a good example of generative music, of an open work in open form; the MAX patch arguably is a semi-autonomous instrument. The performer can influence the result to some extent, while its sound source of radio broadcasts is always beyond the performers control.

Turning it into an autonomous instrument is conceivable, but perhaps not very interesting. To do so, one could route the signal output to the input, after having initialised the sample buffers with some radio material. This, however, is clearly against Collin's artistic intentions. Devil's Music relies heavily on the accessibility of a stream of fresh input, which makes it fundamentally different from the design principle behind autonomous instruments.

8.2.7 George Lewis on improvisation

Machine listening and semi-autonomous instruments evidently thrive in contexts where improvisation is the predominant means of music making. After all, it is pointless to compose rigidly specified scores when part of the performing ensemble will be a semi-autonomous instrument whose exact actions are not foreseeable, and which may depend on contingencies outside the musician's control. This is in contrast to works for soloists and live electronics where score following is used to keep track of where the musician is in relation to the timeline of the score (Orio et al., 2003). Other options include some amount of openness and improvisation.

George Lewis has often made the point that terms such as interactive computer music, open form, or happening have regularly been used to mask the fact that what is really going on is improvisation (Lewis, 2009). The blame is unhesitatingly put on "pan-European contemporary music's widespread disavowal of improvisation" (Lewis, 2009, p. 459), which is not entirely wrong, but overstated. More nuances exist than simply improvised versus written music. The open work possessing a recognisable kernel that remains intact through strongly differing realisations is a case in point. As previously mentioned, when musicians with classical training were asked to improvise parts of the material in aleatoric compositions in the 1950s, many felt uncomfortable doing so and resorted to writing out fixed versions that worked well (Dahlhaus, 2005). Today, however, and increasingly over recent decades, many musicians who specialise in performing contemporary music show less reluctance towards improvisation.

Lewis himself began to make interactive music early on with the KIM microcomputer. In an interview, Lewis pointed to the work of David Behrman and the League of Automatic Composers who had a few microcomputers interconnected: "It sounded a lot like a band of improvising musicians. You could hear the communication between the machines as they would start, stop, and change musical direction. Each program had its own way of playing. [...] I felt like playing, too, to see whether I could understand what these

machines were saying" (Roads, 1985, p. 79). There is a piece called *The Kim and I* from 1979 featuring the computer playing a kind of walking bass, with Lewis as the trombone soloist on top of it. However, it is the system and composition known as *Voyager* for which Lewis is most recognised as a computer musician (Lewis, 1999, 2000). The *Voyager* system grew out of the early work with microcomputers. The musical inventiveness of the group of musicians working with microcomputers, such as those involved in the League of Automatic Composers (later to become The Hub), has often been underestimated. The microcomputers were much less powerful than the mainframes used in pioneering computer music research even in the 1960's, and so were the synthesis techniques. If the history of electronic music is written with a predominantly technological focus (such as Manning, 1993), then there is a tendency to overlook such musically important inventions as those that arose in milieus where relatively simple technology was used in novel ways.

The computer system in *Voyager* uses feature extraction on the incoming signals from one or two musicians, but may also run without input. Thus, it actually qualifies as an autonomous instrument, although it seems only to have been documented as an improvising partner to human soloists. The output consists of several instrumental voices of sampled sounds. The feature extraction consists of a pitch follower, "a device known to exercise its own creative options from time to time" (Lewis, 1999, p. 103); furthermore volume, velocity, sounding duration, inter-onset duration, register, and ranges of these features and their averages over time are used as input to the programme. The output depends not only on this input, but also on a white noise random process. The system has several options for how to react to input, from imitating to opposing or ignoring what the musician plays. This flexibility of response seems to be crucial for the character of *Voyager*. A system that only imitated what the musician presents to it, or that always returned contrasting material, or that just shut its ears and developed its own monologues, would hardly be interesting to play with. In that respect, in designing a successful interactive music system one has to consider criteria similar to those that make a fellow human improviser good or bad company.

8.2.8 Mumma's Hornpipe

Gordon Mumma was a member of the Sonic Arts Union, together with Robert Ashley, David Behrman and Alvin Lucier. Together they performed their music involving live electronics from the late 1960s to the end of the 1970s. With the four member's distinct aesthetics, they never solidified into a group. Mumma also collaborated with David Tudor and John Cage, and occasionally built electronic circuits for their pieces. There were collaborations with the choreographer Merce Cunningham and with many others. Mumma invested much of his artistic output in collaborations—even with the audience as in *Cybersonic Cantilevers* (1973). He built his own studio and emphasised the practical matters of its construction, from the choice of economically affordable and customisable electronic devices to the ergonomics of having the whole setup within arm's-reach (Mumma, 1964). This customisation is taken a step further in the works involving live-electronic processing.

In a number of pieces, Mumma used what he called *cybersonics*, or custom-built circuits for the analysis and processing of sound. *Hornpipe* (1967) was not the first of

these pieces, but is the one that most clearly serves as an early example of adaptive live-electronic processing and a semi-autonomous instrument. The cybersonic processing had already been used in pieces such as *Medium Size Mograph* (1963) for piano. A brief analysis of *Hornpipe* will illustrate some of the difficulties that we face when trying to analyse music made with semi-autonomous instruments.

Hornpipe begins as a horn solo, with the soloist moving around in the performance space. Mumma's own comments describe what is going on behind the scene, in the electronics:

> The cybersonic console monitors the resonances of the horn in the performance space and adjusts its electronic circuits to complement these resonances. During the first part of *Hornpipe*, none of these electronic activities are heard—they are internal processes of the cybersonic console. During these adjustments, certain circuits become unbalanced and attempt to rebalance themselves. While rebalancing, various circuit-combinations occur that produce complex electronic sound-responses. [...]
>
> In each performance the player learns from his own choices and their corresponding electronic responses, which sounds are most likely to unbalance and rebalance the cybersonic console. *Hornpipe* ends when a sustained horn sound balances all of the cybersonic circuits and terminates the electronic sounds (Mumma, 2002, liner notes).

In the current recording (probably the only one available), the piece lasts just over 15 minutes (see Figure 8.1). The cybersonic responses are first heard about one third into the piece. It begins at a slow pace with long, steady tones, sometimes with a two-note motif of (approximately) an ascending minor seventh. If it were not for the recurrence of certain pitches and the two-note motif, the opening would be very fragmentary. Gradually there is an increasing diversity of material, with timbral contrasts of open and stopped tones, and most notably sounds produced by playing the horn with a reed. This playing technique allows the horn to sound like a tenor or baritone saxophone, producing a coarse sound and even multiphonics. The reed tones are juxtaposed with ordinary horn sounds, often in short bursts and in rapid succession.

Amidst the ever-increasing variety of timbre, it is not entirely evident when the cybersonic responses and processing begins. There is a very loud low-pitched note at 6'35 – 6'45, which is most likely an electronically processed sound, at the end of which the room's resonance becomes audible in a new way. However, even before that, there are sounds that could be ring-modulated horn tones. The system as a whole is capable of fast changes between pure, unprocessed, horn tones and cybersonic responses, as can be heard in one of the rare rapid phrases, at about 7'20. Smooth transitions are also in the range of possibilities, as the one starting as a horn note on B4 at 7'35, with electronic sound entering at the same pitch, but with a gradually widening and slightly inharmonic spectrum. On several occasions the cybersonic response consists of a scintillating but rather static tone that suddenly stops as though of its own accord. There is no clue in the recording that would reveal whether it is the performer who causes these endings or if it is due to some internal process of the cybersonic circuit.

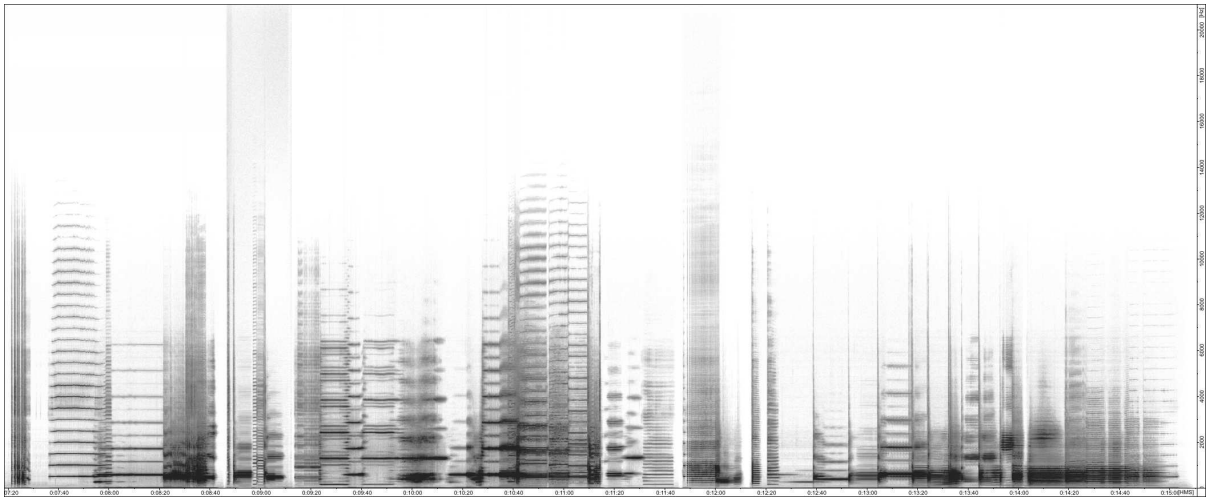

Figure 8.1: Hornpipe, sonogram of the final half of the piece.

There is little to be learnt from the composer's notes that accompany the CD about the processing techniques used. More details can be found from another source (Mumma, 1967). There is a mechanical apparatus for the analysis of resonances in the performance space, which is built of vertical pipes, each containing a microphone and tuned to different resonance frequencies. The pipes are placed behind the performer, and behind the pipes there is a loudspeaker that distributes the sound into the room. As an account of the technical setup for Hornpipe, the following is more evocative than precise:

> The acoustical feedback loop which exists between the French Horn, the resonant pipes, and the loudspeaker, is part of an electronic feedback system which employs amplitude gated frequency translation.
>
> As the performance begins the system is balanced. Sound is produced only when something in the acoustic-electronic feedback-loop system is unbalanced. The initial sounds produced by the French Hornist unbalance parts of the system, some of which rebalance themselves and unbalance other parts of the system. The performer's task is to balance and unbalance the right thing at the right time, in the proper sequence (Mumma, 1967).

Nothing is said about what is the "right thing" and the "right time"; it may well be decisions that are left to the performer's discretion. Neither do we get to know precisely how the system, or what parts of it, become unbalanced. It can be assumed, however, that the unbalancing is caused by feedback, although the electronic processing goes on tacitly in the first third of the piece, ruling out acoustic feedback. Amplitude gating is mentioned as one of the processing techniques, and indeed, it is clearly notable when the cybersonic responses begin or end. Frequency translation, also known as single sideband modulation (see Section 3.3.1), would explain the inharmonic timbre pervading most of the electronically processed sound.

It is a common problem for all music that depends heavily on technology that, as time passes, the setup on which the music is performed becomes obsolete. Documentation may

be sparse, and only related to the specific technology with which the work was realised. There are photos from performances of Hornpipe, showing Mumma with the horn and cybersonic console (Nyman, 1999; Mumma, 2002). The console is simply a metal box with some knobs and cables, carried by the performer. That is not a likely setup to be reconstructed if anyone else would consider performing *Hornpipe* again in the future. We have noted the need for careful documentation of the technicalities for the purpose of poietic analysis, but it will also help in reconstructions of historical electronic music using obsolete systems.

8.2.9 The cybernetic legacy

Since Norbert Wiener introduced the interdisciplinary study of cybernetics in the 1940s (Wiener, 1961) and Shannon introduced the theory of communication which touched upon some related issues (Shannon, 1948), their ideas were also gradually picked up in parts of the contemporary music community. The Swedish composer Lars-Gunnar Bodin made a number of pieces in the 1960s with clear references to cybernetics, science-fiction and parapsychology, including *CYBO II* (1967) and *Semicolon; seance 4* (1965). In Bodin's works, the connotations of cybernetics are mainly conveyed by spoken texts. This is of course entirely different from using principles learnt from cybernetics as organising principles for the composition, as Mumma did. In Russia, cybernetic theories were also applied to music, although there it meant algorithmic composition applied to style imitation (Zaripov, 1969).

Roland Kayn (1933–2011) made many large scale compositions and often hinted at cybernetic principles[2]. His work is not very well known today, and even less is known about how he applied cybernetic ideas to his compositions. Many of his compositions are extremely long, in some cases spanning several hours. Given his enormous output of very long compositions, one may speculate that a good deal of automation of some sort is going on. A stylistic trait seems to be the frequent use of slow-paced or static textures with little small scale detail. Should his music be qualified as being made by autonomous instruments? Quite possibly so, since Kayn (1996) mentions feedback and autonomous processes, the importance of serendipitous discovery, the absence of external control, and the composer losing his original function. More detailed descriptions are not given.

The contributions of the cybernetic movement have largely been absorbed into other disciplines such as artificial intelligence and complex adaptive systems, although a few researchers have continued to refer to cybernetics as their discipline (Heylighen and Joslyn, 2001). However, among composers whose work include semi-autonomous instruments or feedback systems, references to cybernetics are still frequent (e.g. Eldridge, 2008; Bökesoy, 2007).

Mumma's technical setup for *Hornpipe* exemplifies the use of feedback in a way that has then become paradigmatic. The notion of *composed instruments* was introduced in

[2]The primary source to information about Kayn is his website: http://www.kayn.nl/
Original recordings of his music are collector's items. Little, if anything, can be found about Kayn in books about electroacoustic music. Manning (1993) only mentions his collaboration as an improviser with Gruppo di Improvvisazione Nuova Consonanza in the 1960s.

the first chapter (see Section 1.4.3). As it turns out, this concept is already accurately captured in Mumma's own description:

> My own electronic music equipment is designed as part of the process of composing my music. I am really like the composer who builds his own instruments, though most of my "instruments" are inseparable from the compositions themselves. (Mumma, 1967)

On the first page of the score for *Medium Size Mograph* for piano and electronics, there is a circuit diagram detailing the electronic setup to be used (Mumma, 2002). Although in this case there is a traditional score with performance instructions, in an extreme case a composition could be defined simply as the output of the composed instrument, whatever happens to occur. There may well be pragmatic reasons for not fixing every detail in a score when using cybersonic or semi-autonomous instruments. If the instrument is designed to respond in unpredictable ways, the performance cannot easily accommodate its varied behaviour if restricted to follow a fixed notation. Improvisation in some form may then be the best solution.

Intermediate positions are conceivable between improvisation and a detailed score with exact notations of every action as it occurs through time. The performer may be provided with a set of rules to follow; these rules may state what to do in certain circumstances as they arise in the musical output of the cybersonic or semi-autonomous instrument. This is similar to treating the human performer as an actor whose role it is to carry out an algorithm.

8.3 Composers and algorithms

The creative aspects of composing algorithmic music have not been much studied. There has even been a tendency to not think of music realised by computer-assisted algorithmic composition as being the creative product of the composer. Correspondingly, sometimes one finds the misconception that computers used for algorithmic composition are somehow exposing creativity. We will try to demonstrate the problems with these ideas here.

According to Nierhaus (2010), there is a predominance of publications on applications of algorithmic composition for style imitation rather than genuine composition. This imbalance, he explains, is due to the fact that academic researchers publish papers as a natural part of their work, whereas composers primarily publish their music, and only occasionally document their algorithms in papers. A more plausible reason for why academic research has concentrated on style imitation is that the successful imitation of a given style is a clear criterion for evaluation of the gain in scientific understanding. Evaluations of experimental algorithmic techniques aimed at genuine composition should rather apply artistic criteria. Our feature-feedback systems, regarded as methods of algorithmic composition, are at a safe distance from any kind of style imitation. As for their novelty, no respondent to the Autonomous Instrument Song Contest expressly stated that they had never heard anything similar before; on the contrary, one respondent whom we have previously quoted said that such things had already been heard "thousands of times".

The strategy of delegating substantial parts of the compositional decision-making to an automated process appeals more to some composers than to others. Therefore, one should not expect autonomous instruments to be of immediate interest as tools for all composers. In the following, some studies are reviewed that address the issues of personal traits among composers with a bearing on their approach to algorithmic composition.

8.3.1 Composed by machine

Who is really responsible for a composition that is the result of running an algorithm on a computer—could it be the computer alone? There is not much that speaks in favour of this view; after all, a human programmer must have come up with the algorithm in the first place. In fact, nowadays this discussion does not often take place. A more subtle variant of the problem comes up with David Cope's compositions realised with his programme Experiments in Musical Intelligence (EMI). Cope's own work with EMI has, however, met with much criticism, and judging from his reply (Cope, 1999), some of his critics seem to have a point.

EMI, as mentioned in the first chapter (Section 1.4.5), composes music in the styles of other composers. It uses a database of musical works from which it discovers generative rules, and from these rules new compositions complying with the analysed style may be generated. When a performance was given with human performers, a critic argued that the music got its qualities precisely from the human interpreters. Therefore Cope chose to record it with a MIDI-controlled Disklavier, but then another critic found it hard to judge the success of the work because of the lack of human interpretation (which seems to be a reasonable point to make). The programme may be able to output music that is recognisably similar to Mozart or Mahler, but is it as good as the original? Cope hopes that one would listen to the music on its own terms, but for a knowledgeable listener, it is inevitable that comparisons will be made with the stylistic source, the original human composer.

Cope argues that, like chess-playing computers, EMI supports rather than undermines anthropocentric notions about creativity:

> For as with Deep Blue, humans designed and built the computers on which the program runs; a human coded the program which produces the music; humans composed the music that the program uses as a database; and, possibly most importantly, humans listen and evaluate the output. Yet these facts seem lost amid our deep-set fears of being out-classed by machines (Cope, 1999, p. 81).

A more serious objection is that this programme and its musical output challenges ingrained beliefs about the role of originality and perhaps even genius in human creativity. The output of EMI is not exactly a musical analogy of counterfeit in painting, since we are not misled into believing it is some newly discovered piece by an old master, but it *is* a pastiche. Originality has not always had the same status as an artistic criterion in music history, a point often made by Dahlhaus (Dahlhaus, 1983, 1992). Before the eighteenth century, originality did not play an important role, and maybe we are today witnessing a decline in its importance. Our musical repertoire as performed at the concert houses is replete with the old masters—surely we do not need yet another epigone, one may

argue. Furthermore, the utility of EMI and its output is restricted to the light it sheds on questions of style, creativity, listener expectations and so on; it is useful mainly as a tool for the musicologist and less so for the composer. To be fair, it must be added that Cope actually has used his own handcrafted compositions as input to the programme, thus letting it imitate his own style. Similar computer programmes may be of greater utility to other composers if they provide not complete compositions but rather only fragmentary musical material upon which the composers can elaborate.

We are presently so accustomed to wonders such as EMI and other advances in artificial intelligence applied to music that it is hard to be visionary these days. It can then be a healthy reminder to quote Zaripov, who speculated about the future of algorithmic composition by computers in 1960s Russia. That computers would eventually be able to do all that EMI does today was taken for granted by Zaripov (1969, p. 149):

> What will there be after that? Will research into the field of discovering the "secrets" and "mysteries" of creativity then cease, will the development and perfecting of the composer-machine then come to a halt?

This is not what Zaripov thinks, of course. Rather, he imagines that style imitation will be supplanted by extrapolation of existing styles in the computer.

> In other words, it will be able [...] to create a new style, different from those already known and studied, i.e., it will anticipate the style of future composers (Zaripov, 1969, p. 149).

The extrapolation of extant styles or the anticipation of novel styles are apparently not active research frontiers in algorithmic composition, but they *could* be. We think, however, that new styles are better approached head on, as we have done with the autonomous instruments.

8.3.2 Computer psychology

Few would say that a microphone listens—maybe that it hears. Yet "machine listening" is innocently used as the term for the computer analysis of sound. Is this an example of an anthropomorphic tendency to bestow computers with mental faculties?

The Turing test has often been applied and misapplied to evaluate computers' capacity to behave as if intelligent. Turing's test involves a human and a computer who interact with an interrogator through a text medium. The interrogator has to judge its two interlocutors and try to find out which one is human and which is the computer. A computer that, on average over several trials, deceives its interrogators is said to have passed the Turing test. Christopher Ariza (2009) surveys several examples where the Turing test has wrongly been claimed to be involved in musical evaluations of computer output, or musical interaction with computers. According to Ariza, the error committed is that the medium of the Turing test is language, and trying to transfer the test to artistic output does not make sense. "Use of the [Turing test] in the evaluation of generative music systems is superfluous and potentially misleading; its evocation is an appeal to a measure of some form of artificial thought, yet, in the context of music, it provides no more than a listener survey" (Ariza, 2009, p. 49). Also, the Turing test must involve interaction with

the system; it is possible to generate texts by computer that can fool anyone to believe they were written by a human.

To give an example of how tempting it may be to allude to the Turing test in musical contexts, consider the following reply to the Autonomous Instrument Song Contest. To the question whether there were any odd examples, one participant replied:

"B3. Who was the performer?"

It should be said that the purpose of the Autonomous Instrument Song Contest was by no means to test the generative system's capability of illuding human agency, and many respondents correctly pointed out the synthetic and algorithmic character of the sound examples even though not explicitly asked about it.

Ariza mentions that of all his surveyed musical Turing tests, each of them reported success for the machine. This includes instances of algorithmic composition with style imitation using, for example, Markov chains. There is good reason to be more modest about such results that come about from algorithmic composition based on the input of ready-made music or sophisticated music-theoretic constructs. One should not be fooled by the hype into thinking that "the machine made this". If there is input in the form of note transition probabilities taken from analyses of a corpus of folk songs, then the collective human achievement of composing all those folk songs must count at least as much as the computer's reshuffling of them.

An alternative, more stringent test called the Lovelace test (after Ada Lovelace) is also reviewed by Ariza (2009, p. 53). The Lovelace test requires that the computer be creative. The demands on this creativity are restrictive; the machine must be able to produce an artefact "through a procedure that cannot be explained by the creator (or a creator-peer) of the machine." It is certainly not sufficient that the system at times yields surprising results. If one is able, given time and patience, to trace the result back to the system's architecture, then it has failed to pass the Lovelace test. Supposedly, this back-tracing should be understood as being feasible in principle rather than in practice. This much is evident considering the difficulties one faces when trying to debug large programmes.

The question of computer creativity or otherwise has a bearing on how we regard music that comes out of computers. As Ariza argues, it would take some kind of autonomy and intention for a computer to be creative. It cannot suffice that a human programmer feeds it with code and musical knowledge; the incitement for making music would have to come from the machine, and this is, presently, science fiction. Yet, even if the computer lacks autonomy and intentionality, it can "produce output that appears intentional" (Ariza, 2009, p. 64). Our willingness to imagine direct human agency behind algorithmic music is another question, related to listeners' expectations.

Another view on "machine intelligence" is given by Cochrane (2010). For machine intelligence, there has not been any standard measure of intelligence similar to IQ for humans. Cochrane suggested such a measure which takes four aspects into account. These are the processing capability, memory, actuators and sensors. The prevailing view has been to consider only memory and processing, but a system with no input from the outside world cannot be said to possess intelligence, and a system that has no way of communicating its internal states to the rest of the world is useless for all practical purposes. In this sense, a computer coupled to advanced sensors and actuators could

have a comparatively high machine intelligence, a fact worth considering in machine musicianship. This is of course something quite different from passing the Turing test.

8.3.3 The safety nets of algorithmic composition

Instrumental music as a medium for algorithmic composition imposes quite different working conditions on the composer than electronic media do. There are interesting differences as to what can be done (or is typically done) to the material as it comes out of the algorithm in these two cases.

> In its most extreme form, automated composition resembles a form of "found art"; the composer selects the output and, in effect, signs and frames the work with a title and a performance medium (Roads, 1996, p. 845).

This is not too different from using raw field recordings in soundscape composition, although algorithms have another kind of flexibility. As Roads also points out, there are ways around the fixity of algorithms: the composer may modify the programme that generates the output, or if a transcription phase is included in the process, the numerical output can be treated quite selectively when it is transcribed into a standard musical score. The algorithmic composition of instrumental music has an advantage in this respect: the necessary translation into readable notation offers an eminent opportunity to break with the rigidity of the algorithm. The selection and criticism of material finds a natural place in this gap. Thoroughly computer-generated algorithmic sound synthesis does not provide any similar breaking point in the chain from code writing to sound production, although the sound file that comes out of the algorithm may be the point of departure for a series of new interventions with the material.

In algorithmic composition aimed at performance by musicians, the translation into musical notation and the subsequent interpretation by musicians provides a safety net where the most unrealistic or unmusical parts of the algorithmic output can be filtered out. It may appear a welcome convenience to obtain the output of the algorithmic composition programme directly in musical notation (or MIDI) so as to skip the tedious step of translation. Likewise, if one is worried about distortions introduced by musicians' interpretations and human limitations, it may be tempting to skip this step and send the composing algorithm's output directly to a synthesiser for precise translation into sound, or, as Nancarrow did, punch holes in a roll for a player piano. In our context of autonomous instruments, the algorithm of course does not work on a symbolic note level (with the exception discussed in the previous chapter), but generates all levels of the music without any clear hierarchical separation. In that situation, neither of the two safety nets are present; thus the opportunities for things to go wrong are increased. It has often been noted that in electroacoustic composition for fixed media the composer is responsible for the final product in the same way as an interpreter is. There is not much to support abstaining from that artistic responsibility for the final product even in the strictest applications of algorithmic composition with autonomous instruments. Even though one deals with a form of found art, as Curtis Roads puts it, the influence the composer has on the final result comes by an extended process of exploration and critical selection. In any case, the absence of the two instances of safety nets makes

algorithmic sound synthesis with autonomous instruments a much harder endeavour than algorithmic composition intended to be performed by musicians. Or conversely, working with autonomous instruments can be easier because there are no censoring instances that will block the worst ideas.

8.3.4 Intention, interpretation, inspiration

Does art need interpretation in the form of more or less elaborate verbal contextualisation and even explanation, or can one be content with experiencing it directly? There are different views on this point, but it would be surprising to find an academic aesthetician defending the stance that no interpretation is needed as we look at or listen to works of art. There is, however, a risk with interpretation that it becomes over-interpretation, that it ascribes intentions to the author that were not present.

The underlying assumption appears to be that the artist always has an agenda, or a specific intention, which can be uncovered by analysing the art work. It is questionable if this is always the case with all artists; furthermore, the artist is not always in the position to tell the audience how the work is to be interpreted—it may have been made very intuitively and without much reflection. Even if there is an artist's statement that clarifies these issues, there is no reason why we should accept it as the only valid perspective on that particular work.

Talking about the "composer's intention" also shadows the fact that the process of composing a large-scale work takes a considerable amount of time. Are we to expect a single intention to have guided the composer's vision during this whole period? This is not very likely.

Rosemary Mountain (2001) writes about the creative process of composition, and in particular the role played by mental imagery. She lists the various stages in the compositional process as:

- gathering of material (may occur mentally, or by collecting sketches or recordings)
- arranging of gathered material by playing around with it, or by systematic exploration
- encoding of the material for communication to the listener.

These stages of the process are meant to supplant the more unrealistic idea that inspiration is the sole driving force behind composition:

> The initiative for composing is often thought to be inspiration, and in that guise it may subsume gathering and arranging stages. [...]
>
> The most basic and insidious form of what I think of as "the Mozart Myth" pretends that the composer's task is to receive divine inspiration in the form of a musical masterpiece, and then transcribe it [...] Its most erroneous implication is that inspiration arrives in the form of a pure and complete auditory image, already orchestrated, which the composer proceeds to encode from memory, from beginning to end (Mountain, 2001, p. 273).

It is obvious that inspiration has little to do with music making by autonomous instruments. If it still has a function, it is other than envisaging with some precision what the music will sound like. The stages of composing that Mountain lists also seem to fit better as a description of traditional composition rather than of algorithmic composition with autonomous instruments. However, the gathering of material may be rethought as a gathering of algorithmic techniques, small and handy programming tricks that can be applied in many situations, so instead of collecting sounds, one collects subroutines and programme libraries. Arranging this material means importing readymade components (feature extractors, signal generators, filters and other processing units) and trying out various combinations of them. Encoding of the material implies fixing the work in a form intended to be communicated to the listener. With an autonomous instrument, the encoding phase then is as simple as stopping the trial-and-error process of finding an acceptable output and saving the result. We would however argue that this is not so easy after all.

The compositional process of electroacoustic music differs from writing notated instrumental music by the immediacy with which the preliminary results can be heard. Thus, for the composer who has a clear idea of what to express, the electroacoustic medium should provide an excellent opportunity to realise those ideas without any distortion. However, as Mountain notes, in practice this direct specification of the composer's intention in sound is not practicable:

> The excitement with which many composers greeted the advent of electronic instruments was due to the elimination of this necessity of translation, as the original sonic idea could theoretically be reproduced with great fidelity to all its nuances, without having to be mediated by physical limitations of instruments and performers. Unfortunately, there are two tremendous obstacles to this process: the time required to arrive at the desired sound, and the amount of extraneous sounds which may have to be heard in the process, both of which can interfere with the integrity of the remembered sonic image in the composer's mind (Mountain, 2001, footnote, p. 286).

Working with autonomous instruments instead of more malleable synthesis techniques, the "amount of extraneous sounds" that one will have to sift through before finding something of use may be enormous. The point that listening to all those sounds will interfere with one's original intention (if ever there was one) is even more acute in the context of autonomous instruments. Many composers and other artists have experienced that it is extremely difficult to start with a blank canvass, but as soon as there is something there, the creative process gains momentum. Music made with autonomous instruments must not be understood as having materialised independently of the composer/programmer's work and without reflecting the artist's taste. The heuristic process of making gradual adjustments to the autonomous instruments based on what music it currently produces offers plenty of opportunities to tutor the algorithm until it demonstrates some level of musicality.

8.3.5 Algorithmic composing is not for everybody!

While many studies have been made of performing musicians and of listeners' reactions to music, relatively little is known about how composers actually do their composing. Certainly there are self-reports in programme notes, autobiographies and so on, but there are few empirical studies of larger groups of composers and their attitudes towards music and the creative process. A recent exception is a study by Peter Holtz. He interviewed 17 composers, including jazz musicians and composers of movie scores, popular music and musicals, but most were composers of "classical" (contemporary) music (Holtz, 2009). The interviews were carried out twice with each participant. After the first session, Holtz distilled some descriptions of each composer, and then validated these descriptions by discussing them with the interviewees and reaching agreement on the formulations to use.

Holtz asked how the composers created their music; whether they followed rules, or if their moods, memories or surroundings influenced their composing. Next, Holtz asked "whether creating music felt to them more like hard work or more like being kissed by a muse," and whether their music was a "language of the heart" or "tonally moving forms" (the latter with reference to Hanslick's formulation). There were also questions about whether symbolic meanings or the abstract structure of the music was the most important aspect. Finally, the composers were asked about their theories about their listeners, and "what a listener has to bear in mind when listening to the music" (Holtz, 2009, p. 211).

At this point, we can see that Holtz had already elaborated some categories of composers by formulating his set of questions. The composers are listed as belonging to three categories or types: neo-romantic, avant-gardist, and self-disclosing artists. Yet several composers would position themselves in two of these categories; furthermore no "avant-gardist" composer labelled himself as such. Although the preconceived categories may appear problematic, it seems to be an unavoidable effect in this kind of research. Another problem with the study is that it is not limited to one kind of composition, but includes jazz musicians whose live improvisation is supposedly regarded as comparable to pencil and paper composition at the desk. Furthermore, the small number of participants from a limited geographical region (southern Germany) makes it hard to generalise the results. The strength of this study is that the interviews shed more light on the group of composers than any questionnaire would do. From this, a picture emerges of three composer types that may to some extent also be recognisable in other milieus.

Let us consider the results with a view to what may be predicted about the use of autonomous instruments as a compositional strategy. What kind of composer is it that is willing to engage with such systems? The answer appears self-evident: only the avant-gardist composer will care for autonomous instruments, because they are found to prioritise formal aspects of music over emotional outpourings (self-expression). The neo-romantic "must write his/her own music" (Holtz, 2009, p. 220), and express their feelings and thoughts through the music. Nevertheless, this group wants the music to be comprehensible to the audience. Obviously, an immediate self-expression is made inefficacious or entirely blocked by the adherence to algorithmic composition. Like the neo-romantics, the self-disclosing artist has a great need for self-expression, so much so that many of these artists "would perish or go crazy without the possibility to express themselves in

their music" (Holtz, 2009, p. 221). This group relies heavily on improvisation in the creative process. Although the stringency of autonomous instruments and programming seems contrary to the needs of the self-disclosing type, one cannot rule out that some of them might enjoy improvisatory music making with semi-autonomous instruments.

In Chapter 1 we discussed experimental music and mentioned Tim Perkis' dichotomy into experimental and romantic self-expressing types of composers (Perkis, 2003). The experimental type of Perkis obviously fits well in Holtz' avant-gardist type, whereas Perkis' romantic type can be seen as split into two subtypes. There is reason to view results such as these as preliminary at best. After all, how can we find out what types of people there are in the world without first thinking up a few categories and asking whether they belong to one of them? Regardless of what types of composers we end up with, and what labels we attach to them, the fact of the matter remains that not all of them will be interested in pursuing non-interactive algorithmic composition with autonomous instruments. Those who will may be a minority, perhaps with some commonalities in their views about audiences, aesthetics and what musical composition is all about, but it is even more likely that one will find a lot of individuality. In a review of different approaches to algorithmic composition, Ames (1987, p. 182) noted that "composition has *not* been a vehicle of 'expression' or 'affect' in the usual senses of these words" for the composers that he discussed, including Hiller, Xenakis, Brün, Koenig and others.

Another recent study dealt specifically with composers of electroacoustic music and their *cognitive styles*, that is, tendencies to consistently adopt particular types of information processing strategies (Eaglestone et al., 2008). Two cognitive style traits that were found to be relevant in understanding how composers approached the composing process and how they related to software were the *global* versus the *analytical* types, and the *imagers* versus the *verbalisers*. Analytical individuals are described as approaching tasks in a linear, stepwise, logical fashion, whereas global individuals try to establish an overview of how ideas fit together before turning to details. Imagers are good at visual thinking, working with diagrams and pictures, in contrast to verbalisers who are better with words. Eaglestone et al. found two distinct approaches to composition: *refinement* and *synthesis-based* approaches (the latter term does not necessarily imply a strong interest in sound synthesis). Of their 27 questionnaire respondents, 13 were classified as applying the refinement approach and five of them the synthesis-based approach. Composers using the refinement approach begin by establishing the structure of a composition, which is then realised and refined. The synthesis-based approach is described as a "voyage of discovery" in which the composition emerges through experimentation with audio material.

It would be interesting to know whether algorithmic electroacoustic composition such as we have developed fits better with one of the compositional approaches than the other. Refinement appears to indicate that one has an initial diffuse idea of the finished piece that can gradually be steered in the desired direction. Inasmuch as autonomous instruments offer hard-to-control emergent properties, they seem irreconcilable with a refinement approach, although this need not be the case. The refinement offered by small structural modifications and parameter tweaking does not allow detailed control of localised parts, but always influences the generated sound in a global way. Perhaps the synthesis-based approach is more appropriate for work with autonomous instruments. Af-

ter all, it is qualified as a voyage of discovery through experimentation. We have pointed out the experimentalist musical tradition as an important context for music making with autonomous instruments, and surely there is much to discover in the algorithmically generated sounds. However, it seems premature to take a definite standpoint on the question of which approach is more natural for composition with autonomous instruments.

Another interesting theme discussed by Eaglestone's team is the contentment of composers with the music software they use. Regarding one composer who used "home made" software, they noted that this is an implicit criticism against existing software, since it was not sufficient for that composer's needs. In general, those who were content with the software they used were classified as having the verbal cognitive style trait, whereas those who were dissatisfied were typically of the imager type. According to Eaglestone et al., all of the contented verbalisers used programming languages as their main compositional tool. The imagers were mainly concerned with the interfaces of existing software.

Composing algorithmic electroacoustic music with autonomous instruments (feature-feedback systems in particular) is presently only possible through programming one's own routines. The experiments discussed in the previous two chapters were carried out in the entirely text based C++ language, but nothing restricts the ideas from being implemented in any audio programming language or general programming language. While simple, user-friendly interfaces to autonomous instruments are conceivable, this would probably be to misapprehend their proper use. These instruments belong to the class of composed instruments, with the singular characteristic of ad-hoc solutions to specific musical needs.

8.3.6 Programming as composing

Digital instruments are very different from acoustic instruments in several obvious ways. Thor Magnusson (2009) describes digital instruments as epistemic tools, scaffoldings onto which we can delegate parts of our cognitive process. The pencil and paper are helpful tools for tasks such as solving a mathematical problem or writing a piece of music. Magnusson points out that this scaffolding of musical software in particular incorporates music theoretical notions. Some software comes with restrictions motivated by supposed (and often actual) usefulness, such as having readymade scales and tunings at hand instead of having to specify pitches in Hz.

Although restrictions in software may get in the way for composers who know what they would like to do, it is true that "the practically infinite expressive scope of the environment" sometimes results in "a creative paralysis or in the frequent symptom of a musician-turned-engineer" (Magnusson, 2010, p. 62).

Programming, in contrast to live performance with a digital instrument, is often seen as a disembodied activity, although Owen Green (2011) disagrees, noting that it can engender a sense of flow such that the separateness between himself and the machine seems to dissolve. Live coding of course is performance (Nilson, 2007), thereby complicating our tidy conceptual schemes.

The programming environment for writing C++ code is not a very exciting thing to show to an audience; it is simply a text editor. Just like the poet staring at an empty sheet of paper, one may scribble whatever one likes on it. The constraints that exist are purely determined by the programming language and its syntax. In an ecological theory

of musical instruments, the notion of affordances holds a central place. An analogue synth panel with all its knobs and sliders affords turning and tweaking—anyone can immediately see what can be done to it. In contrast, it is not usual to speak of the affordances of a text editor for computer programming. Of course it affords writing code, or even scribbling nonsense that will not compile. Nor is it common to mention the constraints of the same editor. There are very little constraints, compared to what input one can provide to a rudimentary sequencer programme or even a highly sophisticated audio mixing programme. Magnusson (2010) is absolutely right in that respect when he states that affordances are not all-important in digital music instruments but their constraints are highly significant. The constraints are what the user will explore. Many musicians pride themselves on having discovered ways of using an instrument in which it was not intended to be used (Cascone, 2000; Gelineck and Serafin, 2009).

With autonomous instruments, and specifically with feature-feedback systems, there is a constraint in the form the system will take, and another constraint in the assumption that there is no live interaction. These are self-imposed constraints, which are very different to constraints that someone else has imposed on an instrument. Of course the usual situation with digital musical instruments now is that the luthier is the composer is the performer, and the instrument is composed to meet the needs of the situation.

8.3.7 No free lunch and the embedding principle

In evolutionary computing, the so called No Free Lunch theorem states that there is no evolutionary algorithm that is perfect for every conceivable task. If an algorithm is designed to be excellent at solving one type of problem, it will be mediocre on other problems; and conversely, if the algorithm is designed to do well on a majority of problems, there will be particular problem areas where some other algorithm outperforms it (Eiben and Smith, 2003). A somewhat similar limitation faces us with the design of autonomous instruments. We want them to be parsimonious with respect to code size, but we also want them to exhibit as complex behaviour as possible. However, one cannot hope to obtain any long-term unpredictable evolution without putting some effort into the design of the instrument. This means adding layers of control and sensing mechanisms. The necessary addition of these layers is what we have called the embedding principle.

In cybernetics, there is the so-called *Aulin's law of requisite hierarchy* (Heylighen and Joslyn, 2001) which is similar to the embedding principle. If a control loop such as a thermostat is used to reduce perturbations of a desired state, it may not be able to reduce all variation, so one may add another control loop on top of the first. More layers of nested control loops may be added, but as Heylighen and Joslyn note, this can have a negative effect on the regulatory ability because the system as a whole becomes more sensitive to noise or disturbances.

In some exceptional cases, the model may be capable of quite complex behaviour even without too many added layers of control. However, this complex behaviour may only be obtainable by searching the parameter space. Thus, serendipitous discoveries of interesting sounds may occur, but more often they will not come about without spending some time listening to the output of the system as it is modified and its parameters varied. In the words of George Lewis,

> [...] the search for the right algorithm is made gradually by listening, by trial and error. [...] I used to listen to my program all day and half the night. I would turn it on for 10 minutes here, an hour there, and see what it would do. It's like listening to a musician practice, and every so often you say, "No, do it *this* way" (Roads, 1985, p. 83).

Thinking of algorithmic composing as similar to the job of a music instructor is perhaps not the most common perspective, but listening is indeed a crucial part of the process of getting an algorithm to produce the musical output one would like to hear.

It might appear possible to do away with the tedious processes of listening to minutes and hours of mediocre attempts from the autonomous instrument by some form of automated search of its parameter space. Evolutionary algorithms might come to the rescue, if only one knew what to look for and how to define the goal, as discussed in Chapter 6. This problem cannot be ignored. Arguably, the long listening process is valuable in itself, because it gives the composer an opportunity to hear a wide range of not very successful attempts and perhaps some very good ones. The selection of the optimal version cannot be made until one has heard enough variants that they cease to impress by their sheer novelty. Or rather, there is the risk of picking the first best sound one hears, even if it happens to be less interesting than something that might be found after a bit of searching in the parameter space. This brings us to another problem which may sound trivial but is nevertheless very real.

8.3.8 When to stop tweaking the algorithm

Although there are many different approaches to algorithmic composition, those who use it for genuine composition often do so in order to generate material that they could not have thought of themselves. Granted that algorithmic composition may fulfill this need, the question is how far one should go in the potentially endless cycle of making refinements to the algorithm.

The typical workflow of algorithmic composition may in fact itself be described by an algorithm, such as:

1. Write the algorithm
2. Run the programme
3. Listen to the output
4. If unsatisfactory, goto step 1 (revise the code)
5. Else, the composition is finished.

As long as the algorithm is not too complicated, it is almost inevitable that one will learn how to cause it to produce certain kinds of sounds (or note patterns). With autonomous instruments of some complexity such learning may indeed be hard to attain, but an intuitive understanding is a likely side effect of this process. In any case, in developing an autonomous instrument to be used in a composition one has to decide at what point to stop the development process and store the output permanently as a sound file, which

is the final piece, ready for a concert performance or for release on a CD or whatever other distribution method one chooses.

In practice, it might be tempting to apply some final mastering before the end result is fixed, or perhaps the piece will be more effective if a few seconds are trimmed from the beginning. If the composer is very fussy about details, more substantial editing may be needed before the generated material becomes an acceptable musical piece. At some point it may even seem that the use of algorithms as a technique to generate material for further kneading into musical shape is just an unnecessary obstacle to the composer's direct self-expression.

Steps 3 and 4 in the algorithm above are essential. As long as a critical evaluation of the output is part of the process, *musique à priori* is out of the question. Live coding also seems to fit into this scheme but then all the steps will run concurrently. Nick Collins (alias Click Nilson) has described the training routines he submitted himself to in order to attain more fluency with the pressing demands of live coding in front of an audience. He contends that today's standard of live coders is at the level of "potentially talented 11 years olds" (Nilson, 2007). Perhaps, some day, their musical level will reach that of professional musicians who have played acoustic instruments for years, although Collins does not appear to regard this as very likely. If the investigation into feature-feedback systems in the two preceding chapters has made them seem complicated to handle, this impression is entirely warranted. There is little that speaks in favour of using them in live coding situations if one wants some degree of control of the results, but of course anyone is free to try. Although an extended period of algorithm tweaking and exploration could precede any concert performance with such systems, there is a natural deadline which eradicates the problem of when to stop the fine-tuning process.

8.3.9 Design or emergence?

The question poses a dilemma: if a process is used to generate material for musical compositions and we want the material to possess some unique and astonishing properties, then we cannot design it in detail ourselves because that way we would introduce our own predilections. In other words, if emergent behaviour is what we hope to see then we may design the apparatus that generates it, but as soon as we begin to understand it too well the psychological component of surprise evaporates.

As Gelineck and Serafin (2009) found in a study of electronic musicians, many of the musicians enjoyed using musical tools that they did not fully understand, or tools they could use in unintended ways. Therefore, the authors proposed some thought-provoking ideas, such as designing the musical tool for unintended use, or designing a tool so as to balance between being intuitive and unpredictable. Whereas the design for unintended use appears to be a paradoxical strategy, the balance between the intuitive and the unpredictable is clearly feasible. All our examples of feature-feedback systems occupy a position somewhere along this continuum; extended exploration of any system will certainly contribute to making it more predictable.

There are certain aesthetic and practical limitations of adopting ready-made systems for musical composition instead of designing them, as Pressing noted in the case of using chaotic maps for the generation of musical material:

> Procedurally, the method used here could be described as "found process" [...] While the limitations of found process are probably less than for found objects, since tunable parameters are already built into the method, it is no good trying to look for snowballs on the coast of Florida. Some things may have to be built rather than found (Pressing, 1988, p. 44).

Autonomous instruments that do more or less what one would hope for may have to be designed rather than stumbled upon by accident. However, the apparent contradiction need not be irreconcilable. By design, a wide spectrum of potential outcomes within fixed limits may be guaranteed. Insofar as the instrument retains an autonomous aspect, exact control remains impossible.

Let us return to the problematic concept of self-organisation. As Gershenson and Heylighen (2003) point out, there are three loose terms to define: "self", "system" and "organisation". Autonomous instruments may be assumed to be systems (without going too deeply into systems theory, it is sufficient to say that a system consists of several interacting parts, and might have an input and an output). What constitutes organisation is harder to say. At least in a colloquial sense we might say that autonomous instruments organise *themselves*. It is not the programmer who writes the algorithm who decides on every little quirk of the resulting sound given some specific initial conditions and parameters. In that sense, the autonomous instrument is self-organising. Relating this sense of self-organisation to the degree of understanding the programmer has of the processes that create the sound is almost inevitable. With a poor understanding and given intricate results, one will feel compelled to qualify the outcome as self-organised sound. Suppose instead the output is dull and easy to predict—then a fixed point or cyclic orbit of the system has been found. This may happen either with a full understanding or with a frail or faulty intuition of the system's dynamics. We do not describe an oscillator that produces the specified waveform at the specified pitch and amplitude as producing "self-organised" sound. If we had mapped out the relationships of control parameters to sound for an autonomous instrument in such great detail that we could produce exactly those sounds that we intended to, knowing the limitations of the instrument, would it not be correct to say that this is a synthesis model like any other; that there is definitely no magic going on behind the scene, no self-organisation happening? This seems to be the correct conclusion.

However, there is a caveat. One must assume that it is possible to map out the territory of the autonomous instrument and to know the parameter-to-sound relation up to arbitrary precision. For a well-behaved oscillator, one does not have to make tables of what frequency it oscillates on given this and that control parameter value since it is designed to give precisely the frequency we ask for. With a more complicated autonomous instrument, such as one of our feature-feedback systems, one would have to store enormous amounts of data about what parameter configurations (initial conditions) yield which sounds, and there is no shortcut. In that sense, it is impossible to predict its output given hitherto untried parameter combinations.

In Chapter 5, we discussed the idea that emergence is in the eye of the beholder, and now we add self-organisation as crucially depending on our amount of knowledge about the system. Gershenson and Heylighen (2003) made a similar point about the

role of the observer in classifying a system as either self-organising or self-disorganising. Using entropy reduction as the criterion, they provide an example where a number of states of a system can be partitioned in various ways by the observer. Depending on how the partition is made, the same system may then undergo a change that increases or decreases its entropy. We have noted the shortcomings of entropy reduction as a criterion of self-organisation, and here again its deficit is noticed.

Instead of trying to apply a rigorous definition of self-organisation, we will be content with an every-day understanding that *self-organisation indeed occurs in autonomous instruments* when they are sufficiently complicated.

8.4 Conclusion

In this chapter, we have stressed the importance of insights into the poietic side of electroacoustic compositions in order to be able to qualify the piece as being realised with an autonomous instrument. The algorithmic composition of electroacoustic music is an endeavour that may not interest every composer, particularly with the kind of detached involvement typical of autonomous instruments. Semi-autonomous instruments, however, seem to attract growing interest from performing electronic musicians and academic researchers alike (Collins, 2007b; Eldridge, 2008; Jordà, 2007). Nonetheless, autonomous feature-feedback systems offer an interesting opportunity to study a new kind of complex system, which may generate fascinating sounds over relatively long time scales.

We have mostly assumed that the output of a feature-feedback system might be used as is, to constitute an entire musical composition without further editing. Another assumption is that this autonomous instrument generates its stream of audio in one go, so that the entire composition results from a single run of the programme. There is no explicit note level or other higher hierarchic levels; the dynamics depends on rules that work on very short time scales. If any differentiated musical patterns or formal sections should appear, these may be said to be an emergent property of the system.

8.4.1 Summary of new findings

In Chapter 2, we argued that too a restricted view of what timbre is may lead to an unfortunate limitation in timbre studies to pitched sounds. An intuitive understanding of how feature extractors relate to the perceived sound, including timbre, is useful for anyone who uses feature extractors in synthesis models or audio effects. Then, in Section 3.2 we addressed the problem of interdependence of features with examples from additive synthesis. For example, the tristimuli were shown to depend on the odd to even ratio and the spectral slope, which means that these five features are mutually dependent. This has consequences for the choice of feature extractors when the task is to analyse sounds with as few feature extractors as possible.

Some nonlinear synthesis models were discussed in Section 3.3, including wave terrain synthesis, discrete summation formulas and the tremolo oscillator. These techniques offer a powerful global control of timbre by one or a few parameters, which make them suitable for use as signal generators in feature-feedback systems. In Section 3.4, we also pointed out the need for timbral fine-tuning or post-processing of the raw material that comes out

of any synthesis model, including feature-feedback systems. Finally, we called attention to the need to consider higher levels than the single note, which is important in autonomous instruments, where the synthesis of global form occurs in parallel with the micro-level synthesis on short time scales.

The dynamic systems perspective introduced in Chapter 4 is crucial for understanding deterministic feature-feedback systems. Nonlinear FIR-type filters were shown to be a generic model of feature extractors, and filtered maps were introduced as a simplified model of feature-feedback systems (see Section 4.3). In addition, filtered maps were shown to be useful in sound synthesis including physical modelling and nonstandard synthesis. Chaotic maps, being autonomous systems in the dynamic systems sense, are closely related to autonomous instruments when used for sound synthesis.

The dynamic systems perspective was further elaborated in Chapter 6, where the general feature-feedback system equation was introduced (see Section 6.1.1). An algorithm for the estimation of Lyapunov exponents was introduced, although its validity remains to be tested. Spectral bifurcation plots were also introduced and contrasted with sonograms. The effects of hysteresis in feature-feedback systems can be easily identified by comparing the sonogram with a spectral bifurcation plot.

Several case studies of feature-feedback systems were presented in Chapter 6, including the cross-coupled map, the extended standard map, a noise-driven oscillator, and the wave terrain system. Different methods for studying the parameter space of these systems were discussed, such as reducing the system to a low-dimensional map, using feature extractors on the resulting output signal in generalised bifurcation diagrams, and in the case of stochastic systems, to study ensemble averages. A simple mechanism of pitch control that can be used in nonlinear or noise-driven oscillators was described in Section 6.4.4.

Of the systems investigated in Chapter 6, the wave terrain system was the only one that I found to be interesting enough for inclusion in the Autonomous Instrument Song Contest, in company with the discrete summation formula system and the tremolo oscillator developed in Chapter 7. These three instruments are capable of much more variation than can be exhibited in a few sound examples.

In the first chapter (Section 1.3), we discussed the aesthetics of nature and its supposed anti-thesis, the artificial. It seems to be more than a coincidence that nature, in the sense of biological life, is often taken as a model for semi-autonomous instruments. Indeed, some speak of an ecosystemic approach to music making where the semi-autonomous instrument is not just a predictable and controllable instrument, but a musical partner with a voice of its own. The notion of nature in this case is not necessarily that which can be heard in soundscape compositions. It is a more abstract notion of nature that is often expressed in a trust in the generative algorithm. As we also noted, this may include very artificially sounding music that could not have been made with purely acoustic means.

Self-organisation is often associated with feedback systems or iterative processes. It may not even make sense to speak of self-organisation unless there is some kind of feedback from the current state of the system to the next state. We pointed out the lack of precision in terms such as self-organisation and emergence in Chapter 5, and argued that there is scope for refinement. If there is such a thing as self-organised sound, we mentioned

Di Scipio's *Audible Ecosystemic* works as a possible example.

Similar conclusions may be drawn concerning feature-feedback systems and other autonomous instruments. If the resulting music was not specified in detail by the composer, but emerged out of the system's internal dynamics, then it must have been self-organised. On the other hand, as discussed in this chapter, there is often a long process of gradual refinements of the generative algorithm, until it begins to output sounds that satisfy the composer's demands. Consequently, the composer is as responsible as ever for the resulting music, although the way to arrive at a personal expression is very different from that of more intuitive approaches to composition.

Simple and complex music were discussed in Section 5.2, and we argued that perceptual complexity is a useful criterion for evaluating the sounds made with autonomous instruments. However, what is referred to as "simple music" (in contrast to musical examples drawn from the new complexity movement) may still be rather sophisticated when compared to the output of a feature-feedback system that has landed on some fixed point in its state space.

The dynamic systems perspective may sometimes be applied to the analysis of musical works, if they fit into the paradigm. Examples thereof include Lucier's *I am sitting in a room*, and several other works involving feedback processes that were analysed in Section 5.1. Although this dynamic systems perspective should be complemented with more traditional musicological approaches, it can be a valuable addition in the analysis of certain types of process-related or generative music.

For the automated evaluation of the complexity of a sound generated by an autonomous instrument, a rather crude complexity measure such as the SOF measure of timbral complexity (introduced in Section 7.3.2) may be a good place to start, since some static behaviour may then be detected and excluded. The Autonomous Instrument Song Contest provided ground truth for evaluating the SOF measure. However, we found SOF to be an insufficient measure of perceived timbral complexity, although it may be useful for distinguishing simpler signals from those used in the listening test. There are of course other aspects to evaluate in the resulting sound that cannot be subsumed under complexity, and there is always a need for subjective assessment of the success or failure of a given autonomous instrument. The need for objective complexity measures as applied to autonomous instruments arises if the parameter space should be automatically searched with evolutionary algorithms or when using the instrument in generative music without interaction.

Several solutions to the problem of increasing the sonic variability of autonomous instruments were considered in Chapter 7. The techniques involved strategies for avoiding fixed points by perturbing the system if a stationary state was reached, or using statistical feedback by keeping track of the past states of the system. A step sequencer construction was also found to be practical for controlling the traversal of a prescribed list of parameter values. The problem of eschewing too restricted behaviour was argued to depend on the fact that, as a dissipative system, the feature-feedback system will approach an attractor that occupies a limited volume of the total state space. In other words, only a limited set of synthesis parameter values will be available, resulting in a likewise limited range of sounds. The design methods that were devised for breaking up such restricted behaviour add new layers of control to the system, thus exemplifying the embedding principle.

In Chapter 7, Section 7.4, we considered how to extend autonomous instruments to sampling synthesis, as well as applications of concatenative synthesis using the sounds from an autonomous instrument as the corpus. Since the quality of resynthesis with concatenative synthesis depends on the diversity of the corpus, the usefulness of autonomous instruments as sources for the corpus depends on how varied sounds they generate; furthermore, it may depend on how the parameter space is searched.

Motif mappings were then introduced in Section 7.5 as a generalisation of feature-feedback systems to the note level. The motif level descriptors are useful for designing meaningful operations on the motifs, leading to an approach to algorithmic composition similar to constraint programming.

Several examples of supposedly autonomous instruments have been mentioned in passing throughout this thesis. As pointed out in this chapter, the available information on the construction process of the compositions does not always permit correct attributions. This reservation notwithstanding, let us list some works that may qualify as being made with autonomous instruments.

- Dunn's *Nine Strange Attractors* (Dunn, 2007), see Section 4.2.4
- Ikeshiro's *Construction In Self* (Ikeshiro, 2010), see Section 4.2.4
- Xenakis' *Gendy3* and *S.709* (Xenakis, 1992; Hoffmann, 2000; Serra, 1993)
- Perkis' *Clicks* (Bischoff and Perkis, 1989), see Section 1.3.3
- Kayn's large scale electroacoustic compositions (see above, Section 8.2.9)
- Bytebeat one-liners (Heikkilä, 2011), see Section 5.2.4

Although examples of music made with strictly autonomous instruments are rare, a few more examples may be found under the heading of generative music.

8.4.2 Open questions

Many problems remain to be addressed, both for the musicologist interested in a better understanding of music made with more or less autonomous instruments, and for the composer or musician who wants to learn how to design them more efficiently. A difficulty faces the investigation of existing music made with semi-autonomous instruments: frequently we do not know enough of its poietic dimension. Relatively simple generative music systems that depend on some random seed already offer quite a challenge to the music analyst (Collins, 2008a).

In the making of novel autonomous instruments, there is a wealth of combinatory possibilities to explore. We have made use of a rather limited set of feature extractors, synthesis models and mappings, and have by no means exhausted the capacity of any of the proposed autonomous instruments. And if the path of stubbornly non-interactive instruments turns out to be a blind alley, there are many ways to make them attend to the outside world in realtime. We have not failed to notice the difficulty of coercing a feature-feedback system to produce material that goes in a desired direction, especially one that holds a listener's attention for some time by introducing variation and novelty.

Live interaction has become increasingly popular, thereby possibly turning attention away from the old-fashioned métier of non-realtime algorithmic computer music. Live coding of feature-feedback systems would surely make the perfect counterexample, were anyone to bother to try it.

We have neither proved that feature-feedback systems are chaotic, although there is little reason to doubt it, nor that they self-organise in some strict sense. The diagnosis of self-organisation, however, is complicated by the lack of consensus about the concept as well as by the difficulty of applying most of the proposed measures. There are several objective complexity measures that would be interesting to apply to feature-feedback systems. A well chosen complexity measure would make the automated search of the parameter space feasible. This is currently a problem, since the really interesting sounds may be hard to find by manually searching the parameter space.

The development of novel synthesis techniques has hitherto been the work of engineers, and to some extent that of experimentally minded composers. The design and sound modelling perspective has dominated this research, and still does, as can be seen in the numerous publications of papers on physical and spectral modelling in recent years. Feature-feedback systems are complex systems in their own right which may be of interest not only to composers, but potentially also to researchers interested in chaotic and complex systems in general.

There is definitely also more to be done with feature-feedback systems as tools for musical composition. We have sketched many alternatives for how that may happen, from the simplest oscillators to note-level motif mappings. Who knows what kind of music will ensue?

Appendix A

Notations and abbreviations

These abbreviations and symbols are used throughout the text. Other notations are introduced and explained in the context where they occur.

$\text{osc}(f_n)$ Oscillator producing sinus tone with instantaneous frequency f_n

f_s Sampling rate

$*$ Convolution operator. Often used as shorthand for the application of IIR filters

$\langle x \rangle$ Average of x

x^* Fixed point of a map

$\dot{x}$ Time derivative of x; velocity

$\ddot{x}$ Second derivative of x with respect to time; acceleration

x_n Discrete time signal at time n, also written $x[n]$

x_t Continuous time signal at time t, also written $x(t)$

$\hat{x}$ Estimated value of x; feature extractor x

$\mathbb{N}$ The natural numbers

$\mathbb{Z}$ The integers

$\mathbb{R}^n$ n-dimensional Euclidian space

$\mathbb{R}^+$ The non-negative real numbers

$\mathbb{C}$ The complex numbers

$[a, b)$ The half-open interval including a but not b; the set $\{x : a \leq x < b\}$

$f^k(x_0)$ k times iterated map; k-fold function composition $f \circ \cdots \circ f(x_0)$ yielding x_k

l_1**-norm** the distance $d(x, y) = \sum_{i=1}^{N} |x_i - y_i|$

2-D Two-dimensional

A1A3 Relative amplitude of first and third formants (frequency domain)

a1a3 Same as A1A3, but implemented in time domain

AM Amplitude Modulation

DC Direct Current, a constant signal (at 0 Herz)

DFT Discrete Fourier Transform

FFT Fast Fourier Transform

FIR Finite Impulse Response

FM Frequency Modulation

IIR Infinite Impulse Response

MDS Multi-dimensional scaling

MIDI Musical Instruments Digital Interface. Communications protocoll between digital instruments etc.

N-TET N-tone equal temperament scale and tuning (typically $N = 12$)

ODE Ordinary Differential Equation

OER Odd to even ratio (of partials' amplitudes)

PC Pitch Class

RMS Root Mean Square

SOF Spread of Features

ZCR Zero Crossing Rate

Bibliography

Abel, M. and Bergweiler, S. (2007). Synchronization of higher harmonics in coupled organ pipes. *International Journal of Bifurcation and Chaos*, 17(10):3483–3491. 151

Abraham, R. and Ueda, Y., editors (2000). *The Chaos Avant-Garde. Memories of the Early Days of Chaos Theory*, volume 39 of *Nonlinear Science, Series A*. World Scientific, Singapore. 124

Abrams, D. and Strogatz, S. (2004). Chimera states for coupled oscillators. *Physical Review Letters*, 93(17):174102. 157

Adams, N. (2006). Visualization of musical signals. In Simoni, M., editor, *Analytical Methods of Electroacoustic Music*, chapter 2, pages 13–28. Routledge. 338

Adorno, T. (2006). *Towards a Theory of Musical Reproduction*. Polity Press, Cambridge. 18

Aks, D. and Sprott, J. C. (1996). Quantifying aesthetic preference for chaotic patterns. *Empirical Studies of the Arts*, 14(1):1–16. 183

Amatriain, X., Bonada, J., Loscos, A., Arcos, L., and Verfaille, V. (2003). Content-based transformations. *Journal of New Music Research*, 32(1):95–114. 81

Ames, C. (1987). Automated composition in retrospect. *Leonardo*, 20(2):169–185. 358

Anders, T. and Miranda, E. R. (2009). Interfacing manual and machine composition. *Contemporary Music Review*, 28(2):133–147. 36, 320

Arcos, J.-L., López de Mántaras, R., and Serra, X. (1997). SaxEx: a case-based reasoning system for generating expressive musical performances. In *Proceedings of the International Computer Music Conference (ICMC)*, Thessaloniki, Greece. 36

Ariza, C. (2009). The interrogator as critic: The Turing test and the evaluation of generative music systems. *Computer Music Journal*, 33(2):48–70. 352, 353

Ashby, W. R. (1947). Dynamics of the cerebral cortex automatic development of equilibrium in self-organizing systems. *Psychometrika*, 12(2):135–140. 193

Ashby, W. R. (1962). Principles of the self-organizing system. In von Foerster, H. and Zopf, G. W., editors, *Principles of Self-Organization: Transactions of the University of Illinois Symposium*, pages 225–278. Pergamon Press, London, UK. 194

Atay, F., Jalan, S., and Jost, J. (2009). Randomness, chaos, and structure. *Complexity*, 15(1):29–35. 129

Augoyard, J.-F. and Torgue, H. (1995). *A l'écoute de l'environnement. Répertoire des effets sonores.* Editions Parenthèses, Marseille. 160

Bandt, C. and Pompe, B. (2002). Permutation entropy: A natural complexity measure for time series. *Physical Review Letters*, 88(17):174102. 122, 177, 294

Beauchamp, J. (2007). Analysis and synthesis of musical instrument sounds. In Beauchamp, J., editor, *Analysis, Synthesis, and Perception of Musical Sounds. The Sound of Music*, Modern Acoustics and Signal Processing, chapter 1, pages 1–89. Springer. 57, 82, 85

Bennett, G. (1995). Thoughts on the oral culture of electroacoustic music. In *Aesthetics and Electroacoustic Music*, pages 20–25, Bourges, France. Acteon - Mnemosyne. 338

Beran, J. (2004). *Statistics in Musicology.* Chapman & Hall/CRC. 175, 239, 326

Berg, P. (2009). Composing sound structures with rules. *Contemporary Music Review*, 28(1):75–87. 10, 112

Bernardi, A., Bugna, G., and De Poli, G. (1997). Musical signal analysis with chaos. In Roads, C., Pope, S., Piccialli, G., and De Poli, G., editors, *Musical Signal Processing*, pages 187–220. Swets and Zeitlinger. 121, 131, 182

Bidlack, R. (1992). Chaotic systems as simple (but complex) compositional algorithms. *Computer Music Journal*, 16(3):33–47. 127

Bischoff, J. and Perkis, T. (1989). Artificial horizon. Artifact Recordings, ART 1003 [CD]. 20, 367

Bischoff, K., Firan, C., Georgescu, M., Nejdl, W., and Paiu, R. (2009). Social knowledge-driven music hit prediction. In Huang, R., Yang, Q., Pei, J., Gama, J., Meng, X., and Li, X., editors, *Advanced Data Mining and Applications*, pages 43–54. Springer, Berlin / Heidelberg. 189

Blackwell, T. (2007). Swarming and music. In Miranda, E. R. and Biles, J., editors, *Evolutionary Computer Music*, chapter 9, pages 194–217. Springer. 199

Blackwell, T. and Young, M. (2004). Self-organised music. *Organised Sound*, 9(2):123–136. 198, 199

Bloit, J., Rasamimanana, N., and Bevilacqua, F. (2009). Towards morphological sound description using segmental models. In *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, pages 445–450, Como, Italy. 46

Boashash, B. (1992a). Estimating and interpreting the instantaneous frequency of a signal—part 1: Fundamentals. *Proceedings of the IEEE*, 80(4):520–538. 59

Boashash, B. (1992b). Estimating and interpreting the instantaneous frequency of a signal—part 2: Algorithms and applications. *Proceedings of the IEEE*, 80(4):540–568. 59

Böhme, G. (2000). Acoustic atmoshperes. a contribution to the study of ecological aesthetics. *The Journal of Acoustic Ecology. Soundscape*, 1(1):14–18. 17

Bökesoy, S. (2007). Synthesis of a macro sound structure within a self-organizing system. In *Proc. of the Int. Conf. on Digital Audio Effects (DAfX-07).*, Bordeaux, France. 16, 349

Boon, J. and Decroly, O. (1995). Dynamical systems theory for music dynamics. *Chaos*, 5(3):501–508. 175, 176, 177, 182

Borgdorff, H. (2006). The debate on research in the arts. Available at: www.ips.gu.se/digitalAssets/1322/1322713_the_debate_on_research_in_the_arts.pdf. 3, 4

Boros, J. (1994). Why complexity? (part two) (guest editor's introduction). *Perspectives of New Music*, 32(1):90–101. 173, 174

Boulanger, R., editor (2000). *The Csound Book*. The MIT Press, Cambridge, Mass. and London. 25, 312, 341

Bown, O. (2011). Experiments in modular design for the creative composition of live algorithms. *Computer Music Journal*, 35(3):73–85. 16

Bradford, R., Dobson, R., and ffitch, J. (2005). Sliding is smoother than jumping. In *Proceedings of the ICMC 2005*, pages 287–290, Barcelona, Spain. 54

Bregman, A. S. (1990). *Auditory Scene Analysis*. The MIT Press. 48, 75, 86, 337

Broening, B. (2006). Alvin Lucier's I am sitting in a room. In Simoni, M., editor, *Analytical Methods of Electroacoustic Music*, chapter 5, pages 89–110. Routledge. 167, 168

Brown, A. (2005). Extending dynamic stochastic synthesis. In *Proceedings of the ICMC.*, pages 111–114, Barcelona, Spain. 170

Brown, A. and Sorensen, A. (2009). Interacting with generative music through live coding. *Contemporary Music Review*, 28(1):17–29. 326

Brün, H. (2004). *When Music Resists Meaning. The Major Writings of Herbert Brün.* Wesleyan University Press, Middletown, Connecticut. 34, 171

Buchner, T. and Żebrowski, J. (2000). Logistic map with a delayed feedback: Stability of a discrete time-delay control of chaos. *Physical Review E*, 63(016210). 152

Bürger, P. (1984). *Theory of the Avant-Garde*. University of Minnesota Press. 22

Burns, C. (2003). Emergent behavior from idiosynchratic feedback networks. In *Proceedings of the ICMC*, pages 267–270, Singapore. 169, 170

Cadiz, R. and Cuadra, P. (2010). Spectral stochastic resonance sound synthesis. In *Proc. of the ICMC 2010*, pages 353–356, New York. 124

Caetano, M. and Rodet, X. (2010). Independent manipulation of high-level spectral envelope shape features for sound morphing by means of evolutionary computation. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, pages 314–321, Graz, Austria. 110

Carterette, E. (1989). Perception and physiology in the hearing of computed sound. In Nielzen, S. and Olsson, O., editors, *Structure and Perception of Electroacoustic Sound and Music*, Excerpta Medica, pages 83–99. Elsevier. 19

Cascone, K. (2000). The aesthetics of failure: "post-digital" tendencies in contemporary computer music. *Computer Music Journal*, 24(4):12–18. 20, 253, 325, 360

Casey, M. (2009). Soundspotting: A new kind of process? In Dean, R., editor, *The Oxford Handbook of Computer Music*, chapter 20, pages 421–453. Oxford University Press. 80, 308

Casti, J. (1998). Complexity and aesthetics. *Complexity*, 3(5):11–16. 182

Chadabe, J. (1997). *Electric Sound. The Past and Promise of Electronic Music*. Prentice Hall. 32, 271

Chadabe, J. (2002). The limitations of mapping as a structural descriptive in electronic instruments. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*, Dublin, Ireland. 13, 28

Chareyron, J. (1990). Digital synthesis of self-modifying waveforms by means of linear automata. *Computer Music Journal*, 14(4):25–41. 172, 308

Chemillier, M. (2002). Ethnomusicology, ethnomathematics. The logic underlying orally transmitted artistic practices. In Assayag, G., Feichtinger, H., and Rodrigues, J., editors, *Mathematics and Music, Diderot Forum, European Mathematical Society*, pages 161–183. Springer. 34

Chen, G. and Dong, X. (1993). From chaos to order—perspectives and methodologies in controlling chaotic nonlinear dynamical systems. *International Journal of Bifurcation and Chaos*, 3(6):1363–1409. 153

Chion, M. (1983). *Guide des Objets Sonores. Pierre Schaeffer et la Recherche Musicale.* Éditions Buchet/Chastel., Paris. 24, 44, 176

Chowning, J. (1973). The synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society*, 21(7):526–534. 11, 95

Coca, A., Tost, G., and Zhao, L. (2010). Characterizing chaotic melodies in automatic music composition. *Chaos*, 20:033125. 127

Cochrane, P. (2010). A measure of machine intelligence. *Proceedings of the IEEE*, 98(9):1543–1545. 353

Coco, R. (2006). Minimalism and process music: a Pure Data realization of "Pendulum Music". In *SMC Conference*. Sound and Music Computing. 168

Coleman, G., Maestre, E., and Bonada, J. (2010). Augmenting sound mosaicing with descriptor-driven transformation. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, pages 494–497, Graz, Austria. 310

Coley, D. A. and Winters, D. (1997). Genetic algorithm search efficacy in aesthetic product spaces. *Complexity*, 3(2):23–27. 260

Collins, N. (2003). Generative music and laptop performance. *Contemporary Music Review*, 22(4):67–79. 326, 343

Collins, N. (2007a). Live electronic music. In Collins, N. and Escriván, J., editors, *The Cambridge Companion to Electronic Music*, chapter 3, pages 38–54. Cambridge University Press. 166, 167

Collins, N. (2007b). Musical robots and listening machines. In Collins, N. and d'Escriván, J., editors, *The Cambridge Companion to Electronic Music*, chapter 10, pages 171–184. Cambridge University Press. 16, 364

Collins, N. (2008a). The analysis of generative music programs. *Organised Sound*, 13(3):237–248. 26, 341, 342, 367

Collins, N. (2008b). Errant sound synthesis. In *Proc. of the ICMC 2008*, Belfast, Ireland. 144

Collins, N. (2009). Devil's music. EM Records, EM1086DCD [CD]. 344

Cope, D. (1992). Modeling of musical intelligence in EMI. *Computer Music Journal*, 16(2):69–83. 35

Cope, D. (1999). Facing the music: Perspectives on machine-composed music. *Leonardo*, 9:79–87. 351

Corning, P. (2002). The re-emergence of "emergence": A venerable concept in search of a theory. *Complexity*, 7(6):18–30. 191, 192

Cox, C. and Warner, D. (2004). *Audio Culture. Readings in Modern Music.* Continuum, New York. 23

Crutchfield, J. (1984). Space-time dynamics in video feedback. *Physica D*, 10:229–245. 162

Crutchfield, J. (1994). The calculi of emergence: computation, dynamics and induction. *Physica D*, 75:11–54. 192, 263

Crutchfield, J. and Young, K. (1989). Inferring statistical complexity. *Physical Review Letters*, 63(2):105–108. 178, 197

Dabby, D. (1996). Musical variations from a chaotic mapping. *Chaos*, 6(2):95–107. 127

Dack, J. (2005). The 'open' form – literature and music. Paper presented at the 'Scambi Symposium', Goldsmiths College. 334, 336

Dahlhaus, C. (1983). *Foundations of Music History*. Cambridge University Press. 351

Dahlhaus, C. (1992). *Analys och värdeomdöme [Analyse und Werturteil]*. Brutus Östlings Bokförlag Symposion, Stockholm. 185, 189, 351

Dahlhaus, C. (2005). Ästhetische Probleme der elektronischen Musik. In *Gesammelte Schriften*, chapter 11, pages 284–293. Laaber-Verlag. 334, 340, 345

Dahlstedt, P. (2001). Creating and exploring huge parameter spaces: Interactive evolution as a tool for sound generation. In *Proc. of the ICMC 2001*, Havana, Cuba. 260

Dahlstedt, P., Linde, J., and Nordahl, M. (2004). Chaotic sounds in coupled oscillator networks. Preprint, Chalmers. 153

Danto, A. C. (2005). *Unnatural Wonders. Essays from the Gap between Art & Life*. Columbia University Press, New York. 186

Davis, T. and Rebelo, P. (2005). Hearing emergence: Towards sound-based self-organisation. In *Proc. of the ICMC 2005*, Barcelona, Spain. 190, 198

Di Scipio, A. (1990). Composition by exploration of non-linear dynamic systems. In *Proc. of the ICMC 1990*, pages 324–327, Glasgow. 132

Di Scipio, A. (1999). Synthesis of environmental sound textures by iterated nonlinear functions. In *Proc. of the 2nd COST-G6 Conf. on Digital Audio Effects (DAFx'99)*, Trondheim, Norway. 112

Di Scipio, A. (2001). Iterated nonlinear functions as a sound-generating engine. *Leonardo*, 34(3):249–254. 133

Di Scipio, A. (2003). 'Sound is the interface': from interactive to ecosystemic signal processing. *Organised Sound*, 8 (3):269–277. 8, 16, 19, 123, 190, 202

Di Scipio, A. (2008). Emergence du son, son d'emergence. Essai d'epistemologie expérimental par un compositeur. *Intellectica*, 48(9):221–249. 8, 202

Di Scipio, A. (2011). Listening to yourself through the otherself: On Background Noise Study and other works. *Organised Sound*, 16(2):97–108. 123, 201, 202

Döbereiner, L. (2009). PV Stoch: A spectral stochastic synthesis generator. In *Proc. of the SMC — 6th Sound and Music Computing Conference*, pages 179–182, Porto, Portugal. 170

Döbereiner, L. (2011). Models of constructed sound: Nonstandard synthesis as an aesthetic perspective. *Computer Music Journal*, 35(3):28–39. 10, 112

Dobson, R. and ffitch, J. (1996). Experiments with non-linear filters. Discovering excitable regions. In *Proceedings of ICMC 1996*, pages 405–408, Hong Kong. 135

Donnadieu, S. (2007). Mental representation of the timbre of complex sounds. In Beauchamp, J., editor, *Analysis, Synthesis, and Perception of Musical Sounds. The Sound of Music*, chapter 8, pages 272–319. Springer. 48, 49

Dowling, J. (1989). Simplicity and complexity in music and cognition. *Contemporary Music Review*, 4:247–253. 337

Dunn, D. (2007). Autonomous and dynamical systems. New World Records 80660-2 (CD). 132, 338, 367

Dunn, D. and van Peer, R. (1999). Music, language and environment. *Leonardo Music Journal*, 9:63–67. 21

Eaglestone, B., Ford, N., Holdridge, P., and Carter, J. (2008). Are cognitive styles an important factor in design of electroacoustic music software? *Journal of New Music Research*, 37(1):77–85. 358, 359

Ebeling, W., Steuer, R., and Titchener, M. R. (2001). Partition-based entropies of deterministic and stochastic maps. *Stochastics and Dynamics*, 1(1):1–17. 294

Eckmann, J.-P., Oliffson Kamphorst, S., and Ruelle, D. (1987). Recurrence plots of dynamical systems. *Europhysics Letters*, 4(9):973–977. 270

Eckmann, J.-P. and Ruelle, D. (1985). Ergodic theory of chaos and strange attractors. *Reviews of Modern Physics*, 57(3):617–656. 215, 216

Eco, U. (1989). *The Open Work*. Harvard University Press, Cambridge, Massachusetts. 22, 335

Eddins, D. and Green, D. (1995). Temporal integration and temporal resolution. In Moore, B. C. J., editor, *Hearing. Handbook of Perception and Cognition*, chapter 6, pages 207–242. Academic Press, second edition. 50

Eiben, A. E. and Smith, J. E. (2003). *Introduction to Evolutionary Computing*. Springer. 258, 360

Elaydi, S. (2008). *Discrete Chaos with Applications in Science and Engineering*. Chapman & Hall/CRC, second edition. 116, 118, 125, 127, 138, 215, 217, 218, 272

Eldridge, A. (2008). *Collaborating with the Behaving Machine: Simple Adaptive Dynamical Systems for Generative and Interactive Music*. PhD thesis, University of Sussex. 26, 28, 291, 321, 349, 364

Eno, B. (1996). Generative music. "evolving metaphors, in my opinion, is what artists do.". *http://www.inmotionmagazine.com/eno1.html*, Talk delivered in San Francisco, June 8, 1996. 342

Essens, P. (1995). Structuring temporal sequences: Comparison of models and factors of complexity. *Perception & Psychophysics*, 57(4):519–532. 187

Essl, G. (2006a). Circle maps as simple oscillators for complex behavior: I: Basics. In *Proceedings of the ICMC*, pages 356–359, New Orleans. 144

Essl, G. (2006b). Circle maps as simple oscillators for complex behavior: II. Experiments. In *Proc. of the 9th Int. Conf. on Digital Audio Effects (DAFX-06).*, pages 193–198, Montreal. 144

Essl, K. (2007). Algorithmic composition. In Collins, N. and d'Escriván, J., editors, *The Cambridge Companion to Electronic Music*, pages 107–125. Cambridge University Press. 33

Feldman, J. (2004). How surprising is a simple pattern? Quantifying "Eureka!". *Cognition*, 93:199–224. 179, 188

Foote, J. (1999). Visualizing music and audio using self-similarity. In *Proc. ACM Multimedia 99*, pages 77–80, Orlando, Florida. 270

Foote, J. (2000). Automatic audio segmentation using a measure of audio novelty. In *Proceedings of IEEE International Conference on Multimedia and Expo*, volume I, pages 452–455. 270

Friedmann, M. (1985). A methodology for the discussion of contour: Its application to Schoenberg's music. *Journal of Music Theory*, 29(2):223–248. 130, 318

Frøyland, J. (1992). *Introduction to Chaos and Coherence*. Institute of Physics Publishing, Bristol and Philadelphia. 116, 145, 147, 219

Gammaitoni, L. (1995). Stochastic resonance and the dithering effect in threshold physical systems. *Physical Review E*, 52(5):4691–4698. 124

Garcia, R. (2001). Automating the design of sound synthesis techniques using evolutionary methods. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01)*, Limerick, Ireland. 259

Geiger, G. (2006). Table lookup oscillators using generic integrated wavetables. In *Proc. of the 9th Int. Conf. on Digital Audio Effects (DAFX-06)*, pages 169–172, Montreal, Canada. 105

Gelineck, S. and Serafin, S. (2009). From idea to realization - understanding the compositional processes of electronic musicians. In *Proc. Audio Mostly 2009*, Glasgow, Scotland. 360, 362

Gerratt, B. and Kreiman, J. (2001). Measuring vocal quality with speech synthesis. *Journal of the Acoustic Society of America*, 110(5):2560–2566. 45

Gershenson, C. and Heylighen, F. (2003). When can we call a system self-organizing? In *Advances in Artificial Life, 7th European Conference, ECAL 2003*, pages 606–614, Dortmund, Germany. 363

Glazier, J. and Libchaber, A. (1988). Quasi-periodicity and dynamical systems: An experimentalist's view. *IEEE Transactions on Circuits and Systems*, 35(7):790–809. 145

Godøy, R. I. (1997). *Formalization and Epistemology*. Acta Humaniora. Scandinavian University Press, Oslo. 176

Gogins, M. (1991). Iterated functions systems music. *Computer Music Journal*, 15(1):40–48. 127

Goodwin, M. (1998). *Adaptive Signal Models. Theory, Algorithms and Audio Applications*. Kluwer Academic Publishers, Boston. 82, 83

Gottwald, G. and Melbourne, I. (2004). A new test for chaos in deterministic systems. In *Proc. of the Royal Society of London*, volume A, pages 603–611, London, UK. 122

Grachten, M., Arcos, J.-L., and López de Mántaras, R. (2004). Melodic similarity: Looking for a good abstraction level. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR04)*, pages 210–215, Barcelona, Spain. 318

Grassberger, P. (1986). Toward a quantitative theory of self-generated complexity. *International Journal of Theoretical Physics*, 25(9):907–938. 178, 197

Grebogi, C., Ott, E., Pelikan, S., and Yorke, J. (1984). Strange attractors that are not chaotic. *Physica*, 13D:261–268. 216

Grebogi, C., Ott, E., and Yorke, J. (1983). Crises, sudden changes in chaotic attractors, and transient chaos. *Physica*, 7D:181–200. 220, 234

Green, O. (2011). Agility and playfulness: Technology and skill in the performance ecosystem. *Organised Sound*, 16(2):134–144. 29, 359

Gregson, R. and Harvey, J. (1992). Similarities of low-dimensional chaotic auditory attractor sequences to quasirandom noise. *Perception & Psychophysics*, 51(3):267–278. 129, 138, 183

Grey, J. (1977). Multidimensional perceptual scaling of musical timbres. *Journal of the Acoustic Society of America*, 61(5):1270–1277. 47

Hadjitodorov, S. and Mitev, P. (2002). A computer system for acoustic analysis of pathological voices and laryngeal diseases screening. *Medical Engineering & Physics*, 24:419–429. 69

Hajda, J. (2007). The effect of dynamic acoustical features on musical timbre. In Beauchamp, J., editor, *Analysis, Synthesis, and Perception of Musical Sounds. The Sound of Music*, chapter 7, pages 250–271. Springer. 49

Hamming, R. (1998). *Digital Filters*. Dover, Mineola, New York, third edition. 106

Hanche-Olsen, H. (2005). Om kurvers areal. *Normat*, 53(1):2–12. 245

Hao, B.-L. and Zheng, W.-M. (1998). *Applied Symbolic Dynamics and Chaos*, volume 7 of *Directions in Chaos*. World Scientific, Singapore. 129, 145, 232

Heikkilä, V.-M. (2011). Discovering novel computer music techniques by exploring the space of short computer programs. (arXiv:1112.1368v1 [cs.SD]). 180, 367

Henaff, M., Jarrett, K., Kavukcuoglu, K., and LeCun, Y. (2011). Unsupervised learning of sparse features for scalable audio classification. In *Proc. of the 12th International Society for Music Information Retrieval (ISMIR 2011)*, pages 681–686, Miami, USA. 83

Herrera-Boyer, P., Peeters, G., and Dubnov, S. (2003). Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21. 52

Heyduk, R. (1975). Rated preference for musical compositions as it relates to complexity and exposure frequency. *Perception & Psychophysics*, 17(1):84–91. 188, 302

Heylighen, F. and Joslyn, C. (2001). Cybernetics and second-order cybernetics. In Meyers, R. A., editor, *Encyclopedia of Physical Science & Technology*. Academic Press, New York, third edition. 349, 360

Hiller, L. (1981). Composing with computers: A progress report. *Computer Music Journal*, 5(4):7–21. 35

Hoffman, M. and Cook, P. (2006). Feature-based synthesis for sonification and psychoacoustic research. In *Proceedings of the 12th International Conference on Auditory Display*, London, UK. 78

Hoffman, M. and Cook, P. (2007). Real-time feature-based synthesis for live musical performance. In *Proceedings of the 2007 International Conference on New Interfaces for Musical Expression (NIME)*, New York. 79

Hoffmann, P. (2000). The new GENDYN program. *Computer Music Journal*, 24(2):31–38. 170, 367

Holland, J. (1998). *Emergence from Chaos to Order*. Oxford University Press. 190

Holmes, T. (2008). *Electronic and Experimental Music. Technology, Music, and Culture*. Routledge, third edition. 128, 166

Holopainen, R. (2001). Metoder i syntes och signalbehandling. Unpublished Master's thesis, University of Oslo. 88

Holopainen, R. (2007). Nonlinear filters. In *Proceedings of the ICMC 2007*, volume 1, pages 283–286, Copenhagen, Denmark. 135, 136

Holopainen, R. (2009). Feature extraction for self-adaptive synthesis. *Sonic Ideas/Ideas Sónicas*, 1(2):21–28. 15, 43, 208

Holopainen, R. (2010). Self-organised sounds with a tremolo oscillator. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, pages 412–418, Graz, Austria. 105, 286

Holopainen, R. (2011). Logistic map with a first order filter. *International Journal of Bifurcation and Chaos*, 21(6):1773–1781. 136, 219, 220

Holtz, P. (2009). What's your music? subjective theories of music-creating artists. *Musicæ Scientiæ*, XIII(2):207–230. 357, 358

Holtzman, S. R. (1979). An automated digital sound synthesis instrument. *Computer Music Journal*, 3(2):53–61. 9

Holzfuss, J. and Lauterborn, W. (1989). Liapunov exponents from a time series of acoustic chaos. *Physical Review A*, 39(4):2146–2152. 130

Horgan, J. (1995). From complexity to perplexity. *Scientific American*, pages 104–109. 200

Horner, A. (2003). Auto-programmable FM and wavetable synthesizers. *Contemporary Music Review*, 22(3):21–29. 9, 78

Horner, A. (2007a). A comparison of wavetable and FM data reduction methods for resynthesis of musical sounds. In Beauchamp, J., editor, *Analysis, Synthesis, and Perception of Musical Sounds*, pages 228–249. Springer. 98, 109

Horner, A. (2007b). Evolution in digital audio technology. In Miranda, E. R. and Biles, J., editors, *Evolutionary Computer Music*, chapter 3, pages 52–78. Springer. 259

Houtsma, A. (1995). Pitch perception. In Moore, B. C. J., editor, *Hearing. Handbook of Perception and Cognition*, chapter 8, pages 267–298. Academic Press, second edition. 50, 65, 86

Hramov, A., Khramova, A., Khromova, I., and Koronovskii, A. (2004). Investigation of transient processes in one-dimensional maps. *Nonlinear Phenomena in Complex Systems*, 7(1):1–16. 220

Hubler, A. (2012). Is symbolic dynamics the most efficient data compression tool for chaotic time series? *Complexity*, 17(3):5–7. 129

Hunt, A., Wanderley, M., and Paradis, M. (2002). The importance of parameter mapping in electronic instrument design. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*, Dublin, Ireland. 13, 282

Ikeshiro, R. (2010). Generative, emergent, self-similar structures: Construction in self. In *Proc. of the ICMC 2010*, pages 365–368, Ann Arbor. 133, 367

Ishi, C. T. (2004). A new acoustic measure for aspiration noise detection. *Interspeech*, pages 941–944. 66, 68

Iverson, P. and Krumhansl, C. (1993). Isolating the dynamic attributes of musical timbre. *Journal of the Acoustic Society of America*, 94(5):2595–2603. 47

Jacobsen, E. and Lyons, R. (2003). The sliding DFT. *IEEE Signal Processing Magazine*, 20(2):74–80. 54

Jaffe, D. (1995). Ten criteria for evaluating synthesis techniques. *Computer Music Journal*, 19(1):76–87. 112, 307

James, S. (2005). Developing a flexible and expressive realtime polyphonic wave terrain synthesis instrument based on a visual and multidimensional methodology. Master's thesis, Edith Cowan University, Australia. 102

Jennings, H., Ivanov, P., Martins, A., da Silva, P. C., and Viswanathan, G. M. (2004). Variance fluctuations in nonstationary time series: a comparative study of music genres. *Physica A*, (336):585–594. 182

Jensen, K. (2005). Noise upon the sinusoids. In *2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. 96, 97

Jordà, S. (2004). Digital instruments and players: Part I – efficiency and apprenticeship. In *Proceedings of the 2004 Conference on New Instruments for Musical Expression (NIME-04)*, pages 59–63, Hamamatsu, Japan. 31

Jordà, S. (2007). Interactivity and live computer music. In Collins, N. and d'Escriván, J., editors, *The Cambridge Companion to Electronic Music*, chapter 5, pages 89–106. Cambridge University Press. 16, 364

Kant, I. (2003). *Kritik av omdömeskraften [orig. Critik der Urtheilskraft, 1790]*. Thales, Stockholm. 17

Kantz, H. and Schreiber, T. (2003). *Nonlinear Time Series Analysis*. Cambridge University Press, second edition. 120, 121, 122, 179, 180, 182, 268, 271

Karplus, K. and Strong, A. (1983). Digital synthesis of plucked-string and drum timbres. *Computer Music Journal*, 7(2):43–55. 163, 172, 308

Kayn, R. (1996). Sozio-, technologische- und aesthetische Aspekte akustischer Innovation am Beispiel eigener Werke. *http://www.kayn.nl/publications.html*. 349

Kendall, G. (1995). The decorrelation of audio signals and its impact on spatial imagery. *Computer Music Journal*, 19(4):71–87. 239

Kendall, R. and Carterette, E. (1993). Identification and blend of timbres as a basis for orchestration. *Contemporary Music Review*, 9(1-2):51–67. 47

Képesi, M. and Weruaga, L. (2006). Adaptive chirp-based time-frequency analysis of speech signals. *Speech Communication*, 48:474–492. 65

Kersten, S., Maestre, E., and Ramirez, R. (2008). Concatenative synthesis of expressive saxophone performance. In *Sound and Music Computing Conference*. 36

Kitano, M., Yabuzaki, T., and Ogawa, T. (1983). Chaos and period-doubling bifurcations in a simple acoustic system. *Physical Review Letters*, 50(10):713–716. 130, 165

Klapuri, A. (2008). Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):255–266. 65, 312

Klatt, D. and Klatt, L. (1990). Analysis, synthesis, and perception of voice quality variations among female and male talkers. *Journal of the Acoustic Society of America*, 87(2):820–857. 67

Kleczkowski, P. (1989). Group additive synthesis. *Computer Music Journal*, 13(1):12–20. 90

Kleimola, J., Pekonen, J., Penttinen, H., Välimäki, V., and Abel, J. (2009). Sound synthesis using an allpass filter chain with audio-rate coefficient modulation. In *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, pages 305–312, Como, Italy. 245

Kling, G. and Roads, C. (2004). Audio analysis, visualization, and transformation with the matching pursuit algorithm. In *Proc. of the 7th Int. Conference on Digital Audio Effects (DAFx'04)*, pages 33–37, Naples, Italy. 82

Kojs, J., Serafin, S., and Chafe, C. (2007). Cyberinstruments via physical modeling synthesis: Compositional applications. *Leonardo Music Journal*, 17:61–66. 11

Kostek, B. (2005). *Perception-Based Data Processing in Acoustics. Applications to Music Information Retrieval and Psychophysiology of Hearing*. Springer. 52, 75

Krimphoff, J., McAdams, S., and Winsberg, S. (1994). Caractérisation du timbre des sons complexes. II. Analyses acoustiques et quantification psychophysique. *Journal de Physique*, 4(C5):625–628. 47, 49, 67

Krumhansl, C. (1989). Why is musical timbre so hard to understand? In Nielzen, S. and Olsson, O., editors, *Structure and Perception of Electroacoustic Sound and Music*, Excerpta Medica, pages 43–53. Elsevier, Amsterdam. 47

Krumhansl, C. and Kessler, E. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review*, 89(4):334–368. 315

Kuhn, T. (1977). *The Essential Tension. Selected Studies in Scientific Tradition and Change*. The University of Chicago Press, Chicago and London. 40

Kuivila, R. (2004). Open sources: Words, circuits and the notation-realization relation in the music of David Tudor. *Leonardo*, 14:17–23. 166

Kumar, S., Forster, H., Bailey, P., and Griffiths, T. (2008). Mapping unpleasantness of sounds to their auditory representation. *J. Acoust. Soc. Am*, 124(6):3810–3817. 185

Kuramoto, Y. and Battogtokh, D. (2002). Coexistence of coherence and incoherence in nonlocally coupled phase oscillators. *Nonlinear Phenomena in Complex Systems*, 5(4):380–385. 157

Kurz, T. and Lauterborn, W. (1988). Bifurcation structure of the Toda oscillator. *Physical Review A*, 37(3):1029–1031. 149

Lacasa, L., Luque, B., Ballesteros, F., Luque, J., and Nuño, J. C. (2008). From time series to complex networks: The visibility graph. In *Proc. of the National Academy of Sciences*, volume 105, pages 4972–4975, USA. 122

Ladefoged, P. (2005). *Vowels and Consonants*. Blackwell Publishing, second edition. 48, 68

Lakatos, S. (2000). A common perceptual space for harmonic and percussive timbres. *Perception & Psychophysics*, 62(7):1426–1439. 48

Lakoff, G. and Núñez, R. (2000). *Where Mathematics Comes From. How the Embodied Mind Brings Mathematics into Being.* Basic Books. 11, 245

Landy, L. (1991). *What's the Matter with Today's Experimental Music? Organized Sound Too Rarely Heard.* Harwood academic publishers. 14, 23

Landy, L. (2002). La synthèse sonore : enfin l'émancipation ? In *Journées d'Informatique Musicale, 9e édition*, pages 5–15, Marseille. 339

Langton, C., editor (1995). *Artificial Life. An Overview.* The MIT Press, Cambridge, Massachusetts. 17, 200

Large, E. and Kolen, J. (1994). Resonance and the perception of musical meter. *Connection Science*, 6(2 & 3):177–208. 151

Lartillot, O. and Toiviainen, P. (2007). A Matlab toolbox for musical feature extraction from audio. In *Proc. of the 10th Int. Conf. on Digital Audio Effects (DAFX-07)*, Bordeaux, France. 55

Lauterborn, W. and Cramer, E. (1981). Subharmonic route to chaos observed in acoustics. *Physical Review Letters*, 47(20):1445–1448. 130, 210

Lauterborn, W. and Holzfuss, J. (1991). Acoustic chaos. *International Journal of Bifurcation and Chaos*, 1(1):13–26. 130

Lauterborn, W. and Parlitz, U. (1988). Methods of chaos physics and their application to acoustics. *Journal of the Acoustic Society of America*, 84(6):1975–1993. 116, 130

Lazzarini, V., Timoney, J., Kleimola, J., and Välimäki, V. (2009a). Five variations on a feedback theme. In *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, pages 139–145, Como, Italy. 97

Lazzarini, V., Timoney, J., and Lysaght, T. (2007). Adaptive FM synthesis. In *Proc. of the 10th Int. Conf. on Digital Audio Effects (DAFX-07)*, pages 21–26, Bordeaux, France. 79

Lazzarini, V., Timoney, J., Pekonen, J., and Välimäki, V. (2009b). Adaptive phase distortion synthesis. In *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, pages 28–35, Como, Italy. 79, 102

Le Brun, M. (1979). Digital waveshaping synthesis. *Journal of the Audio Engineering Society*, 27(4):250–266. 95, 102

Legge, K. and Fletcher, N. (1989). Nonlinearity, chaos, and the sound of shallow gongs. *J. Acoust. Soc. Am*, 86(6):2439–2443. 130

Lemaitre, G., Susini, P., Winsberg, S., and McAdams, S. (2003). Perceptively based design of new car horn sounds. In *Proceedings of the 2003 International Conference on Auditory Display*, Boston. 47

Lemons, D. (2002). *An Introduction to Stochastic Processes in Physics*. The Johns Hopkins University Press, Baltimore. 124, 247, 250

Lewis, G. (1999). Interacting with latter-day musical automata. *Contemporary Music Review*, Vol. 18, Part 3:pp. 99–112. 31, 346

Lewis, G. (2000). Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal*, 10:33–39. 346

Lewis, G. (2009). Interactivity and improvisation. In Dean, R., editor, *The Oxford Handbook of Computer Music*, chapter 21, pages 457–466. Oxford University Press. 345

Li, T.-Y. and Yorke, J. (1975). Period three implies chaos. *Am. Math. Month.*, 82:985–992. 138, 218

Lindemann, E. (2007). Music synthesis with reconstructive phrase modeling. *IEEE Signal Processing Magazine*, 24(2):80–91. 111

Liou, C., Wu, T., and Lee, C. (2009). Modeling complexity in musical rhythm. *Complexity*, 15(4):19–30. 187

Lipshitz, S., Vanderkooy, J., and Wannamaker, R. (1991). Minimally audible noise shaping. *Journal of the Audio Engineering Society*, 39(11):836–852. 124

Lorenz, E. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20:130–141. 125

Lorrain, D. (1980). A panoply of stochastic 'cannons'. *Computer Music Journal*, 4(1):53–81. 94

Lu, L., Jiang, H., and Zhang, H. (2001). A robust audio classification and segmentation method. In *ACM Multimedia*, pages 203–211. 62, 71

Mackenzie, J. P. (1995). Chaotic predictive modelling of sound. In *Proceedings of ICMC 1995*, pages 49–56, Banff, Canada. 122

Mackey, M. and Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289. 141

Magnusson, T. (2009). Of epistemic tools: musical instruments as cognitive extensions. *Organised Sound*, 14(2):168–176. 29, 359

Magnusson, T. (2010). Designing constraints: Composing and performing with digital musical systems. *Computer Music Journal*, 34(4):62–73. 359, 360

Manning, P. (1993). *Electronic & Computer Music*. Oxford University Press, second edition. 25, 128, 346, 349

Markel, J. (1972). The SIFT algorithm for fundamental frequency estimation. *IEEE Transactions on Audio and Electroacoustics*, 20(5):367–377. 64, 65

Masri, P. and Bateman, A. (1996). Improved modelling of attack transients in music analysis-resynthesis. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 100–103, Hong Kong. 63

Mathews, M. (1995). The esthetic situation and actual view of electroacoustic music performance. In *Aesthetics and Electroacoustic Music*, pages 65–73, Bourges, France. Acteon - Mnemosyne. 336

Mauceri, F. (1997). From experimental music to musical experiment. *Perspectives of New Music*, 35(1):187–204. 4, 5, 21

Mauch, M. and Levy, M. (2011). Structural change on multiple time scales as a correlate of musical complexity. In *Proc. of the 12th International Society for Music Information Retrieval (ISMIR 2011)*, pages 489–494, Miami, USA. 188, 269, 294

Mayer-Kress, G., Choi, I., Weber, N., Bargar, R., and Hübler, A. (1993). Musical signals from Chua's circuit. *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, 40(10):688–695. 144

McAdams, S. (1999). Perspectives on the contribution of timbre to musical structure. *Computer Music Journal*, 23(3):85–102. 46, 49

McAdams, S., Winsberg, s., Donnadieu, S., De Soete, G., and Krimphoff, J. (1995). Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent classes. *Psochological Research*, 58:177–192. 47, 49

McCormack, J., Eldridge, A., Dorin, A., and McIlwain, P. (2009). Generative algorithms for making music: Emergence, evolution, and ecosystems. In Dean, R., editor, *The Oxford Handbook of Computer Music*, chapter 18, pages 354–379. Oxford University Press. 16, 193, 202

McDermott, J., Griffith, N., and O'Neill, M. (2006). Timbral, perceptual, and statistical attributes for synthesized sound. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 179–185, New Orleans. 53, 67, 70

McIntyre, M. E., Schumacher, R. T., and Woodhouse, J. (1983). On the oscillations of musical instruments. *Journal of the Acoustic Society of America*, 74(5):1325–1345. 130

Meilgaard, M., Civille, G. V., and Carr, T. (2007). *Sensory Evaluation Techniques*. CRC Press, fourth edition. 44, 50

Melara, R. and Marks, L. (1990). Interaction among auditory dimensions: Timbre, pitch, and loudness. *Perception & Psychophysics*, 48(2):169–178. 50

Miranda, E. R. (2007). Cellular automata music: From sound synthesis to musical forms. In Miranda, E. R. and Biles, J., editors, *Evolutionary Computer Music*, chapter 8, pages 170–193. Springer. 171

Mitchell, M. (2009). *Complexity. A Guided Tour*. Oxford University Press. 163, 173, 177, 178

Mitsuhashi, Y. (1982). Audio signal synthesis by functions of two variables. *Journal of the Audio Engineering Society*, 30(10):701–706. 100, 102

Moorer, J. (1976). The synthesis of complex audio spectra by means of discrete summation formulas. *Journal of the Audio Engineering Society*, 24(9). 103, 104

Morris, J. (2007). Feedback instruments: Generating musical sounds, gestures, and textures in real time with complex feedback systems. In *Proc. of the ICMC 2007*, pages 469–476, Copenhagen, Denmark. 164, 166

Mountain, R. (2001). Composers and imagery: Myths and realities. In Godøy, R. I. and Jørgensen, H., editors, *Musical Imagery*, chapter 15, pages 271–288. Swets and Zeitlinger. 355, 356

Mumma, G. (1964). An electronic music studio for the independent composer. *Journal of the Audio Engineering Society*, 12(3):240–244. 346

Mumma, G. (1967). Creative aspects of live-performance electronic music technology. *Audio Engineering Society Preprint*, 33rd Convention. 348, 350

Mumma, G. (2002). Live-electronic music. Tzadik TZ7074 (CD). 347, 349, 350

Murail, T. (2005). The revolution of complex sounds. *Contemporary Music Review*, 24(2/3):121–135. 176, 180

Néda, Z., Ravasz, E., Vicsek, T., Brechet, Y., and Barabási, A. L. (2000). Physics of the rhythmic applause. *Physical Review E*, 61(6):6987–6992. 151

Nielsen, J. and Svensson, U. P. (1999). Performance of some linear time-varying systems in control of acoustic feedback. *J. Acoust. Soc. Am*, 106(1):240–254. 165

Nierhaus, G. (2010). *Algorithmic composition: Paradigms of automated music generation*. Springer. 33, 35, 350

Nilson, C. (2007). Live coding practice. In *Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07)*, pages 112–117, New York. 30, 326, 359, 362

Nyman, M. (1999). *Experimental Music. Cage and Beyond*. Cambridge University Press. 21, 22, 23, 349

Ogorzałek, M. (1993). Taming chaos: Part II—control. *IEEE Transactions on Circuits and Systems–1: Fundamental Theory and Applications*, 40(10):700–706. 152

Orio, N., Lemouton, S., and Schwarz, D. (2003). Score following: State of the art and new developments. In *Proceedings of the 2003 Conference on New Instruments for Musical Expression (NIME-03)*, pages 36–41, Montreal, Canada. 26, 345

Ott, E. and Antonsen, T. (2008). Low dimensional behavior of large systems of globally coupled oscillators. *Chaos*, 18:037113. 155

Ott, E., Grebogi, C., and Yorke, J. (1990). Controlling chaos. *Physical Review Letters*, 64(11):1196–1199. 152, 218

Pantaleone, J. (2002). Synchronization of metronomes. *American Journal of Physics*, 70(10):992–1000. 151, 154

Park, T. H., Biguenet, J., Li, Z., Richardson, C., and Scharr, T. (2007). Feature modulation synthesis (FSM). In *Proceedings of the ICMC 2007*, volume II, pages 368–372, Copenhagen, Denmark. 79

Park, T. H. and Li, Z. (2009). Not just prettier: FMS toolbox marches on. In *Proceedings of the ICMC 2009*, pages 211–214, Montreal, Canada. 79

Park, T. H., Li, Z., and Biguenet, J. (2008). Not just more FMS: Taking it to the next level. In *Proceedings of the ICMC*, Belfast, Ireland. 79

Paulus, J. and Klapuri, A. (2008). Acoustic features for music piece structure analysis. In *Proc. of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland. 270

Pecora, L., Carroll, T., Johnson, G., Mar, D., and Heagy, J. (1997). Fundamentals of synchronization in chaotic systems, concepts, and applications. *Chaos*, 7(4):520–543. 151, 153

Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the cuidado project. Technical report, IRCAM. 53, 66, 71, 86, 89

Peeters, G. and Deruty, E. (2008). Automatic morphological description of sounds. In *Acoustics 08*, Paris. 46

Peeters, G., Giordano, B., Susini, P., Misdariis, N., and McAdams, S. (2011). The timbre toolbox: Extracting audio descriptors from musical signals. *Journal of the Acoustic Society of America*, 130(5):2902–2916. 5, 52, 57, 75

Peeters, G., McAdams, S., and Herrera, P. (2000). Instrument sound description in the context of MPEG-7. In *Proc. of the ICMC*, San Francisco. 55

Perkis, T. (2003). Complexity and emergence in the American experimental music tradition. In Casti, J. and Karlqvist, A., editors, *Art and Complexity*, pages 75–84. Elsevier. 8, 19, 24, 193, 358

Pigott, J. (2011). Vibration, volts and sonic art: A practice and theory of electromechanical sound. In *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, pages 84–87, Oslo, Norway. 161

Plomp, R. (2002). *The Intelligent Ear. On the Nature of Sound Perception*. Lawrence Erlbaum Associates, Mahwah, New Jersey. 42

Plomp, R. and Levelt, W. J. M. (1965). Tonal consonance and critical bandwidth. *J. Acoust. Soc. Am*, 38:548–560. 70

Poepel, C. and Dannenberg, R. (2005). Audio signal driven sound synthesis. In *Proceedings of ICMC 2005*, pages 391–394, Barcelona, Spain. 79

Polansky, L., Barnett, A., and Winter, M. (2011). A few more words about James Tenney: dissonant counterpoint and statistical feedback. *Journal of Mathematics and Music*, 5(2):63–82. 275, 280

Pollard, H. F. and Jansson, E. V. (1982). A tristimulus method for the specification of musical timbre. *Acustica*, 51:162–171. 57, 89

Polotti, P. and Rocchesso, D., editors (2008). *Sound to Sense, Sense to Sound. A State of the Art in Sound and Music Computing*. Logos Verlag, Berlin. 52, 53

Pousseur, H. (1959). Scambi – description of a work in progress (1959). *Gravesaner Blätter*, 13. 335

Povel, D. J. and Essens, P. (1985). Perception of temporal patterns. *Music Perception*, 2(4):411–440. 187

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (2007). *Numerical Recipes. The Art of Scientific Computing*. Cambridge University Press, third edition. 119, 226

Pressing, J. (1988). Nonlinear maps as generators of musical design. *Computer Music Journal*, 12(2):35–46. 126, 127, 265, 288, 319, 363

Pressing, J. (1990). Cybernetic issues in interactive performance systems. *Computer Music Journal*, 14(1):12–25. 28

Proakis, J. and Manolakis, D. (2007). *Digital Signal Processing. Principles, Algorithms, and Applications. Fourth Edition*. Upper Saddle River: Pearson Prentice Hall. 59, 200, 214, 247

Prokopenko, M., Boschetti, F., and Ryan, A. (2008). An information-theoretic primer on complexity, self-organization, and emergence. *Complexity*, 15(1):11–28. 177, 179, 192, 197

Ravelli, E., Richard, G., and Daudet, L. (2008). Fast MIR in a sparse transform domain. In *Proc. of the International Society for Music Information Retrieval (ISMIR 2008)*, pages 527–532, Philadelphia, USA. 83

Reich, S. (2002). *Writings on Music, 1965-2000*. Oxford University Press. 166, 168, 336

Reiss, J. D. and Sandler, M. B. (2003). Nonlinear time series analysis of musical signals. In *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*, London, UK. 121

Repetto, D. I. (2004). crash and bloom: A self-defeating regenerative system. *Leonardo Music Journal*, 14:88–94. 164

Ribeiro, H., Zunino, L., Mendes, R., and Lenzi, E. (2012). Complexity-entropy causality plane: A useful approach for distinguishing songs. *Physica A*, 391(7):2421–2428. 123, 176

Risset, J.-C. (1991). Timbre analysis by synthesis: Representations, imitations and variants for musical composition. In De Poli, G., Piccialli, A., and Roads, C., editors, *Representations of Musical Signals*, pages 7–43. The MIT Press, Cambridge. 32, 83, 307

Roads, C., editor (1985). *Composers and the Computer*. William Kaufmann, Inc, Los Altos, California. 171, 346, 361

Roads, C. (1996). *The Computer Music Tutorial*. The MIT Press, Cambridge. 9, 25, 28, 33, 77, 78, 98, 354

Roads, C. (2001). *Microsound*. The MIT Press, Cambridge, Massachusetts. 80

Röbel, A. (2001). Synthesizing natural sounds using dynamic models of sound attractors. *Computer Music Journal*, 25(2):46–61. 122

Rodet, X. (1993). Models of musical instruments from Chua's circuit with time delay. *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, 40(10):696–701. 131

Rodet, X. and Schwarz, D. (2007). Spectral envelopes and additive + residual analysis/synthesis. In Beauchamp, J., editor, *Analysis and Synthesis of Musical Instrument Sounds. The Sound of Music*, chapter 5, pages 175–227. Springer. 67

Rodet, X. and Vergez, C. (1999a). Nonlinear dynamics in physical models: From basic models to true musical-instrument models. *Computer Music Journal*, 23(3):35–49. 136

Rodet, X. and Vergez, C. (1999b). Nonlinear dynamics in physical models: Simple feedback-loop systems and properties. *Computer Music Journal*, 23(3):18–34. 136, 142

Ross, M., Shaffer, H., Cohen, A., Freudberg, R., and Manley, H. (1974). Average magnitude difference function pitch extractor. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 22(5):353–362. 137

Rossignol, S. (2000). *Segmentation et Indexation des Signaux Sonores Musicaux*. PhD thesis, L'Université Paris 6. 85

Rossignol, S., Rodet, X., Soumagne, J., Collette, J.-L., and Depalle, P. (1999). Automatic characterisation of musical signals: Feature extraction and temporal segmentation. *Journal of New Music Research*, 28(4):281–295. 53

Rowe, R. (2009). Split levels: Symbolic to sub-symbolic interactive music systems. *Contemporary Music Review*, 28(1):31–42. 312

Roy, S. (2003). *L'analyse des musiques électroacoustiques : Modèles et propositions*. L'Harmattan, Paris, France. 331, 338, 339, 340

Ruelle, D. (1987). Diagnosis of dynamical systems with fluctuating parameters. In *Proc. of the Royal Society of London*, pages 5–8. 269

Sakaguchi, H. (2008). Synchronization in coupled phase oscillators. *Journal of the Korean Physical Society*, 53(2):1257–1264. 155

Saunders, J., editor (2009). *The Ashgate Research Companion to Experimental Music*. Ashgate. 24

Schaeffer, P. (1966). *Traité des objets musicaux*. Éditions du Seuil, Paris. 14, 24, 41, 42, 43, 51, 176, 324, 325, 326

Schaeffer, P. and Reibel, G. (1998). *Solfège de l'Objet Sonore*. INA-GRM, new edition. 44, 45, 51, 326

Schattschneider, J. and Zölzer, U. (1999). Discrete-time models for nonlinear audio systems. In *Proc. of the 2nd COST-G6 Conf. on Digital Audio Effects (DAFx'99)*, Trondheim, Norway. 134

Schmidhuber, J. (1997). Low-complexity art. *Leonardo*, 30(2):97–103. 184

Schmidhuber, J. (2009). Simple algorithmic theory of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. *Journal of SICE*, 48(1):21–32. 185, 186

Schneider, A. (2000). Inharmonic sounds: Implications as to "pitch", "timbre" and "consonance". *Journal of New Music Research*, 29(4):275–301. 88

Schnell, N. and Battier, M. (2002). Introducing composed instruments, technical and musicological implications. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*, Dublin, Ireland. 31

Schroeder, M. (1986). Auditory paradox based on fractal waveform. *Journal of the Acoustic Society of America*, 79(1):186–189. 131

Schwarz, D. (2000). A system for data-driven concatenative sound synthesis. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*, Verona, Italy. 80

Schwarz, D. (2006). Concatenative sound synthesis: The early years. *Journal of New Music Research*, 35(1):3–22. 73, 80

Schwarz, D. (2007). Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine*, 24(2):92–104. 73, 80

Schwarz, D. (2011). State of the art in sound texture synthesis. In *Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11)*, pages 221–231, Paris, France. 111

Serra, M.-H. (1993). Stochastic composition and stochastic timbre: Gendy3 by Iannis Xenakis. *Perspectives of New Music*, 31(1):236–257. 19, 170, 367

Serra, X. and Smith, J. (1990). Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24. 70, 82

Sethares, W. (2005). *Tuning, Timbre, Spectrum, Scale*. Springer, second edition. 70, 87, 88, 200, 312, 315

Shalizi, C. R., Shalizi, K. L., and Haslinger, R. (2004). Quantifying self-organization with optimal predictors. *Physical Review Letters*, 93(11):118701. 190, 197, 198

Shannon, C. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423. 177, 349

Shepard, R. (1964). Circularity in judgments of relative pitch. *Journal of the Acoustic Society of America*, 36(12):2346–2353. 88

Shmulevich, I. and Povel, D. J. (2000). Measures of temporal pattern complexity. *Journal of New Music Research*, 29(1):61–69. 187, 294

Sigeti, D. and Horsthemke, W. (1989). Pseudo-regular oscillations induced by external noise. *Journal of Statistical Physics*, 54(5/6):1217–1222. 248

Slater, D. (1998). Chaotic sound synthesis. *Computer Music Journal*, 22(2):12–19. 131

Smalley, D. (1997). Spectromorphology: explaining sound-shapes. *Organised Sound*, 2(2):107–126. 325

Smith, J. O. (1991). Viewpoints on the history of digital synthesis. In *Proc. of the ICMC*, pages 1–10, Montreal, Canada. 9

So, C. and Horner, A. (2004). Wavetable matching of pitched inharmonic instrument tones. *Journal of the Audio Engineering Society*, 52(5):516–529. 91, 109

Sommerer, C. and Mignonneau, L. (2003). Modeling complexity for interactive art works on the internet. In Casti, J. and Karlqvist, A., editors, *Art and Complexity*, pages 85–107. Elsevier. 177, 179

Sprott, J. C. (2010). *Elegant Chaos. Algebraically Simple Chaotic Flows*. World Scientific, Singapore. 141, 144, 147, 149, 154, 184

Streich, S. (2006). *Music Complexity: A Multifaceted Description of Audio Content*. PhD thesis, Pompeu Fabra, Barcelona, Spain. 177, 181, 182, 187, 293, 294, 296, 337

Strogatz, S. (1994). *Nonlinear Dynamics and Chaos. With Applications to Physics, Biology, Chemistry, and Engineering*. Westview Press. 26, 116, 118, 148, 217, 219, 287

Strogatz, S. (2000). From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D*, 143:1–20. 155

Sturm, B. (2004). Matconcat: An application for exploring concatenative sound synthesis using Matlab. In *Proc. of the 7th Int. Conference on Digital Audio Effects (DAFx'04)*, pages 323–326, Naples, Italy. 73, 80

Sturm, B. (2006). Adaptive concatenative sound synthesis and its application to micromontage composition. *Computer Music Journal*, 30(4):46–66. 80

Sturm, B., Roads, C., McLeran, A., and Shynk, J. (2009). Analysis, visualization, and transformation of audio signals using dictionary-based methods. *Journal of New Music Research*, 38(4):325–341. 82

Tanaka, A. (2009). Sensor-based musical instruments and interactive music. In Dean, R., editor, *The Oxford Handbook of Computer Music*, chapter 12, pages 233–257. Oxford University Press. 26, 27

Tatlier, M. and Şuvak, R. (2008). How fractal is dancing? *Chaos, Solitons and Fractals*, 36:1019–1027. 182

Taylor, R. (2003). Fractal expressionism—where art meets science. In Casti, J. and Karlqvist, A., editors, *Art and Complexity*, pages 117–144. Elsevier. 182, 183

Tél, T. and Gruiz, M. (2006). *Chaotic Dynamics. An Introduction Based on Classical Mechanics*. Cambridge University Press. 94, 116, 119, 138, 147, 215, 216, 221

Temperley, D. (2001). *The Cognition of Basic Musical Structures*. The MIT Press. 315, 337

Thoresen, L. (2007). Spectromorphological analysis of sound objects: an adaptation of Pierre Schaeffer's typomorphology. *Organised Sound*, 12(2):129–141. 43

Timoney, J., Lazzarini, V., Gibney, A., and Pekonen, J. (2010). Digital emulation of distortion effects by wave and phase shaping methods. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, pages 419–422, Graz, Austria. 103

Todd, P. and Miranda, E. R. (2006). Putting some (artificial) life into models of musical creativity. In Deliège, I. and Wiggins, G., editors, *Musical Creativity. Multidisciplinary Research in Theory and Practice*, chapter 20, pages 376–396. Psychology Press, Hove and New York. 200

Toffoli, T. (1994). Occam, Turing, von Neumann, Jaynes: How much can you get for how little? (a conceptual introduction to cellular automata). *InterJournal Complex Systems*. 171, 196

Toffoli, T. (2000). What you always wanted to know about genetic algorithms but were afraid to hear. (arXiv:nlin/0007013v1 [nlin.AO]). 258

Tolonen, T., Välimäki, V., and Karjalainen, M. (1998). Evaluation of modern sound synthesis methods. Technical Report 48, Helsinki University of Technology, Espoo. 9, 78, 112, 113

Toop, R. (1993). On complexity. *Perspectives of New Music*, 31(1):42–57. 174

Toop, R. (2010). Against a theory of musical (new) complexity. In Paddison, M. and Deliège, I., editors, *Contemporary Music. Theoretical and Philosophical Perspectives*, chapter 4, pages 89–97. Ashgate. 173, 175

Truax, B. (1990). Chaotic non-linear systems and digital synthesis: An exploratory study. In *Proc. of the ICMC 1990*, pages 100–103, Glasgow. 131, 288

Truax, B. (1994). The inner and outer complexity of music. *Perspectives of New Music*, 32(1):176–193. 174

Tzanetakis, G. and Cook, P. (2000). Marsyas: A framework for audio analysis. *Organised Sound*, 4(3):169–175. 55

Ueda, Y. (2000). Strange attractors and the origin of chaos. In Abraham, R. and Ueda, Y., editors, *The Chaos Avant-Garde. Memories of the Early Days of Chaos Theory*, pages 23–56. World Scientific. 125

Valsamakis, N. and Miranda, E. R. (2005). Iterative sound synthesis by means of cross-coupled digital oscillators. *Digital Creativity*, 16(2):90–98. 98, 224

Van Noort, D., Wanderley, M., and Depalle, P. (2004). On the choice of mappings based on geometric properties. In *Proceedings of the 2004 Conference on New Instruments for Musical Expression (NIME-04)*, pages 87–91, Hamamatsu, Japan. 282

Verfaille, V. (2003). *Effets audionumériques adaptatifs : théorie, mise en œuvre et usage en création musicale numérique*. PhD thesis, Université Aix-Marseille II. 6, 53, 54, 62, 64, 67, 75, 81, 82

Verfaille, V. and Arfib, D. (2001). A-DAFX: Adaptive digital audio effects. In *Proc. of the COST-G6 Conf. on Digital Audio Effects (DAFX-01)*, Limerick, Ireland. 5, 81

Verma, T., Levine, S., and Meng, T. (1997). Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals. In *Proc. of the ICMC*, Thessaloniki, Greece. 82

Voss, R. and Clarke, J. (1978). "1/f noise" in music: Music from 1/f noise. *Journal of the Acoustic Society of America*, 63(1):258–263. 182

Wannamaker, R., Lipshitz, S., and Vanderkooy, J. (2000). Stochastic resonance as dithering. *Physical Review E*, 61(1):233–236. 124

Wehn, K. (1998). Using ideas from natural selection to evolve synthesized sounds. In *Proc. of the first COST-G6 Workshop on Digital Audio Effects (DAFX98).*, Barcelona, Spain. 259

Wiener, N. (1961). *Cybernetics: or Control and Communication in the Animal and the Machine*. The MIT Press, second edition. 159, 349

Wiesenfeld, K. and Jaramillo, F. (1998). Minireview of stochastic resonance. *Chaos*, 8(3):539–548. 124, 251

Wishart, T. (1994). *Audible Design. A Plain and Easy Introduction to Practical Sound Composition*. Orpheus the Pantomime. 80, 81, 99, 341

Wishart, T. (1996). *On Sonic Art*. Harwood Academic Publishers, Amsterdam, new and rev. edition. 53, 187

Wolf, A., Swift, J., Swinney, H., and Vastano, J. (1985). Determining Lyapunov exponents from a time series. *Physica*, 16D:285–317. 120, 121

Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644. 179, 195

Wolpert, D. and Macready, W. (2007). Using self-dissimilarity to quantify complexity. *Complexity*, 12(3):77–85. 179

Xenakis, I. (1992). *Formalized Music. Thought and Mathematics in Music*. Pendragon Press, Stuyvesant, revised edition. 32, 34, 94, 170, 311, 338, 367

Ystad, S. (1998). *Sound Modeling Using a Combination of Physical and Signal Models*. PhD thesis, L'Université Aix-Marseille II, France. 90, 111

Yumoto, E., Gould, W., and Baer, T. (1982). Harmonics-to-noise ratio as an index of the degree of hoarseness. *Journal of the Acoustic Society of America*, 71(6):1544–1550. 69

Zaripov, R. K. (1969). Cybernetics and music. *Perspectives of New Music*, 7(2):115–154. 349, 352

Zeraoulia, E. and Sprott, J. C. (2008). A minimal 2-D quadratic map with quasi-periodic route to chaos. *International Journal of Bifurcation and Chaos*, 18(5):1567–1577. 219

Zeraoulia, E. and Sprott, J. C. (2010). *2-D Quadratic Maps and 3-D ODE Systems*, volume 73 of *Nonlinear Science, Series A*. World Scientific, Singapore. 131, 183, 219, 256

Zils, A. and Pachet, F. (2001). Musical mosaicing. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01)*, Limerick, Ireland. 80

Zölzer, U. (2002). *DAFX. Digital Audio Effects*. Wiley. 57, 62, 64, 96, 134