Title
Mixing: Composition Theory and Chaos in an Autonomous Music-Making System

Author
Chung, Matthew F.

Publication Date
2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Mixing: Composition Theory and Chaos in an Autonomous Music-Making System

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Arts

in

Music

by

Matthew F. Chung

Committee in charge:

Professor Shahrokh Yadegari, Chair
Professor Shlomo Dubnov
Professor Miller Puckette

2021

The thesis of Matthew F. Chung is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

Thank you to Shahrokh Yadegari, Shlomo Dubnov, and Miller Puckette for their steadfast guidance and patience throughout the challenging time of this writing.

I would also like to thank my mentor in composition, Thomas DeLio. *Mixing* was composed while studying with him at the University of Maryland, College Park during 2018-19. It would not have come about without my many discussions and lessons with him.

Finally, I would like to thank my family, for always being there.

ABSTRACT OF THE THESIS

Mixing: Composition Theory and Chaos in an Autonomous Music-Making System

by

Matthew F. Chung

Master of Arts in Music

University of California San Diego, 2021

Professor Shahrokh Yadegari, Chair

Our system generates a class of musical works. It does so without human intervention by using Lorenz attractors at multiple time scales to construct several time series for controlling sound synthesis parameters. It is implemented with Python, SuperCollider, and Open Sound Control. An indefinite number of works can be autonomously generated, yet each is unique: each exhibits unique musical form globally and unique sonic transformations locally. We discuss the system in the context of (1) Laske's composition theory and (2) phenomenological perspectives on chaos and self-similarity.

Chapter 1

Music-Making Systems

Throughout the ages, composers have invested in a variety of mechanistic practices.[1] Examples include the talea-based isorhythms of Machaut, species counterpoint of Fux, twelve-tone technique of Schoenberg, chance music of Cage, and countless more following the development of computers. Many fields have emerged that address mechanistic practices in composition, including automated composition, algorithmic composition, generative music, machine improvisation, artificial creativity, and more. A significant reason for interest in mechanistic practices is their potential for *automation*. How would one approach something like the automation of composition? It is not granted that composition is entirely mechanistic—nor entirely computable, a standpoint called *Myhill's thesis*. After all, would human composers concede that rote application of rules suffices as composition? So we must leave room for the possibility of non-mechanistic aspects of composition. Nonetheless, we may still ask to what extent composition is mechanizable. Pursuing mechanization to its fullest extent has led many to *metacomposition*, the design of music-making systems. These systems serve as models for understanding the human compositional act. However, metacomposition also anticipates the advent of nonhuman creativity, when the non-mechanistic becomes computable—ultimately, it concerns not merely virtual human music but the pre-music of other personhoods. Thus, it may be said that, in this chapter, we will explore the eerie world of music-making without composition.

[1]We use the term *mechanistic* in its naivest sense, inspired by Xenakis' usage (1971/1992, chapter 5). Note also that mechanization was a historical precursor to the modern concept of automation. (Nof 2009, p. 14)

1.1 Introduction

We will not go into great depth regarding music-making systems, which would require expounding a new theory. Rather, we will simply propose a provisional terminology and discuss open form.

1.1.1 Provisional Terminology

We here propose a provisional terminology for this thesis. A *music-making system* refers to a conceptual system for making music, and not a physical one. Such a system may be equivalently expressed in different languages. An *implementation* is a particular physical concretization of a music-making system. This requires specifying a particular language(s) (in our case Python, OSC, and SuperCollider), along with a physical means for sound production. A *rendition* is a resulting sound recording (e.g. a SuperCollider playback and recording).

We might further elaborate on this terminology. (1) The system definition should only include essential details. It need only provide the minimal specification necessary to fully express the idea of the composition. The notion of 'full expression' suggests an irreducible, minimum complexity of specification depending on both the complexity of the idea and the choice of language. This can be quantified by algorithmic (Kolmogorov) complexity. Note that, depending on how irreducible an idea is, a system may in fact involve a high amount of specificity. Some compositional ideas involve many explicit details at their outset. Thus, if any feature be removed or added, the system will no longer faithfully represent the idea. (2) An implementation can simultaneously incorporate multiple languages. For example, subsystems might each be implemented in their own languages. However, to cohere as a system, all subsystems must be able to interact with each other, and thus must share some common language. (3) An implementation of a system should be a specification of that system at the scale of, and including, how sound will be produced and controlled. Implementations thus must be specific enough to account for speakers, acoustic instruments, or other physical mechanisms of sound production as well as setting, acoustic space, and environment. (4) A system is by definition abstract and underspecified: it remains at the level of variables and degrees of freedom. Anything from an on-off switch to a parameter knob gives rise to a space of *configurations*, which is merely a specification of parameters in an implementation. (5) A rendition is the unpacking of a particular configuration and is embodied as the final physical production of sound.

In summary, a system is a constraint which determines a class of implementations which each generates a parameter space of configurations that each can potentially generate a distinct rendition. The author emphasizes that this is not meant to be a universal terminology, but rather a provisional one improvised for convenience.

1.1.2 Open Form

Open form arises naturally when discussing music-making systems. *Open form* generally refers to when the large-scale phrasal structure of a musical work, such as the organization of sections or movements, remains undetermined (e.g. unspecified by the definitive score). Structure can emerge organically from small-scale forces and interactions (e.g. interpretation, improvisation) or be stipulated by someone other than the composer (e.g. by performers, either arbitrarily or by applying rules; by machine; etc.). Whether open form means the top-down, largest-scale organization of sections or merely relatively large-scale organization remains ambiguous and seems a matter of defining 'form'. This may involve addressing the duality of 'form' and 'content', and other contextual concerns. Definition of 'form' is addressed in chapter 1 of De Bièvre (2012).

Historically, the term was introduced by Brown, who was inspired by Calder's mobile sculptures. Brown's work *Twenty-Five Pages* (1953) is recognized as the first open form score (Welsh 1994). In the work, the orientation and order of twenty-five pages of sheet music are determined by one to twenty-five pianists. Open form is most closely associated with Brown's work, but examples can be found from Stockhausen, Boulez, Haubenstock-Ramati, and Wolff. For example, Wolff's *Duo for Pianists II* (1958) involves 'blocks of material' which have no fixed ordering but are chosen by performers based on cues at the end of the preceding block (Pritchett 1993)—rather like a state-transition machine.

In a composition theoretic context, one can think of open form as the offloading of organizational work from the composer: as Collins (2009) remarks, 'Open form and indeterminacy are just ways of invoking where and whose human effort is applied.' De Bièvre likewise writes that 'openness of the form has all to do with how much responsibility the composer is willing to delegate to the performer(s)' (De Bièvre 2012). It follows, then, that to analyse a work of open form the source of that effort and responsibility should be addressed: on this, Andersen (2020) writes that 'as form in these works emerges in real time, it is the performance—and not the score—that must be central to the analysis', echoing a similar observation by Lochhead (2015). Note that open form is a quality only apprehensible from multiple listenings (as discussed in the section on self-similarity).

In the context of computer music that does not involve performers, i.e. autonomous or 'closed' music-making systems, open form is restored as the designer's task. In part, this is because the score and performer are merged within the system. For instance, we can model the performer's choices as random selection events, e.g. a Gaussian distribution, Markov chain, etc. How one chooses to autonomously handle open form is a subtle art. The 'autocracy' of autonomous systems eliminates many layers of checks and balances of human intuition and evaluation, especially that of trained performers. Because of this, the designer of autonomous systems must analyze beforehand how decisions are to be handled. This often involves systematic testing, evaluation, and taxonomization of parameter spaces for their aesthetic potentials, and devising some method of consistently arriving at musically worthwhile results. Ideally, the system itself would be equipped with some autonomous method of carrying this out. Thus, we see how open form, as a problem of automation, involves once again identifying cognitive compositional processes. A particularly relevant discussion of open form for algorithmic composition is Collins (2009).

For more on open form, see Welsh (1994), De Bièvre (2012), and Andersen (2020).

1.2 The Problem of Autonomy

Metacomposition

Metacomposition is the act of designing music-making systems, building on Laske's usage in Laske (1980). In truth, metacomposition is not about automating composition, i.e. offloading the human compositional act to machines. It is about designing composers, i.e. human interaction with machines such that autonomous music-making systems emerge which may not reflect human compositional acts. We must acknowledge that by excluding human intervention, we exclude certain aspects of human composition, given the current state-of-the-art in AI.

System Autonomy

We can describe a system as falling on a *spectrum of autonomy* with respect to music-making, building on Fernandez and Vico (2013). We can say the spectrum ranges from lower degrees of autonomy (e.g. tools aiding human composition) to higher degrees (e.g. closed systems which require no further human intervention to make music). Autonomy is the highest degree in this spectrum, and suggests full automation[2] of the composer's cognitive processes. This builds on Charles Ames' introduction of the term 'automation' in his unpublished book, *Automated Composition* (c. early 1980's).[3] However, Ames' original criterion for automated composition encompasses 'all compositional processes which are systematic enough to be implemented as computer programs', i.e. *computability*. Contemporary albeit controversial theories such as Tegmark's computable universe hypothesis (Tegmark 2008) pose issues for the criterion of computability. Our criterion is constrained to system *autonomy*, i.e. music-making systems with no human intervention. Note that to arrive at the more natural concept of autonomy, we must first adopt a systems perspective. Design of autonomous music-making systems is well established as a practice. Pioneers of autonomy include Gottfried Michael Koenig and Otto Laske. A concise historical review is the paper *Automated Composition in Retrospect:*

[2] 'Automation, in general, implies operating or acting, or self-regulating, independently, without human intervention' (Nof 2009, p. 14).

[3] See http://www.charlesames.net/automated-composition/index.html.

1956-1986 (Ames 1987). However, we will see theoretical concerns remain.

A Contemporary Perspective on Automation

Autonomy often arises as full automation. Let us briefly reflect on automation. A remarkable reference is the *Springer Handbook of Automation* (Nof 2009). We will not delve into definitions or history, which are discussed there. Automation can be situated in cybernetics or control theory, and is fundamentally constrained by considerations of computability and complexity. A subtler issue is that of motivations for automating, which belong in a broader discourse on energy and values. For example, how automation reduces human effort is deeply related to thermodynamics, human biology, and economics. The sociocultural and ethical considerations of automation cannot be neglected. Without a systematic understanding of what automation accomplishes physically, we cannot sensibly undertake automation in human systems with physical consequences. As we continue zooming out to the most philosophical contexts, pressing issues remain. For example, different views of how humans and machines interact lead to very different views of what can be or should be automated. In particular, holistic paradigms like distributed cognition or Haraway's cyborg (Haraway 1985) give rise to very different information processing models than dualistic paradigms, and thus very different possibilities of what might be automated. We ultimately need a philosophical framework to help us integrate (or dispel the dualistic illusion of) the worlds of humanity and machinery.

A Contemporary Perspective on Composition

Recent attempts at a scientific understanding of composition itself might include works like Herbert Simon's *Science of the Artificial* (1969), and fields like scientific aesthetics, information dynamics, complex systems science, and artificial creativity. However, a longstanding issue is that some internal acts of composition remain inaccessible to external observation. Many are only accessible through introspection, and the empirical validity of introspection has yet to be fully restored. An example of an introspectively derived theory is Giulio Tononi's integrated information theory, a phenomenological model of human consciousness that is extremely promising. The task of modeling the act of composition remains complex and subtle (and it would seem to lie downstream of an understanding consciousness). Furthermore, modeling composition presupposes definitions of art-objects and art-makers, of art-making and art-experiencing, and theories explaining how these are all inextricably linked.

Philosophies of Art

We will also briefly remark here that how we define what the compositional act brings into existence affects both metacomposition and composition theory. One aspect is that the creative act is not wholly concentrated in the compositional act, but also distributed in the act of listening. We see this in the opening quote in Laske (1989) from David Lewin: 'Since "music" is something you *do*, and not just something you *perceive* (or understand), a theory of music can not be developed fully from a theory of musical perception' (Lewin 1986). The paper this quote comes from discusses the phenomenology of music perception, and how experiencing music is a creative act. In fact, some have entirely refocused on the act of experiencing: in *Art as Experience* (1931), John Dewey developed a philosophy of art centered on the idea of *aesthetic experience*. In his writings we find the same notion of experience as an active, creative process: 'For to perceive, a beholder must *create* his own experience. And his creation must include relations comparable to those which the original producer underwent. But with the perceiver, as with the artist, there must be an ordering of the elements of the whole that is in form, although not in details, the same as the process of organization the creator of the work consciously experienced' (Dewey 1931). Note that, in his work, Dewey distinguishes between passive 'undergoing' and active 'doing', similar to Lewin's passive 'perceiving' and active 'doing'. For a more recent survey on aesthetic experience in contemporary philosophy, we recommend Goldman (2013).

Myhill's Thesis

Myhill's thesis is the critical observation by Myhill (1952) that *not all compositional acts might be computible*. Kugel (1990) named this position after Myhill (1952), and develops the point more rigorously with limiting computing (Balaban et al. 1992, chapter 2). This statement need not imply Cartesian dualism or a departure from physicalism. However, given the debates on computability of the universe and mind, we might simply suggest that not all compositional acts are mechanistic (e.g. deductive from finitely axiomatizable systems). Ultimately, we must better understand human cognition to address what Myhill's thesis is getting at. For example, humans inductively infer patterns from immense quantities of sensory data. This gives rise to memory, subjective experience, and perhaps intuition, which all can play a role in the composition act. Induction and free will would seem pathologically non-mechanizable, although machine learning is beginning to approach 'computerized induction'. In particular, there remain potential unknowns about consciousness and personhood: some cognitive processes seem unique, spontaneous, irreducible, private, and often subconscious in a way inaccessible to introspection without splitting focus.

Other Composition Theories

Before proceeding to Laske's composition theory, we will briefly mention some other relevant composition theories. Recently in contemporary philosophy of art, Nick Zangwill has developed a theory of aesthetic creation (Zangwill 2007). This is perhaps the closest to what we have in mind when talking about a 'theory of art-making'. Otherwise, the ancient Greeks discussed concepts such as *poiesis* (production), *physis* (emergence), and *techne* (craft). Heidegger discusses the creation of art in *On the Origin of Art* (1950). In complex systems science there is the concept of *emergence*, referring to the phenomenon of order arising from disorder and wholes becoming greater than their parts. Emergence is the topic of John Holland's book *Emergence* (Holland 1999) and also innumerable other writings in complex systems science and complex adaptive systems. In fact, Chadabe writes that Laske was deeply influenced by Bertalanffy's general systems theory, in chapter 4 of Laske (1999). For more on generation and self-organization in complex systems science, we refer the reader to the *dynamic generative systems* paradigm proposed in John Holland's *Signals and Boundaries* (Holland 2011), and the agent-based modeling and simulation paradigm (Wilenski 2015). These are promising frameworks for studying and understanding emergence, which may yield insight into art-making.

1.3 Laske's Composition Theory

Otto Laske

In an introduction to Laske's work, chapter 4 of Tabor (1999), Joel Chadabe remarks that Laske's most prominent lifelong emphasis is on *theories of how music is made*: the cognitive processes of the composer, how they are identified and extracted as algorithms, and then offloaded to other machines. Laske's theory of composition will serve as the framework for our following discussion. He has written particularly compelling work in the vein of complete automation. In part this is due to the breadth of disciplines across which his insights are synthesized, spanning cognitive science, psychology, linguistics, systems science, computer science, artificial intelligence, and continental philosophy. A broad interdisciplinary synthesis is necessary to address the extremely challenging philosophical situation presented earlier. No narrower of an approach will yield a framework that is consistent across so many fields of knowledge. In conjunction with theorizing, he is also a creator of art, including poetry, composition, and painting. Having a creative background is indispensible because one needs first-hand experience to introspectively study the act of composition. Valuable references on Laske include Jerry Tabor's *Otto Laske: Navigating New Musical Horizons* (1999), the book *Understanding Music with AI* (Balaban et al. 1992), Laske's many papers and books published over the decades, and the Otto Laske Papers (1964-2010) collection at Schneider Music Library of Texas State University.

Composition Theory

Laske defines *composition theory* as 'a theory of the processes by which imagined (virtual) music becomes materially (sonically) or symbolically (notationally) real' (Laske 1989). He further poses that the central question of the discipline is 'How can one build an artificial agent that composes music?' (Laske 1991, p. 239). To answer such a question, what subject matter is to be studied in composition theory? On this Laske writes 'This discipline could deal with the mental processes involved in composing, or it could focus on the structures delt with by such processes. At its most adventurous the discipline could attempt to build a model of the interactions between these two things [...] modeling the interaction between the observable processes taking place in composition, on the one hand, and the structures resulting from the composition act, on the other. Such modeling demands that the task environment in which composition is pursued, i.e. the set of operands and operators used, be clearly identified. To do so in compositional research is difficult since it lies in the nature of the (truly) compositional act not to be pursued in standardized task environments. Rather, the design of the elements that define a composer's task environment is an integral part of the compositional design' (Laske 1980). Important papers expounding composition theory include Laske (1980, 1989, 1990, 1991).

1.3.1 Central Concepts

We will here introduce some of the central concepts from Laske's exposition of composition theory.

Task Environments

Laske defines a *task environment* as 'the totality of tools and communications that form an artist's habitat' (Laske 1990). We might broadly construe this to include hardware and software, but also acoustic instruments, notation systems, scores, and even pencil and paper. Ultimately, it would include any conceptual or physical system that features in compositional thought or action. Just as composers invent, explore, and familiarize themselves with tools, so they do with assemblages of tools into task environments. Laske emphasizes that a task environment is necessary for composition: 'composition is unthinkable without tools, both of the mind and the hand' (Laske 1991). In this way, task environments embody compositional practice, and give actual, tangible form to potential compositional thought. Ultimately, Laske's introduction of task environments allows us to study the interaction between composer and tools, and thereby 'trace' the composition process (Laske 1991).

A task environment can be interpreted loosely as a practical concept. However, it can also be rigorously modeled discretely as a *language*. Upon defining a finite number of tools, and fixing a finite number of ways of using them, we arrive at a limited number of possible changes. This syntax of tool functions is very much like a linguistic constraint—Koenig calls this an 'objectified grammar'. Hence, a composer can be said to be constrained by an objectified grammar of possible decisions. Especially illustrative of this linguistic aspect are task environments like IDE's, DAW's, QWERTY keyboards, and piano keyboards. Thus, we might discretely model a task environment as a language, especially if we can invent useful typologies or coarse-grainings of continuous degrees of freedom.

There is still much to be said on task environments. For example, how does a composer choose tools? How do we explain what new interactions with other tools emerge? How might we think of task environments as *complex systems* of interacting tools? It may be useful to discuss classes of closely related task environments. This is because in practice a composer might add or remove nonessential tools without changing the essential identity of the task environment. However, a composer ultimately cannot work with an amorphous task environment, continually adding and removing tools. Otherwise, one task environment could morph into any other, like Thales' ship. In the end, choice of tools is part of the compositional process and cannot be excluded. This brings us down to the basic question of what tools are. Tools have a definite, reliable function. However, how tools are seen can change the 'syntax' of their use. A deeper study of tools themselves is important for understanding task environments.[4]

Laske's portrayal of task environments almost suggest an ecological perspective (e.g. 'habitat'), as if they were a composer's ecological niche. It might be valuable to approach our questions from the field of ecology, especially theories of niche construction and coevolution. A brief introduction is Schoener (2009).

[4]For example, Heidegger's philosophical writings on tool analysis, e.g. *Being and Time* (1927) and *The Question Concerning Technology* (1953).

Example-based and Rule-based Composition

Laske distinguishes between two paradigms of composition: *example-based* and *rule-based*. We quote a clear passage here: 'The use of computers in composition has led to a gradual replacement of example-based composition (E. Blevis's term) by rule-based composition. Example-based composition relies on examples abstracted from remembered music. Rule-based composition relies on analysis of compositional processes which specifies setes of computational objects. (A subtype of rule-based composition is constraint-based composition [C. Ames's term].)' (Laske 1990). These are not opposites, but rather mutually incommensurable forms of model-based composition, as is more subtly articulated and developed in Laske (1991).

Example-based composition was the norm for common-practice era music. As hinted earlier, because of this compositional mode's reliance on memory of heard examples—and thus selective and private experience—example-based composition is challenging to model. Laske does not focus as much on example-based composition in his writings. This is notable given the current trends of machine learning, which are decidedly example-based. For example, neural nets train on databases of music examples: the examples are simplified and internally represented as features which are transformed, recombined, and reconstituted into new music. As an illustration, the music group DADABOTS' neural nets can indefinitely generate always-new 'neural heavy metal' constituted from databases of music examples. Perhaps Laske's composition theory seems not as currently relevant because of its lack of emphasis on example-based composition. However, his fundamental approach and observations do apply widely, and concern all acts of composition. Thus, we call for further development of composition theory for example-based composition. We refer to Laske (1991, 1992) and his references for more on example-based composition.

Laske writes much more on rule-based composition. He emphasizes that 'rule-based compositional thinking is not based on the analysis of existing music, but on an awareness, if not an analysis, of one's own compositional process' (Laske 1989). He distinguishes three subcategories: 'There are three different approaches to rule-based composition: interpretive, stipulatory and improvisational [...]. Interpretive composition is based on the interpretation, by the composer, of computer-generated data, while design-based composition yields a definitive score based on highly detailed input specifications; improvisational composition starts from a rough outline which is elaborated by the composer/performer during a concert performance' (Laske 1990). An example of interpretive rule-based composition would be mapping integers outputted by a program to musical elements, which is discussed in Laske (1990). Examples of improvisational rule-based composition can be found in the works of Christian Wolff and Earle Brown. An example of stipulatory composition would be Laske's *Furies and Voices* (1992), which is discussed in Laske (1992).

1.3.2 Illustrations

We will discuss the rule-based work of two composers, Xenakis and Koenig. Xenakis may be said to have applied computer-aided algorithmic composition using stipulated rules of interpretation, whereas Koenig very nearly approached autonomous algorithmic composition with *Project Two*.

I. Xenakis

Xenakis is well known for his pioneering work in stochastic approaches, and exploring the possibility of their simulation on computers. Early pieces such as *Pithoprakta* (1956), *Achorripsis* (1957), and *Morsima-Amorsima* (1962) incorporate mechanistically generated data along with procedures to stochastically organize sound. However, Xenakis still uses these in conjunction with his own intuitive, subjective design. In *Formalized Music* (1971) chapter 5, he discusses the use of computer programs to perform what would have been tedious calculations by hand, emphasizing the general principle of offloading mechanical processes onto computers. He writes that 'The creative thought of man gives birth to mental mechanisms, which, in the last analysis, are merely sets of constraints and choices. [...] Certain mechanizable aspects of artistic creation may be simulated by certain physical mechanisms of machines which exist or may be created.’ This observation foreshadows algorithmic composition. It also suggests the need for a parallel understanding in cognitive science: how to identify mental music-making processes (e.g. through empirical study of composers or introspection) and how they are transferred out of the mind (e.g. externalization, extended mind). For more on cognitive and computational offloading, some recent references include Risko (2016) and Dunn (2016).

G. M. Koenig

As Xenakis turned to the computer for stochastic calculations, so did Koenig for serialist calculations. He programmed *Project One* (PR-1) from 1964 to 1970, a tool for generating variations of musical elements and notes. A human can then analyze these for potentials and incorporate into composition. Koenig calls PR-1 a closed program because ‘the composer can exert very little influence on the program itself’ (Roads 1978). PR-1 led to *Project Two* (PR-2) (1966-1970), a tool which allowed the user to more fully control the compositional syntax of musical elements. PR-2 presents the user with a ‘questionnaire of more than sixty questions. According to those questions, which refer to the musical material and the rules, the program would combine or compose a piece.’ (Roads 1978) Thus, PR-2 is an example of stipulatory or design-based composition. We see the act of composition beginning to de-center on the human and re-center on the machine, in an almost post-anthropocentric turn. Analogous to Xenakis’ premise that calculation can be offloaded, we see with PR-2 that design, too, can be offloaded. This extends as far as the design of the entire composition—all that remains is the initial choice of how to specify the ‘identity’ of the piece in a space of pieces, which is determined by the user’s initial variables. References include Roads’ interview with Koenig (Roads 1978), Laske’s composition theoretic discussions of PR-1 and PR-2 (Laske 1981a, 1981b, 1989), and Koenig’s own website, which as of this writing is hosted at http://koenigproject.nl.

Chapter 2

Chaos

2.1 Chaos in Musical Practice

In an interview with Banfield and Peitgen in 1991, Xenakis remarks that it is the 'aliveness of the instrumentalist that gives something to the music'. Perhaps we can hear an aliveness in Xenakis's stochastics or Cage's indeterminacy. How might one give the impression of aliveness with a non-living music-making system? We will explore how chaos can furnish this quality.

2.1.1 History and Criteria of Chaos

We will briefly present a history and discussion of definitions of chaos. An excellent account of chaos is Motter and Campbell (2014). For a historical and qualitative overview of chaos, we recommend Gleick (1987). For technical introductions to chaos, we refer to Lorenz (1963), Sundbye (2018), Strogatz (2015), and Ott (2002). For recent discussion on criteria of chaos, we refer to Zuchowski (2017).

History

Henri Poincaré encountered chaotic behavior in his study of the three-body problem as early as the 1880's. He did not fully seize upon the phenomenon, but did identify sensitive dependence on initial conditions (SDIC). Later in the 1960's, Edward Lorenz rediscovered SDIC due to computer rounding errors while running computer simulations of ODE models of atmospheric convection. In his now classic 1963 paper, he developed a minimalistic model exhibiting SDIC (the Lorenz attractor) and discussed its implications. This was the first full identification of chaos, but it was not appreciated until the work of Li and Yorke in 1975. Chaos is now an established and active field. Promising directions include reaching consensus on a criteria and definition (e.g. Zuchowski 2017), developing approaches for control and prediction (e.g. Ott et al. 1990; Tang et al. 2020), and integrating with fields like complex systems.

Criteria

Defining chaos remains an active area of research. There exist many working definitions, all tending to reflect common intuitions about what counts as 'chaotic'. However, these intuitions can be inherently ambiguous, admitting many possible independent formalizations. Some have chosen to approach this issue inclusively, via phenomenology[1]. A phenomenological approach is particularly appealing to artists, who may primarily value the effects of chaos. Thus, we will elaborate on this approach. We will avoid mathematics while striving for an accurate portrayal, especially drawing on the work of Zuchowski (2017). They suggest there are five phenomenological criteria: (1) determinism, (2) transitivity, (3) (quasi-)periodicity, (4) aperiodicity, and (5) SDIC.

(1) *Determinism* means the present state uniquely specifies the future state. This implies that same inputs yield same outputs. It also implies that, upon choosing an initial state, all future states are determined. Since we will discuss the distinction between chaos and 'randomness', it is valuable to also introduce indeterminism. Indeterminism means that the present state allows multiple future states, i.e. underspecification in time. Often indeterminism is described probabilistically. It is also often ascribed to some source of 'randomness'. However, in a deterministic full context, 'randomness' does not imply metaphysical causelessness (e.g. Epicurus' atomic swerves) but rather epistemic uncertainty. In other words, indeterminism can arise in a model because the model does not account for all sources of specification, due to descriptive coarseness[2]

[1]This is the term commonly used in philosophical discussions of chaos theory, e.g. Zuchowski (2017), to describe intuitive aspects. Sometimes intuitions are called 'pretheoretic', e.g. Werndl (2009). For our purposes, *phenomenological* will refer to intuition derived from subjective, qualitative, first-hand experience.

[2]One may have used insufficiently many variables to capture a deterministic system's degrees of freedom.

and/or narrowness [3].

(2) *Transitivity* can be variously defined but generally describes how phase space is eventually 'filled out' by trajectories. For example, one form is topological transitivity, 'the existence of least one trajectory which visits all possible subregions of an appropriate region of phase space' (Zuchowski 2017, p. 54). We can note the similarity of topological transitivity to, say, denseness: 'any region [...] eventually visits every region in phase space' (Werndl 2009, p. 11). Transitivity is closely related to concepts like denseness and ergodicity, which tend to cohere as one intuition.

(3) *Quasi-periodicity* means behavior that is approximately (within arbitrary bounds) repeated after finite time. In contrast, periodicity here means behavior that is repeated exactly after finite time.[4] Quasi-periodicity captures the intuition that trajectories recurrently trace out certain shapes, giving rise, for example, to the distinctive structure of a Lorenz attractor.

(4) *Aperiodicity* means behavior that never exactly repeats. For example, this describes how each point in a Lorenz trajectory is unique. If even a single point were to recur twice, then because of determinism it would have to follow out the same path as before and reach itself once again. That would lead to a finite loop, and thus periodicity.[5] Hence, by this simplified account, chaos can approximately but never exactly repeat.[6]

(5) *Sensitive dependence on initial conditions* (SDIC) means that microscopic differences in initial conditions propagate into macroscopic differences in solutions, i.e. the butterfly effect. This can be described as global unpredictability. In Lorenz's own words: 'when the present determines the future, but the approximate present does not approximately determine the future' (Danforth 2013). This is of great practical significance because we often don't know the exact states or initial conditions of physical systems (e.g. the weather). SDIC can be formalized with

[3] One may have drawn a boundary around an *under*determined system that omits sources of specification.

[4] Zuchowski used *periodicity* for the third criterion, in part because Devaney and others call for forms of strict periodicity. To avoid confusion, we use *quasi-periodicity*. This aligns with their discussion of weak periodicity.

[5] This can be described as local unpredictability, bearing in mind that determinism in fact does allow us to predict future states from present states.

[6] We will not cover discrete forms of chaos, which also exist.

Lyapunov exponents as the exponential divergence of arbitrarily close trajectories over infinite time. However, there are many caveats to this description over finite times (e.g. Werndl 2009, p. 11-12). SDIC is also related to phase space stretching and folding, or 'kneading'. SDIC is perhaps the most distinctive feature of chaos.

Devaney chaos is an early attempt by Robert Devaney to formalize the criteria of chaos. It is defined for a dynamical system as follows: (1) denseness (of periodic points), (2) transitivity, and (3) SDIC (Devaney 1989). Werndl (2009) proposes defining chaos as *mixing*, which implies (1) transitivity and (2) SDIC. They demonstrate this in fact captures all 'pretheoretic' intuitions.

Lorenz Attractor

The Lorenz attractor is a particularly well-known chaotic dynamical system, and was first presented in the classic paper Lorenz (1963). It is a simplification of a convection model studied by Saltzman (1962). Much can be said and has been written on the properties of the Lorenz attractor; the book Sparrow (1982) is dedicated entirely to the topic.

The Lorenz system of ordinary differential equations are

$$\frac{dx}{dt} = \sigma(y - x)$$

$$\frac{dy}{dt} = x(\rho - z) - y$$

$$\frac{dz}{dt} = xy - \beta z.$$

The Lorenz triple of coefficients (σ, ρ, β) is a point in parameter space. Each point is associated with a phase space, containing solution functions $(x(t), y(t), z(t))$. Different initial conditions $(t_0, x(t_0), y(t_0), z(t_0))$ result in different trajectories. An illustration is shown in Fig. 2.1.

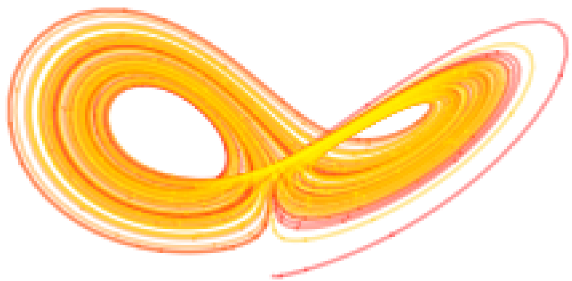

Figure 2.1. An illustration of the Lorenz attractor.

2.1.2 On the Aesthetic Value of Chaos

The value of chaos for artists requires deeper philosophical investigation and broader interdisciplinary synthesis. A historical study of chaos in music theory and composition is Salter (2009).

A Phenomenological Approach

Because artists are concerned with aesthetic experiences, they tend to value perceived effects of phenomena rather than scientific analyses. For example, figurative painters study artistic anatomy, not medical anatomy. This difference in concern is due to difference in goals between artists and scientists. In undertaking the goal of aesthetic creation, the artist is concerned with whatever is needed to create the desired effect. Of course, creating truly interesting effects often requires a great amount of knowledge, such as color theory for painters; but this is not colorimetry. Thus, to account for the effective qualities of chaos that might appeal to an artist, we have taken a phenomenological approach. We will offer a personal explanation for two qualities in particular. Let us begin by reflecting on the nature of chaos, and we will see how they arise.

Chaos is not interchangeable with ‘total randomness’, because we undeniably see macroscopic structure. Yet it seems impossible to distinguish between chaos and randomness on a reductionistic basis, because both are aperiodic. Neither, of course, is chaos ‘total orderedness’, because there is continually unprecedented information. Yet it seems impossible to say that the macroscopic structure we witness in chaotic phenomena can be accidental and not caused. Thus, we confront a gap between total order and total randomness: the paradox of order emerging from chaos. This resembles the paradox posed more generally by complex systems, i.e. of the whole being greater than the sum of its parts. For example, life is not reducible to any random configuration of its parts, but is some special configuration of them. It is exactly the aliveness of chaos that appeals to the composer.

We might even venture a psychological explanation: chaos can be readily anthropomorphized. This might explain chaos's appeal because, psychologically, humans value the impression of intelligent presence. After all, self-expression is often thought of as communication of the artist's emotions, experiences, etc. which inherently reflect the artist's person-hood. In fact, simulating emotional intelligence may be seen as a challenge central to broader composition theory: how can an autonomous music-making system elicit emotional responses when it is incapable of emotions? As another example, understood intent plays a role in prediction. This direction might be investigated with theory of mind (TOM) or ‘thinking through other minds’ (TTOM) (Veissière 2020). Our personal anthropomorphic, psychological explanation is that the Lorenz attractor has a ‘personality’ to which it remains faithful in continually new ways, i.e. ‘spontaneously’. Personality will explain why a composer would not interchange a Lorenz trajectory with a random ordering of the equivalent set of values. Spontaneity will explain why a composer would not interchange a real, unpredictable Lorenz trajectory with a predictable stereotype of itself. Let us delve deeper into these two perceptual qualities.[7]

[7] Note that many other phenomena have ‘personality’ and ‘spontaneity’, e.g. Markov chains, pseudorandomness, etc. By our explanation, these would all be interchangeable.

Personality (Complexity)

Personality describes the distinctiveness and regularity of emergent macroscopic structures, and is a qualitative aspect of complexity. Although macroscopic structure can be described in different ways, one easy was is *positive algorithmic complexity*. In contrast, algorithmic randomness is what humans might intuitively think of as 'pure randomness'. Note that positive algorithmic complexity is not sufficient as a definition of chaos (many nonchaotic signals have positive algorithmic complexity!). (Batterman 1993) Thus, chaos's personality might at least be partially described by positive algorithmic complexity. On a more quantitative note, we might venture that, in conservative dynamical systems, it is precisely conservation of energy in conjunction with a particular measurable transformation that gives rise to macroscopic structure.[8]

The study of chaos's personality/complexity might intersect with fields like information dynamics (Abdallah et al. 2009) and artificial/computational creativity (Boden 1998). For example, in information dynamics, the relationship between musical complexity and aesthetic 'goodness' is often modeled as an inverse-U curve. On this curve, both too little and too much musical complexity will fail to capture an audience's interest, which is optimally captured by music of intermediate complexity. This model was preceded by Fechner's 'principle of aesthetic middle' (Cupchick 1986) in scientific aesthetics. Chaos falls between the extremes of complexity.

Spontaneity (Unpredictability)

Spontaneity describes the continually new ways that personality is brought about, and is a qualitative aspect of unpredictability. Complexity and predictability are distinct but nonetheless deeply related. For example, the relationship between musical predictability and aesthetic 'goodness' can also be modeled as an inverse U-curve. Music that too frequently subverts or fulfills an audience's expectations will fail to capture their interest. Expectation is central to time-based media like music, and is the subject of much current research, e.g. Koelsch et al. (2019), Pearce et al. (2011), Huron (2006). Predicting chaos is also a subject of much current

[8]This is in fact related to the nonequilibrium thermodynamics of computation, and the 'driving' of energy through algorithms, as well as optimization.

research (Tang et al. 2020). Hence, from the perspective of expectation, we must emphasize that chaos serves two different roles: (1) shaping the way that expectations are formed (which it will do predictably), and (2) as a means of controlling the fulfillment or subversion of those expectation (which it will do unpredictably). In other words, we might expect the Lorenz attractor shape to emerge in space, but we cannot expect how it will emerge in time. Quantitatively, the unpredictability of chaos is treated by Werndl (2009), which concludes that 'for predicting any event all sufficiently past events are approximately probabilistically irrelevant'. The value spontaneity has for the artist is that it stimulates the audience's senses: it is the diversity in 'unity and diversity'.

Symbolism

We will briefly add that chaos might be presented as a *symbol* of aliveness, whether or not it actually has anything to do with aliveness. This emblematic or stereotypical usage might take root in the field of computer music as a custom. On this note, there are suggestive analogies between chaos and free will, noted by various philosophers, e.g. Bishop (2002), Garson (1995). Salter (2009) also acknowledges the same suggestive analogy.

2.2 Self-similarity of Music

Chaos is deeply related to self-similarity. Let us turn to the definition and musical properties of self-similarity.

2.2.1 Definition and Musical Context

Self-similarity is a resemblance between a whole and its parts. In its original geometric sense, it is similarity[9] between the shape of an object and the shape of its constituent objects. Thus, it is a property that can be verified visually. Classic examples include the Sierpinski gasket and Koch curve. In its broadest sense, self-similarity is defined as *invariance under scaling*. A

[9]In the Euclidean geometric sense, i.e. congruence after uniform scaling.

distinction is often made between exact and statistical self-similarity, where the former implies strict equivalence after scaling and the latter implies approximate equivalence based on some similarity measure. An accessible scientific reference on self-similarity is Peitgen et al. (2006, 2013).

The musical world's increasing interest in self-similarity has paralleled the scientific world's. Of musical literature treating self-similarity, Pareyon (2011) delineates two veins: (1) the practical use of self-similarity in matters of 'engineering' such as design or composition, and (2) the analysis of self-similarity from musicological perspectives including those historic, cultural, aesthetic, and cognitive. Pareyon's bibliography includes further references on self-similarity in music. Yadegari (1992) and Pareyon (2011) are representative works from the design and analysis veins respectively and are introduced below.

Yadegari (1992) explicates a technique and software implementation for synthesizing a class of self-similar signals using multiple layers of nested cells in a way resembling L-systems. In later work he refers to this technique as recursive granular synthesis. The resulting sonic material is highly distinctive in its crystalline structure of evolution, as can be heard and seen by spectrogram. It can be incorporated into compositions, as illustrated in Yadegari's piece *Tear* (1999). Thus, self-similarity is stipulated as the essential principle of design, rather than emerging from other principles. This has implications from a cognitive musicological standpoint. In part addressing this, Yadegari situates his technique within a broader discussion of the nature and musical significance of self-similarity. Further work by Yadegari (1991, 2004) has emphasized the theme of self-similarity, both within a compositional and cultural context.

Pareyon (2011) is a comprehensive treatise on self-similarity in music. He analyzes the concept from the perspectives of various cognitive domains within a post-structuralist framework, discusses it's historic and musicological relevance, and introduces new models of self-similarity based on synechdoche and intersemiosis for future musicological undertakings.

2.2.2 Relation to Chaos

There is an important relationship between self-similarity and strange attractors, namely that the phase space of a strange attractor is self-similar. This is because stretching and folding leads to a Cantor set structure. Ultimately, this is what gives rise to SDIC. For further discussion, see Yadegari (1992) or Strogatz (2015).

2.2.3 Perception of Self-Similarity

How do we perceive self-similarity in music? Alexander Kobalyakov, in the article *Semantic Aspects of Self-similarity in Music* (1995), makes the important observation that 'It is necessary to include the perception (*mentality*) factor in a considered phenomenon of self-similarity.' A phenomenological investigation of various kinds of self-similarity would be of value to artists, and would likely intersect with music perception. A particularly important consideration is how we perceive phenomena that are 'self-similar in time', where the scale is in time or frequency instead of space. This would involve the phenomenological role of *memory*. For example, how does the role of memory change when the similarity is heard across short-term versus long-term time scales? Texture and structure are likely not perceived through the same mechanisms. Thus, a study of the perception of self-similarity might explore when memory is used or not, and how it is accessed in different ways. Also, the mechanisms involved in perceiving self-similarity may be related to those giving rise to the rhythm-pitch spectrum, since both depend on time scaling.

Chapter 3

Our Chaotic Music-Making System

The system is called *Mixing*. This refers to one of the definitions of chaos, and perhaps also reflects the turbulence of the music. Unfortunately, no measure theory was involved in the creation of this system.

Guidance from Systems Science

Systems science provides broad principles on how to describe systems. These principles are not formalized or standardized. For example, Mobus and Kalton (2015, chapter 12) discuss system decomposition using 'microscopes' to convert black boxes into white boxes. We will describe our system according to the following principles. (1) Our first principle is that we only describe at one scale at a time. Otherwise, we risk nonuniformity if we shift between scales over the course of a description, or risk incoherence if we simultaneously describe at two scales. (2) Another principle is that, for multiple descriptions each at a distinct scale, the order of scales should be consistent. This gives the effect of zooming in or out consistently. This can be achieved by a top-down or bottom-up approach. Top-down means breaking down systems into subsystems, and then subsystems into components. Bottom-up means building up subsystems from components, and then systems from subsystems. Note that our two principles were chosen *a priori*. There are many, arbitrary ways of completely describing a system (e.g. enumerating components and interactions as a list), and we chose these two out of convenience. (Note that we will not delve into the code that gives rise to this system.)

Specifically, we have chosen three main scales at which to describe our system, following a top-down ordering: (i) a number of high-level overviews of the system's structure and information flow, (ii) a low-level description of each of the two subsystems' structure and information flow, and (iii) a discussion of potential modifications at this low-level scale by zooming in to each component of the subsystem.

Objects and Functions

The main components are objects and functions which act on objects. *Objects* refer to any data values stored in memory. They include instances of object classes, as well as lists and values. *Functions* input and output objects. They include sending (where the output is the input), selecting (inputting ranges of values and outputting single values), shifting, scaling, partitioning, and so forth. It is through the action of functions that information flows in our system. Furthermore, when zooming in or out, descriptions of functions tend to change more than descriptions of objects. For example, a function can be expressed as a black box or a white box of objects and functions. Our three scales of description will thus illustrate increasingly detailed description of functions.

3.1 High-Level Overviews

The highest level description is a black box—a black music box. In a relatively accurate caricature, the user simply pushes a button and the music-making system plays a complete musical rendition. This rendition is a monophonic stream of sound, exhibiting phrasing at multiple time scales, including musical form. Each button-push yields a unique rendition.

We can zoom in slightly to arrive at a more meaningful description. The system can be decomposed into two parts: a *control subsystem* (coded in Python) and a *synthesis subsystem* (coded in SuperCollider). The two are connected via Open Sound Control, a network-based protocol for sending and receiving audio data. (1) The control subsystem inputs a matrix of forty values from the user. This recalls the questionnaire of sixty values in Koenig's PR-2. Technically, the user does not even need to input these values, since they are set by default. The control subsystem then outputs several streams of values, in parallel and asynchronously, for controlling synthesis parameters. (2) The synthesis subsystem inputs the output of the control subsystem. It then outputs sound, either as a file saved to disk or played back through speakers.

We can zoom in slightly further to arrive at a decomposition of the two subsystems into objects and functions. We have made a diagram, Fig. 3.1, which illustrates the two subsystems and their components at this scale. (1) The control subsystem begins with two objects: a set of time series generated from randomized initial variables and a user-inputted matrix of values. From the set, two are selected and combined to create a time series of pairs of values. The pair of values at a given instant in time bounds an interval within which we select a single value, using a selection procedure based on a stochastic process. Note that the selection procedure runs at a sampling rate independent of the update rate of the interval bounds. We perform several transformations on this time series, such as shifting, scaling, normalizing, and reordering of partitions. Then instantaneous intervals from this transformed time series are used to bound a stochastic process that selects the ultimate value used to control the instrument parameter. This is done for each instrument parameter. (2) The synthesis subsystem stores, for each instrument parameter, a pair of constant values which bound a stochastic process used to select instantaneous values. This is to autonomously produce continuous activity governed by a probability distribution, even when no new data is received from the control subsystem. These stored pairs are updated at rates varying in time (according to a separate time series for this purpose) and independently (asynchronously and in parallel) upon receiving the time series pairs sent from the control subsystem.

We here digress to discuss granular synthesis and tendency masks, as they will be central to our low-level description.

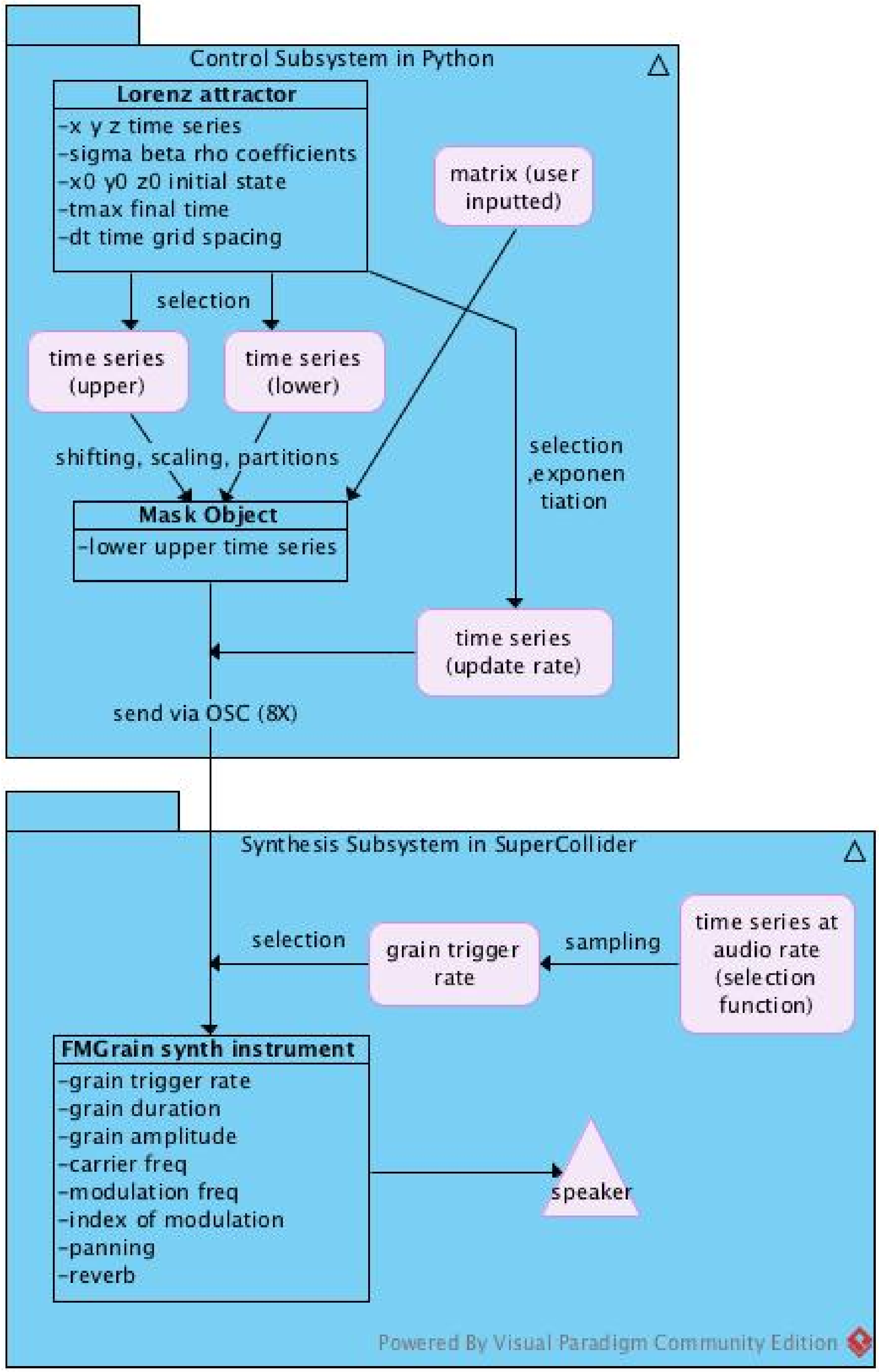

Figure 3.1. A diagram of the music-making system. There are two subsystems, for control and synthesis, which are linked by OSC. Boxes represent objects; their key properties are listed inside them. Purple boxes specifically represent lists or values. Arrows represent functions (indicated by arrow labels) relating two objects. Unlabeled arrows simply represent sending. The triangularly shaped speaker can be considered part of the synthesis subsystem, but this is arbitrary: it could be brought outside as well.

Granular Synthesis

A concise introduction to granular synthesis is Roads (1988). Further valuable references on granular synthesis include the comprehensive *Microsound* (Roads 2002), and Timothy Opie's website http://granularsynthesis.com.

In *Formalized Music* (1971), chapter 2, Xenakis put forth the hypothesis that 'All sound, even all continuous sonic variation, [can be] conceived as an assemblage of a large number of elementary grains adequately disposed in time'. Although framed as a hypothesis about the nature of sound, it is worth remarking that this notion of a grain would be quite constraining if it were adopted as the definition for synthesis. In historic and contemporary practice, granular synthesis draws less from a 'particle physics paradigm' and more from a 'time-frequency analysis paradigm'. In other words, it is less about the syntactic organization of elementary sound particles, and more about the *localization in time* of sound to create interplay between the time and frequency domains. Indeed, in mathematics we find the anticipation of granular synthesis in Gabor's work (1946) which laid the foundations for time-frequency analysis. Further discussion in this vein, of the relation between time-frequency analysis and granular synthesis, can be found in Clarke (1996) and Truax (2003).

Granular synthesis naturally gives rise to questions of control and organization for a composer. Xenakis originally devised a complex theory of *screens* to control the stochastic behavior of clouds of grains (Xenakis 1971). Roads (1978) approached the higher-level organization of grains into clouds using what he called *events*, and specifically discusses multiscale organization in Roads (2012). Truax (1988) developed the stochastic technique of *tendency masks*, which built upon concepts from Koenig and is further discussed in Truax (1990). Laske (1992) further developed the tendency mask approach. For example, in his composition *Furies and Voices* (1990) he used four levels of tendency masks (which he called the *virtual*, *movement*, *sectional*, and *variant* masks) as a top-down structure ultimately controlling five parameters (duration, modulation index, frequency, amplitude, and decay), varying them independently to produce what he called *parametric counterpoint* (Laske 1992).

Tendency Masks

A *mask* is a time-indexed sequence of a set of values; thus, it can be thought of as a time series with sets of values instead of single values. A mask must be paired with a *selection function* which, for each time index, chooses a single member of the set. An illustration of the selection process using a white noise generator is shown in Fig. 3.2, assuming that one value is chosen per mask pair (which is not the case in a real performance). Thus, a mask is a 'surface' of possible time series, and can be thought of as a kind of 'flow' (a space of possible trajectories). We will here present some basic construction approaches.

We implement masks as an 'upper' time series and 'lower' time series. All parameters involved are well-ordered, so at every time index the upper time series value is always greater than or equal to the lower time series value; we might call this the *mask condition*. To ensure this condition is satisfied when constructing masks (using the original Lorenz time series), we perform zero-shifting and normalization as necessary. It is the interval between the upper and lower values that serves as the actual set of values at that time index. In terms of software design, the mask-object class is a subclass of the data-object class. For the mask-object to be instantiated, the upper and lower time series have to pass a verification process that double-checks if the mask condition is satisfied.

The idea of masking was historically developed by Koenig and taken up most notably by Truax and Laske. It is an intuitive approach to stochastics and aleatorism and we can imagine much more can be done with them along the lines of 'formalized music'. For example, masks do not need to be time-dependent, but can be parametrized by other degrees of freedom. In their most general form they are simply (parameter-)varying constraints. This suggests geometric and dynamical systems approaches to 'formalized music', such as composition with manifolds or flows, where renditions are the possible curves or trajectories. We can imagine manifolds in timbre or feature spaces serving as generalized scores. The real challenge still remains not so much the mathematics but the audification of the mathematics, as we will discuss later.

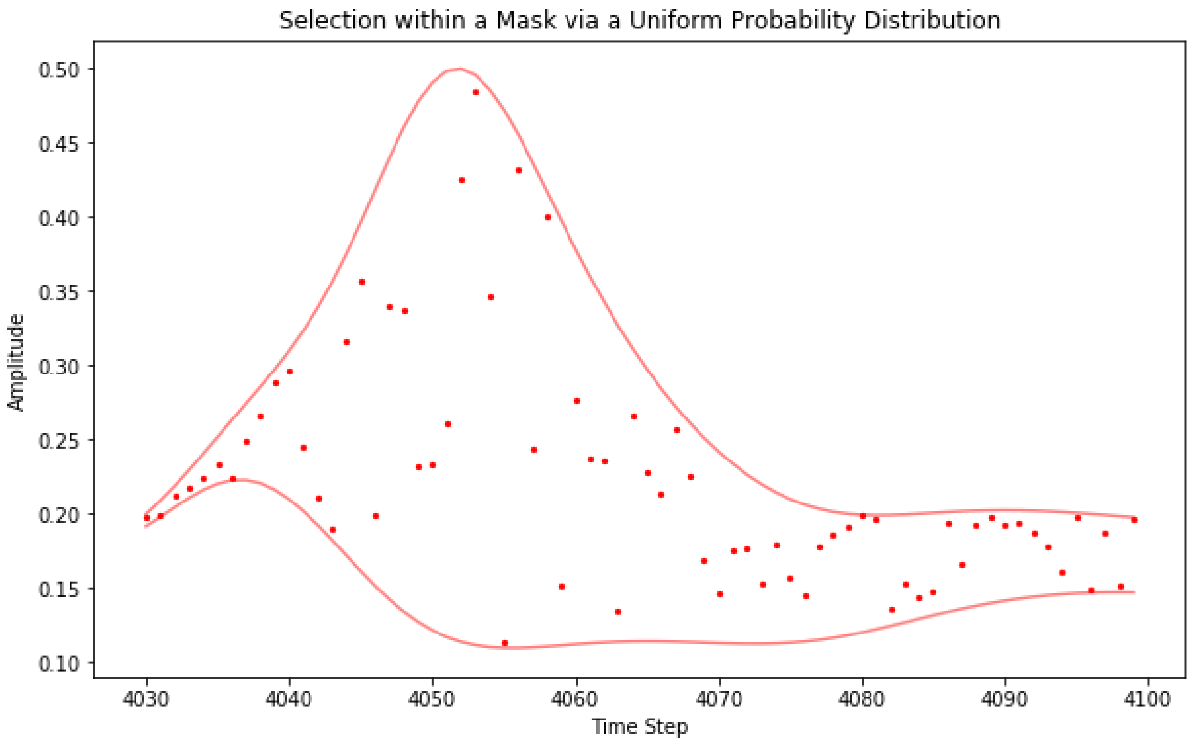

Figure 3.2. Mask selection.

3.2 Low-Level Description

We now proceed to discuss the system at the component level, in significantly more depth. Note that for our Python implementation, the Lorenz system of equations is numerically solved using SciPy's `odeint` (applying LSODA), on a time grid of linearly spaced points up to a maximum time. We did not conduct an exploration of the various regions of the Lorenz system's parameter space and phase spaces, although this would be ideal.

3.2.1 Control Subsystem

The control subsystem must have two pieces of information at the outset of any rendition: (1) a matrix of values specified by the user, and (2) a collection of Lorenz attractor time series automatically generated by a simple stochastic system. The two are combined to create a parameter mask.

(1) The matrix of values consists of the following five values provided for eight synthesis parameters: the number of segments n, the scaling range $[k_{\min}, k_{\max}]$, the parameter interval range $[x_{\min}, x_{\max}]$. Additionally there is a string argument specifying whether the segment order is monotonically increasing or decreasing, or randomized. The eight synthesis parameters are trigger rate, grain duration, amplitude, carrier frequency, modulation frequency, modulation index, reverberation, and 8-channel panning. The particular values used in recorded renditions settled on heuristically.

(2) The number of attractors and their typical variables (initial states, coefficients, time duration and samples) are all easy for the user to modify. The default is four attractors with randomized initial states and coefficients but fixed time duration and samples. The initial states are simply chosen in $[0, 1]$ according to a Gaussian probability distribution. The coefficients are generated by simple stochastic functions. These functions are designed to consistently yield attractors with the archetypal Lorenz shape. The resulting collection of time series is used as our initial control data, and their shape characterizes the probability distributions of grains. An example of the x-component time series of a Lorenz attractor is Fig. 3.3.

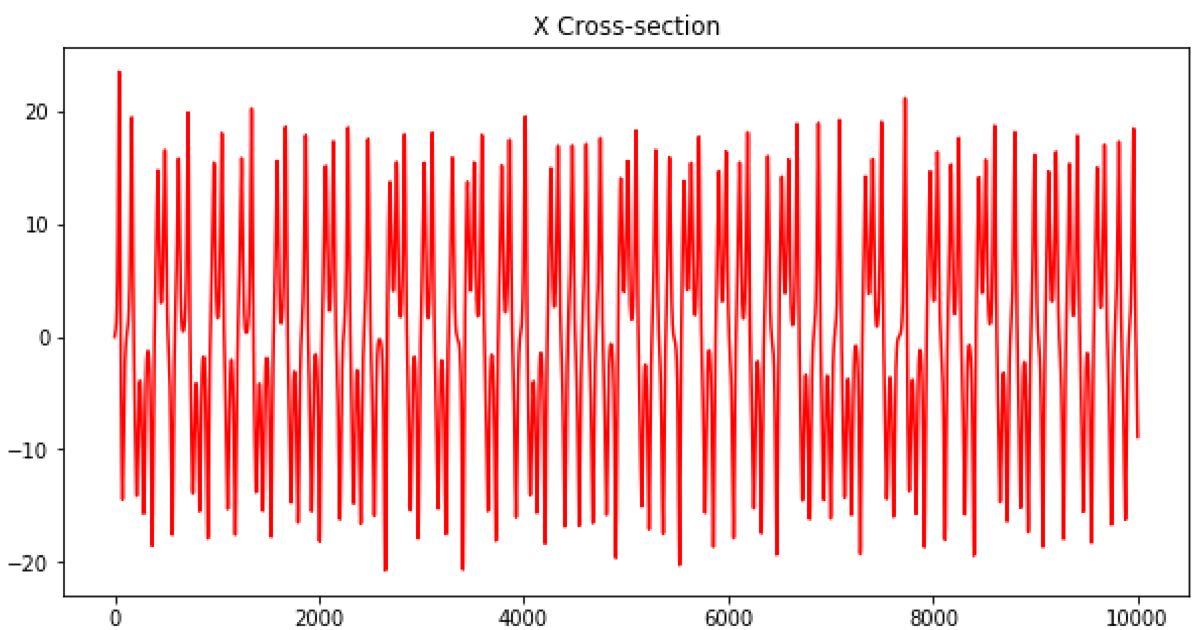

Figure 3.3. The x-component time series of a Lorenz attractor.

Then, for each parameter, an arbitrary pair of time series are selected and combined to make a mask: the upper and lower time series of the mask respectively are the sum and difference of the two zero-shifted time series. Zero-shifting (shifting to the horizontal axis) ensures that the sum is always greater than the difference. Then both upper and lower mask time series are zero-shifted together (we subtract the minimum of the lower mask from both) and normalized together to the parameter interval range $[x_{\min}, x_{\max}]$. This yields the first stage of our parameter mask, an example of which is shown in Fig. 3.4.

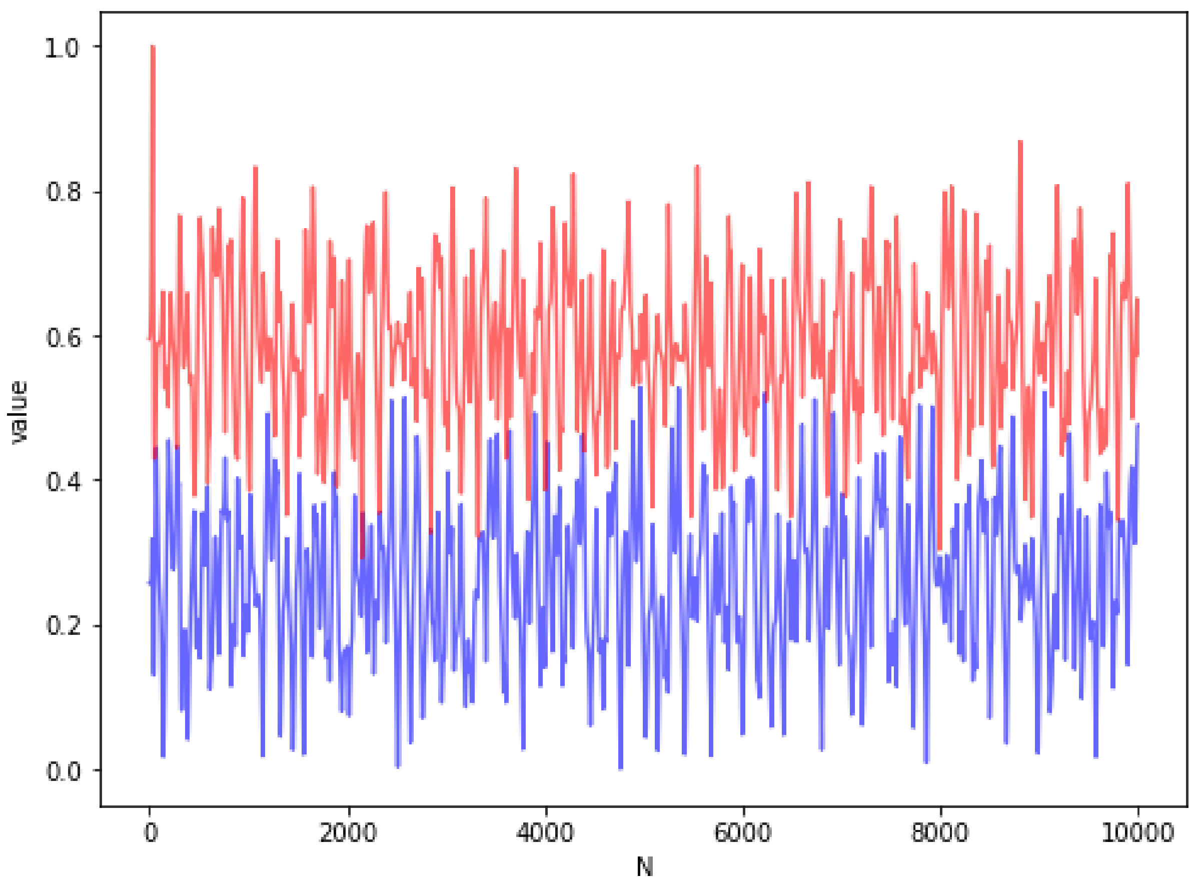

Figure 3.4. A first stage parameter mask, created from the x-component time series of two different Lorenz attractors.

The resulting mask is sometimes more and sometimes less interesting, depending on (a) the 'noisiness' or 'frequency-richness' of the initial time series, which might be proportionally measured by the noise spectral density, and (b) how different they are from each other, for which relative measures can be invented. An example of a mask with lower values in both (a) and (b) is shown in Fig. 3.5. A closer look at another first stage parameter mask is Fig. 3.6.

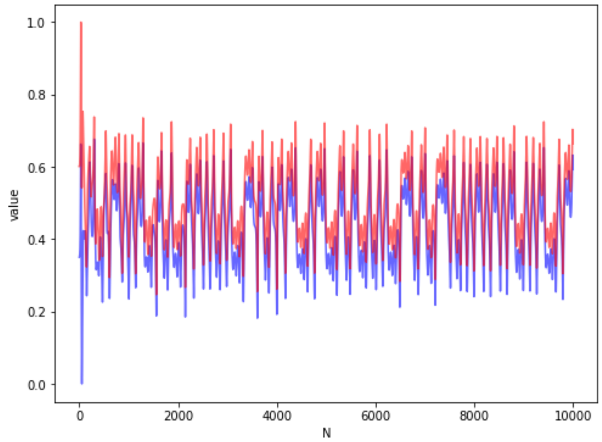

Figure 3.5. A first stage parameter mask with more regularities.

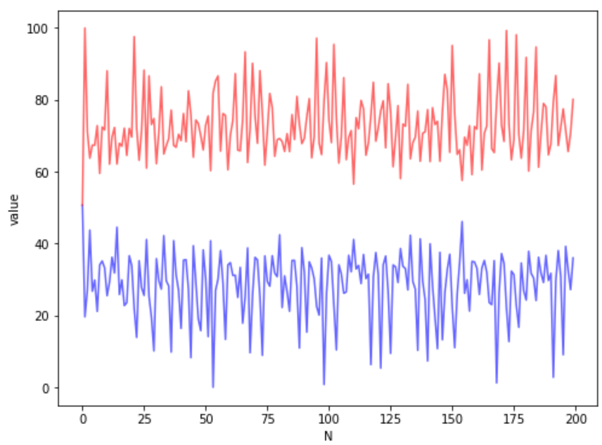

Figure 3.6. A closer look at yet another first stage parameter mask.

Even at this stage, we can begin a typology of masks based on attractor behavior and permutations of pairs of x, y, and z time series. We see the same clear ‘Lorenz-like’ character in all masks we generate: by performing only linear transformations such as shifting and linear scaling, the essential character of the time series is preserved. At every step of design there are many subjective choices: some were guided by musical intuition but in fact many are arbitrarily stipulated.

Next, the parameter mask is scaled for the first time. To generate our scaling factors, we create another mask by combining two more randomly selected time series in the same way as above, and use the upper and lower time series to normalize a third randomly selected time series. The result is a time series of scaling factors of equal length to our parameter mask. To scale, we first normalize our parameter mask to $[0, 1]$. Then we can either (i) take the product of the scaling factors and the parameter mask, (ii) take the mask time series to the power of one over the scaling factors, or (iii) take the scaling factors to the power of each of the mask time series. This choice is randomized by default. An example of a time series of scaling factors is shown in Fig. 3.7. An example of a second stage mask is shown in Fig. 3.8.

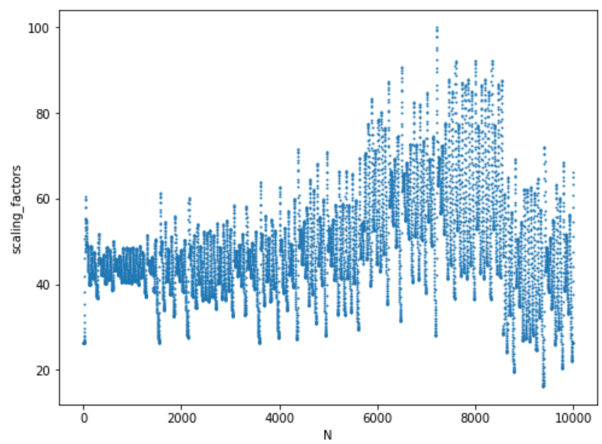

Figure 3.7. A time series of scaling factors.

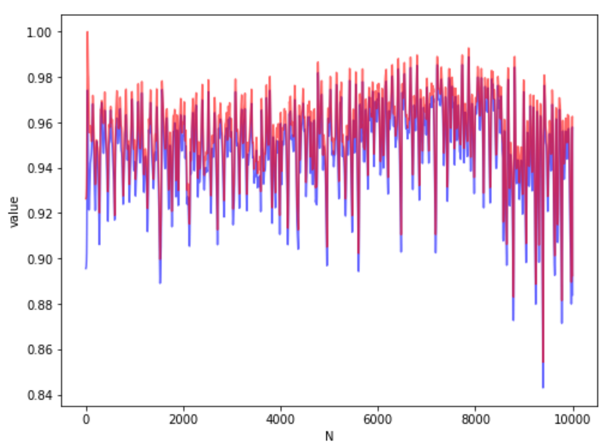

Figure 3.8. A second stage mask.

In the third and final stage, a second stage mask is partitioned into n (the user specified value) segments of equal length, each of which is each scaled by a single different factor. The scaling factors are an evenly spaced n-sampling of a logarithmic function, beginning with k_{min} and ending with k_{max}. The segments are then shuffled into a random order. The segmentation gives a sense of phrasing, and the logarithmic scaling is best suited for human ears. This is the final parameter mask. An example of third stage masks are shown in Fig.'s 3.9 and 3.10.

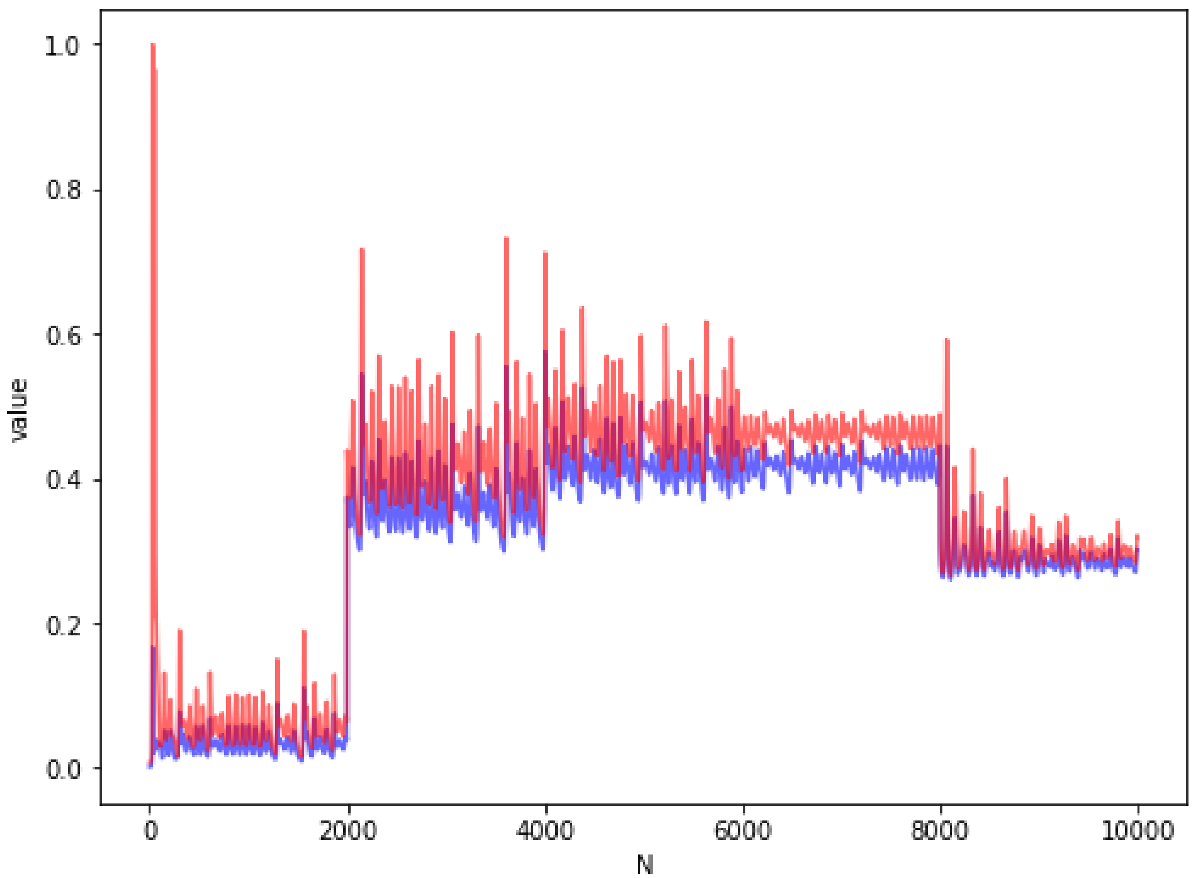

Figure 3.9. A third stage mask.

The mask values are sent to SuperCollider according to a variable update rate, which is controlled by a time series generated for that purpose. The update rate time series is calculated as 1.14 raised to the power of a randomly selected Lorenz time series, normalized to [1e-3, 1]. (This is for most parameters. The panning is [1e-4, 1e-2].) A graph of an update rate time series

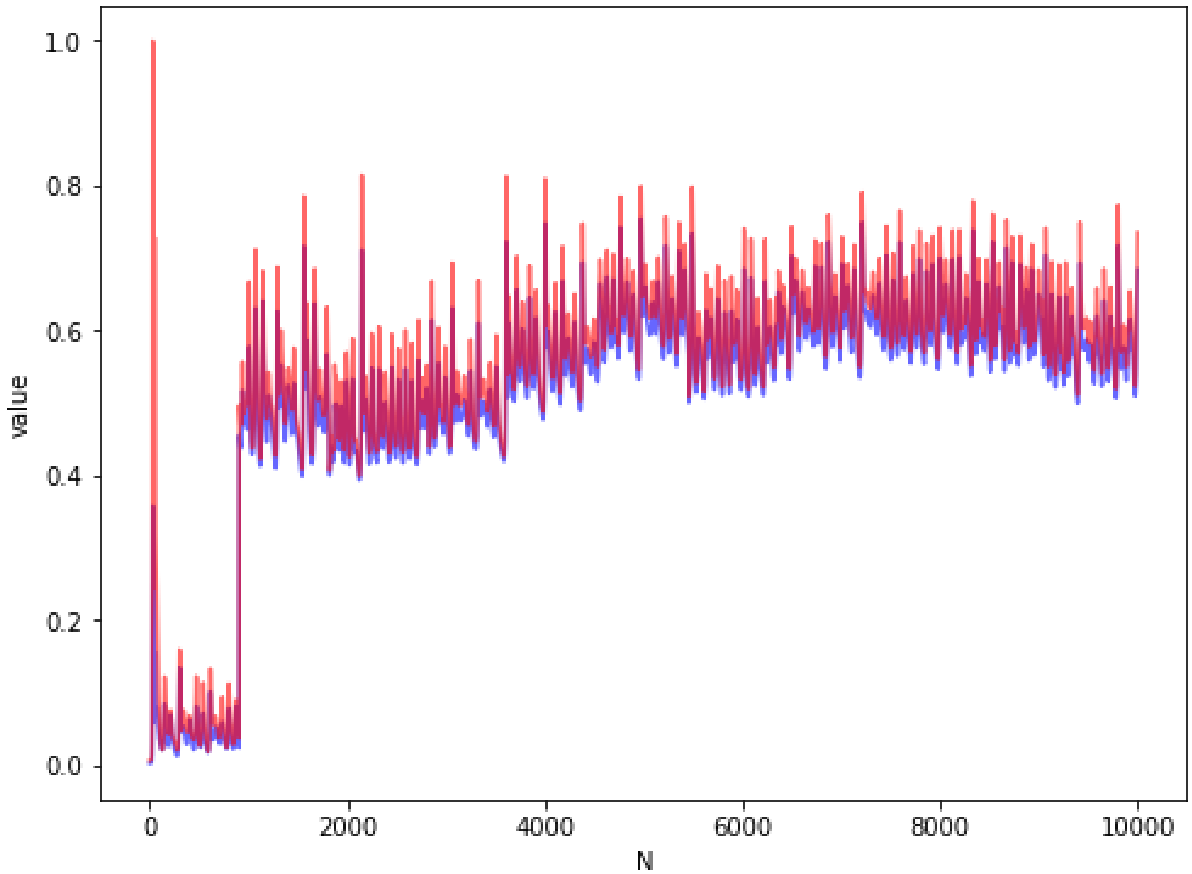

Figure 3.10. Another third stage mask.

is shown in Fig. 3.11.

To send parameter mask values to the synthesis subsystem, we (1) import the module `PythonOSC`, (2) specify the client ID for SuperCollider, and (3) send a pair of values of time index t, while (4) pausing according to the update rate time series value at index t before sending the next pair of values.

This concludes our low-level description of the control subsystem.

3.2.2 Synthesis Subsystem

The SuperCollider instrument is a `GrainFM` synth equipped with `OSCFunc` receivers for each parameter. The `GrainFM` synth is based on an object-oriented model of a grain, where a sine wave is frequency modulated by another sine wave and then windowed by a Hann envelope.

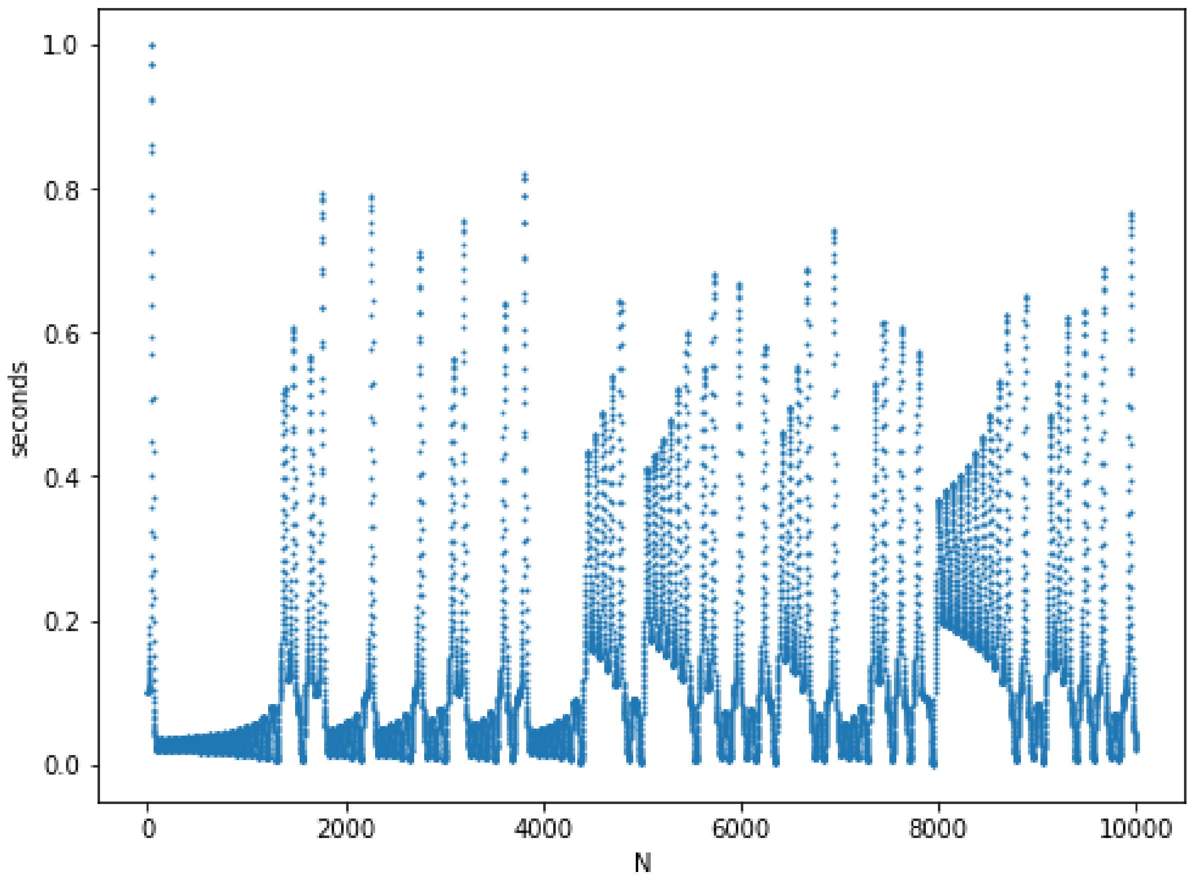

Figure 3.11. An update rate time series. This is the number 1.14 raised to the power of a randomly selected Lorenz time series.

It is a function inputting approximately eight parameters and outputting a sound signal. The eigh parameters are as follows. (1) *Trigger rate* starts a new grain based on an impulse generator whose rate is controlled by instantaneous Lorenz attractor values as solved in real-time at the audio rate. Parameters such as (2) *amplitude*, (3) *grain duration*, (4) *carrier frequency*, (5) *modulation frequency*, and (6) *modulation index* are self-explanatory. The default number of channels is 8 for an octaphonic experience (as performed at UCSD), so parameter (7) *panning* is over 8-channels. A simple FreeVerb reverb is applied with parameters of (8a) *mix* for dry/wet balance, and (8b) *room size*. The instantaneous values of these eight parameters are controlled by a Lorenz attractor instantiated from `ChaosGens` (using the default initial state and coefficients) that is sampled at the audio rate with linear interpolation, and normalized between a min and max. These upper and lower bounds are pair-wise updated via eight OSC receivers. The attractor continues to evolve at the audio rate regardless of whether new values are received, and its instantaneous value controls the parameter value. Thus, it is the updating of the min and max values according to a parameter mask that gives rise to that parameter's evolving bounds of variety over the course of the music.

The user cannot easily modify values in the SuperCollider code, but can change between using the real-time Lorenz attractor to white noise or other chaotic or noise generators that SuperCollider comes equipped with, e.g. in the `ChaosGen` class. A white noise selection function results in a much more balanced distribution, although can sound harsher because of discontinuity between values. A Lorenz attractor selection function on the other hand results in a much more continuous evolution of sound, yet can sometimes feel less varied. The selection function has a significant impact on the final sound, and a future version of this system would ideally incorporate algorithmic changes to selection functions, although this is challenging to do in SuperCollider.

The most accurate depiction of the selection process is shown in Fig. 3.12 and 3.13. These show a time-stretched mask (distorted and discretized by the update rate time series) bounding a Lorenz attractor time series (at the audio rate) which is enveloped by the mask and sampled at the grain trigger rate. Thus, not all control structure is always heard, but sometimes recedes due to 'undersampling'. What is heard is the change from narrow to wide bounds of selection, the height of this band, and the rates of update and selection. Ultimately, Fig.'s 3.12 and 3.13 are the most accurate depictions of what is happening instantaneously. We see the yellow curve at a shorter time scale being shaped by the blue curves at a longer time scale. This two-scale construction of time series is meant to give a sense of Lorenz-like structure locally (by the selection function) and globally (by the mask). This is done for each parameter, resulting in parametric counterpoint, which can be heard by listening to the different degrees of freedom in the sound varying simultaneously. Finally, in the global picture, if we were to zoom out farther from Fig. 3.13, the parameter masks would be segmented into dramatically different ranges

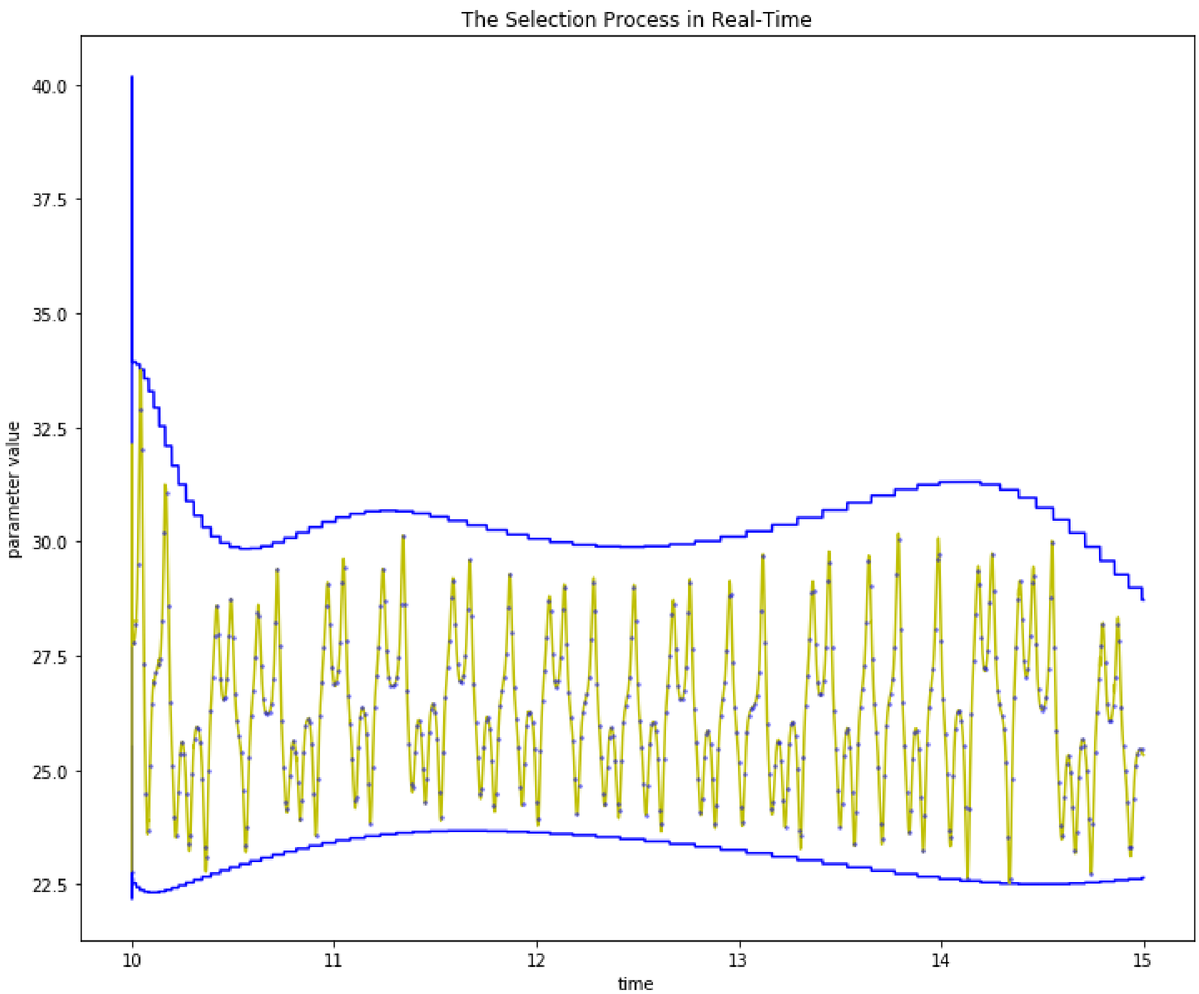

Figure 3.12. Mock-up of the selection process in real-time, close-up. Notice the clear Lorenz-like signal at one time scale (yellow curve) bounded by two Lorenz-like signals at a larger time scale (blue curves). The samples (blue dots) are distributed in time according to an update time series.

(recall Fig.'s 3.9 and 3.10).

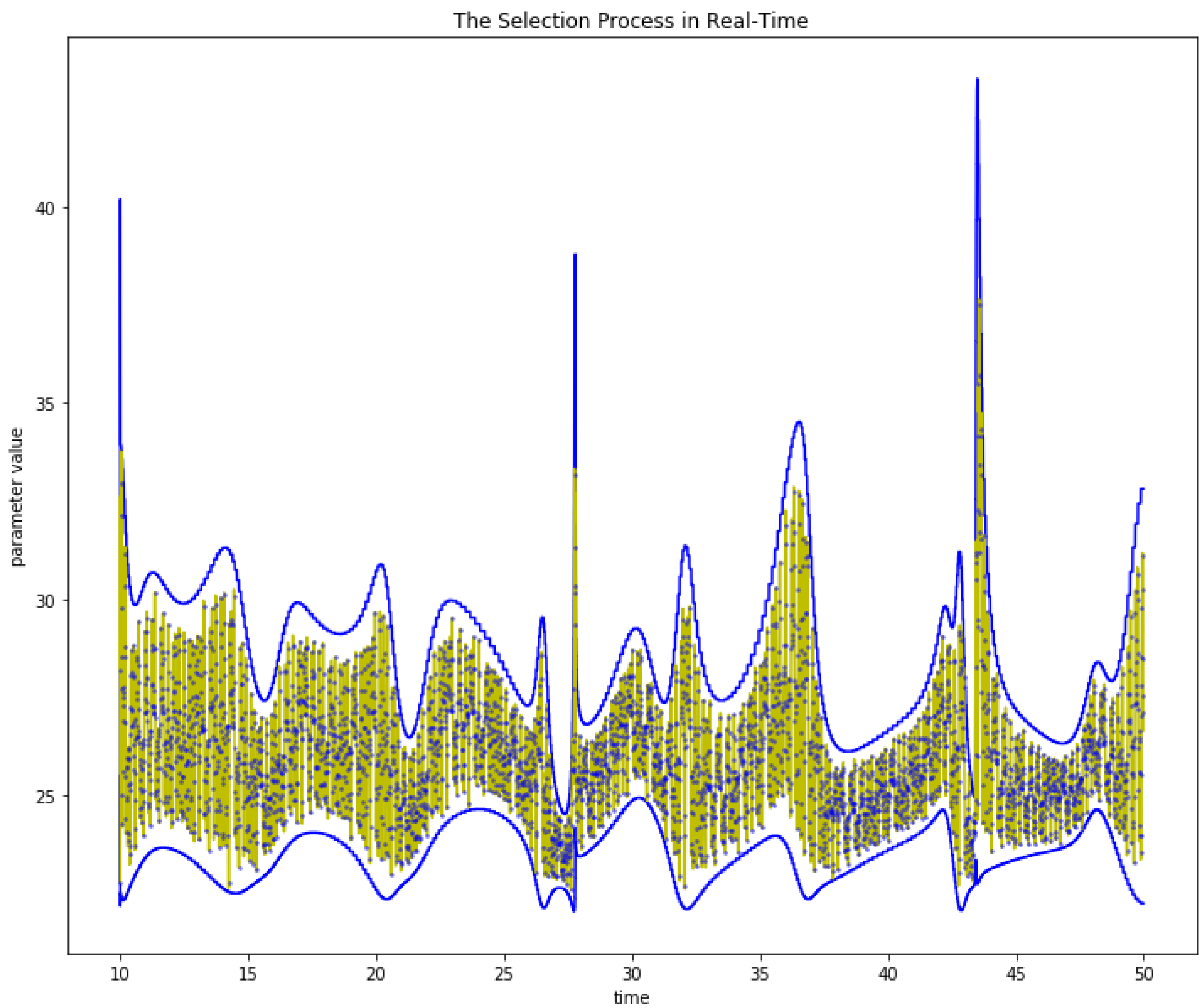

Figure 3.13. A mock-up of 40 seconds of the 'score' for a single parameter. This larger scale view illustrates the third stage parameter mask (blue curves) more clearly.

This concludes our low-level description of the synthesis subsystem.

3.3 User Interactivity

At one extreme, the system can be viewed as a completely closed (autonomous) system needing no user input or intervention. At another extreme, it can be viewed as a completely open system[1] admitting complete user involvement, depending on how interactively the user

[1] Note that *open system* should not be confused with *open form*.

chooses to work with the system. Because the system is implemented without a user front end, the user can engage with the implementation along a *spectrum of interactivity* with closed and open being at the spectrum extremities. For example, the user might interact with the system as a black music box, or might delve into the code itself and modify variables or options.

Usage as a Closed System

We will briefly describe how to use the program as an entirely closed system. This is unfortunately a description of the current implementation only, which is not yet streamlined. The user first (1) loads the instrument in SuperCollider[2] and then (2) opens a terminal window for each instrument parameter (each corresponding to an OSCdef) and runs a pre-specified line of code (i.e. runs the Python program with arguments). The author has narrowed down values for these arguments by trial and error over the course of several months by intuition. This means aesthetic judgements of 'musicality' as guided by principles such as the inverse U-curve, minimizing fatigue, and maximizing variety; as well as subjective personal preferences such as selecting for dramatic contrast. Upon executing the terminal code, a graph of the mask is shown and a printed text display tells the user (1) the current index in the time series in real-time, (2) the maximum index in the time series, (3) the lower and upper values of the mask interval, and (4) the time elapsed. (The display's update rate is the update rate time series.)

Usage as an Open System

We can discuss several DIY approaches to using the program, as an alternative to the out-of-the-box performance. These will be discussed in the next two subsections.

3.3.1 Modifying the Control Subsystem

The control subsystem can be modified in a number of ways by one willing to code in Python. The modifications discussed here are restricted to those which keep the relationships between modules invariant when holding other modules constant. We will not discuss possible

[2]This means that one boots the server, runs the block of code which creates and adds the SynthDef and OSCdefs, and runs the line which actually starts the synth node on the server.

additional modules. The modifications address (1) aspects of software design, (2) the technique of combining time series to construct masks, (3) the technique of partitioning and reordering mask segments, (4) fine-tuning of scaling principles based on audience constraint, (5) changing the dynamical system. These five modifications will be discussed in the following paragraphs.

Elaboration of Software Design

The central object class in this music-making system (as implemented in Python) is the *data-object*. It is simply an array of labeled time series (themselves NumPy arrays), with convenient built-in self-operations, such as indexing, iteration, and statistics. The reason for adding this overhead to NumPy arrays is to provide a general class with customizable properties, under which Lorenz attractors, masks, and so forth are subclasses. This allows us to define well-formedness conditions for subclasses, such as mask-objects being required to satisfy the mask condition (as we will discuss soon), and Lorenz attractor-objects including metadata on their coefficients, initial states, etc. Although only mask-object subclasses are implemented at the time of this writing, this object-oriented approach seems a fruitful and robust approach for realizing ‘formalized music’ or music-making systems. Defining and drawing from object classes that have well-formed properties (and that self-verify them) allows for building up complex systems of such objects. For example, making a subclass for dynamical systems and subsubclasses for particular dynamical systems would allow one to aleatorically choose from a set of dynamical systems while accounting for each of their idiosyncracies (e.g. metadata, which is crucial for full automation). Although all of this could be accomplished functionally, the object-oriented paradigm facilitates a certain way of thinking (e.g. systems engineering), and is a robust approach for scaling to larger systems.

Alternative Mask Construction Techniques

We discussed masks earlier. There are many simple ways of combining two given time series to create a well-defined mask-object, each yielding slightly different results. One of the simplest is to simply zero-shift one of the given time series and either add or subtract it from the other, which yields three possible well-formed pairs of upper and lower time series. We used the sum and difference here. Far more interesting results can be imagined, involving convolution or modulation, but that will be left for others. Note that the mask-object class is designed so that normalizing the mask as a whole normalizes the upper and lower time series together, thereby preserving the mask condition. We perform various scaling operations this way.

Because masks were sent in pairs of values from Python to SuperCollider, the selection function was implemented in SuperCollider; we used a sampled Lorenz attractor signal normalized between the mask values as our selection function, although a white noise signal (Gaussian distribution) would work just as well.

Segmentation for Musical Form

Masks are partitioned into segments, the number of which range from two to seventeen by default, or arbitrarily by user input. (We discussed musical rationale in the section on open form.) The process itself is simply to partition a mask-object into segments which are randomly reordered. However, we can significantly elaborate on this method in future versions. One simple improvement might be to proportion segments additively according to the golden ratio: then the first segment would be the longest, the second would be shorter by a factor of $1/\phi$, and so forth. This would give a sense of acceleration structurally. The rationale for this might be that human perception of time is likely not uniformly linear but somehow 'cumulating'. Considerations like this are discussed further in the following section on audification. Another improvement might be to add further detail to intermediate structure, i.e. phrasing within segments. Overall, segmentation is what gives a sense of large-scale shift or movement in this music, and is how we have chosen to address musical form.

Considerations for Audification

Audification may be defined as the mapping of information to the domain of human hearing. This implies having some knowledge of human hearing, whether through informal, personal trial and error or established science. For automation, ideally we might have a formal, simplified model of the human auditory system, which explicitly serves as our overarching constraint. Equal-loudness contours, the Bark scale, and signal processing models such as Sottek et al. (2005) are all examples of such formal models. However, they remain fragmentary, reflecting current, incomplete knowledge of human hearing. Even with a more complete knowledge, the proliferation of interpretations due to selective attention would remain a major concern, most likely lying beyond any model exclusively of the auditory system. For more on audification, sonfication, and auditory display, we recommend Hermann (2008), Roddy et al. (2014), and Barrass et al. (1999).

More challenging than the scientific considerations of audification remain its aesthetic considerations: different sensory encodings have different aesthetic value. How does one methodically or even intuitively approach the space of all mappings of a piece of information to the auditory domain? Typically, this is not how one approaches audification because it is impractical, but it raises interesting theoretical questions. In a broader context, we can generally define a *sensory mapping* as a mapping of information to the human sensory domain, including visual, aural, tactile, olfactorial, gustatorial, proprioceptive, and other subdomains. It would encompass mappings from nonsensory domains to sensory domains (e.g. visualizing data, audifying data) as well as between sensory domains (e.g. visualizing sounds, audifying images). One might approach sensory mappings as sequences of encodings or transformations, such as from data to instrument parameters to acoustic parameters to perceptual parameters. Note that although musical space and spatialization can often be simply modeled as filters, they are also aesthetically a kind of remapping of musical elements; if the human auditory system could be modeled as a coherent whole, one could simulate how such mappings are perceived, which would be of benefit to automated composition.

In the earliest stages of design, the question of audification for human ears and minds was one of the first that emerged. A Lorenz trajectory, if appropriately normalized, can be used to control the frequency of a sinusoid (a kind of frequency modulation). At a slow time scale of modulation, the resulting sound is straightforward enough; faster modulation (e.g. a Lorenz time series updated at the audio sample rate) results in an interesting noise texture. The stochasticity of masks was here introduced as a way to flesh out this chaotic behavior, which otherwise would remain one-dimensional. Further developments led to questions of how to scale masks ideally for human hearing and to maximize interest. For example, just as the eye has a small, central area of particularly sharp vision due to the fovea, so can we recall the equal-loudness curves and think of the most sensitive frequency ranges as a kind of 'acoustic fovea'[3]. In fact, Xenakis explicitly began with the Fletcher-Munson curves when composing his *Analogique A et B* (1958-59) as described in chapter 2-3 of Xenakis (1971). When scaling our frequency masks, we took human hearing into account to maximize interesting behavior in the acoustic fovea. Similar considerations could be applied to the other parameters (such as grain duration, amplitude, etc.) based on other knowledge of human hearing (attention to sudden attack noises, etc.).

Alternative Dynamical Systems

There are countless other chaotic dynamical systems that can serve in place of the Lorenz attractor, for generalizations of this system. Thus, some space of such dynamical systems might be defined from which the germinative dynamical system is selected. Alternatively, perhaps throughout the course of the piece one transitions between different dynamical systems. Doing so in a continuous, smooth manner poses an interesting challenge.

3.3.2 Modifying the Synthesis Subsystem

The synthesis instrument is almost arbitrarily changeable. This is because the control subsystem is easily adapted to other synthesis instruments, with more or fewer parameters. Adapting would involve identifying the absolute parameter ranges one wants variety within, and gauging the scaling factors heuristically by trial and error to yield the most 'interesting' final behavior.

[3] A term which technically comes from the field of animal echolocation, e.g. the study of bat and dolphin sonar.

Some synthesis instruments are more or less relevant to the character of the piece. For example, we can invent an even better instrument that our current one: a granular synthesis technique where Lorenz envelopes can be constructed and modulated with each other. How might this be done? Given a Lorenz time series, we can randomly select a closed interval of the time series of any length, and zero-shift and scale accordingly to create a window of a Lorenz shape. How much the attack and decay are smoothed is (inversely) proportional to the roughness of the texture. They can be smoothed by further applying a very brief 3-piece window with a linear ramp up to 1, plateau at 1, and a linear ramp down to 0 (e.g. a 'fade-in' and 'fade-out'). This can be done multiple times to create a set of grain envelopes. To increase the noisiness and variety, as well as create a sense of continuous transitions, we add two envelopes with coefficients (which sum to 1) as a linear combination to create mixtures of envelopes. The granular parameters to be controlled might then be trigger rate, grain envelope number, grain duration, and grain amplitude, as well as a second grain envelope and coefficients for a linear combination. Then the phrasing of the composition could be further articulated by grain envelope. Examples of Lorenz-like grain envelopes are shown in Fig. 3.14 and 3.15.

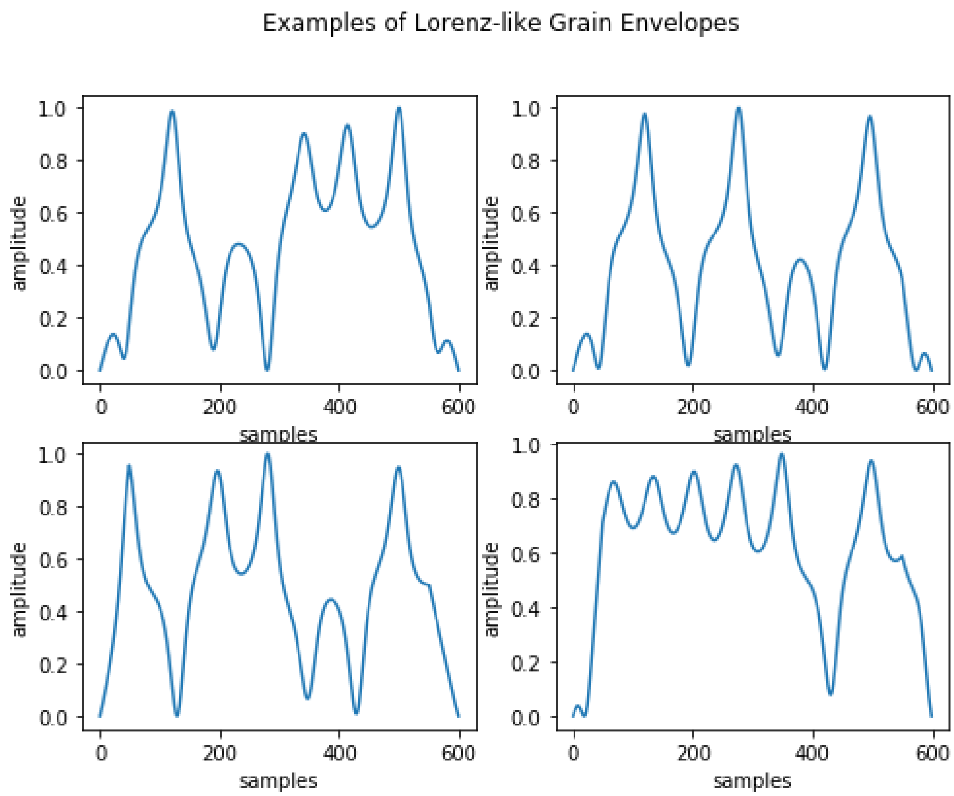

Figure 3.14. Examples of Lorenz-like grain envelopes that have already been windowed by a ramped step function.

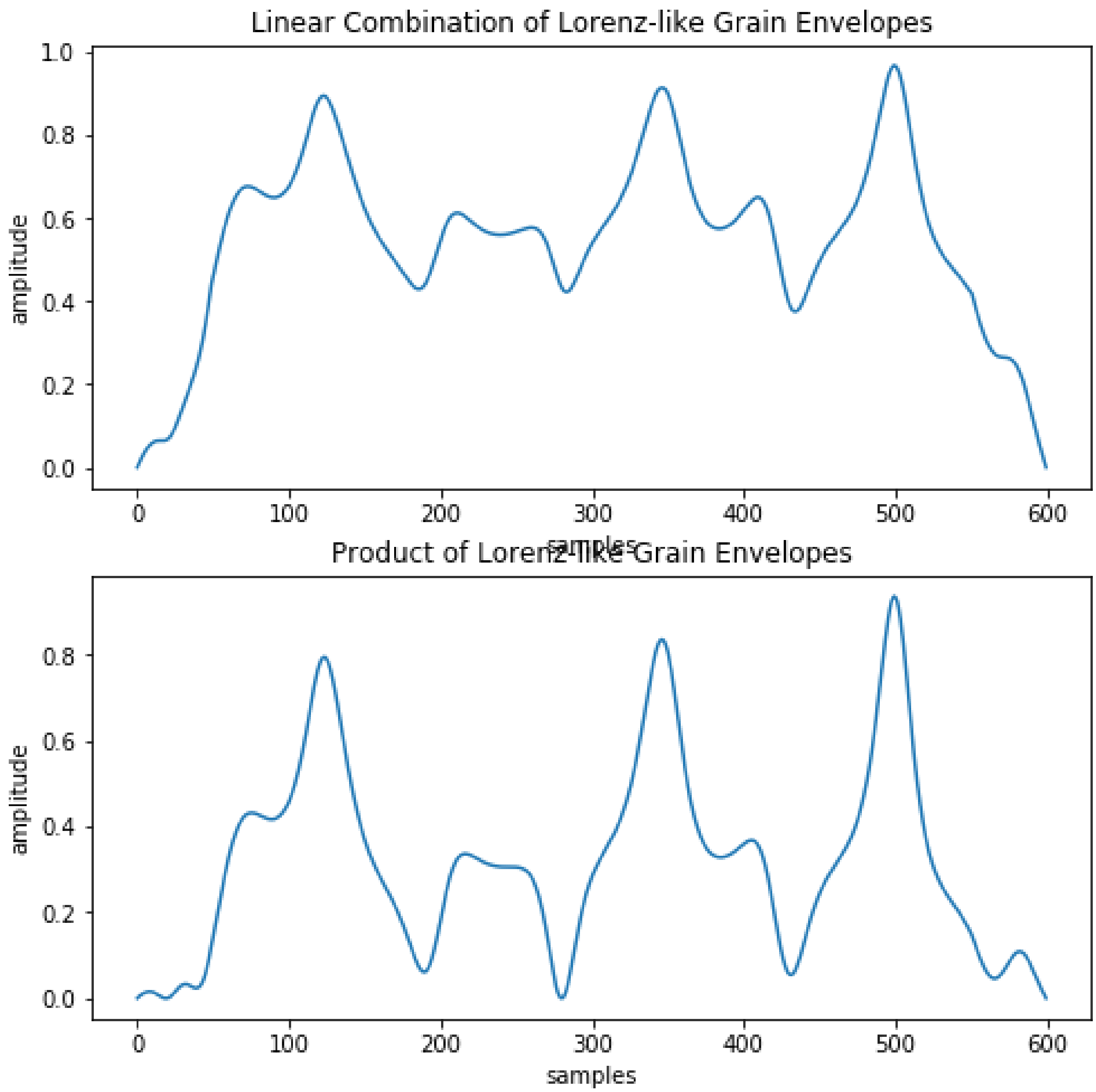

Figure 3.15. Above is an example of a linear combination of envelopes, calculated as (0.5*env1 + 0.5*env2)*step. Below is an example of a product of envelopes, calculated as env1*env2*step.

Chapter 4

Discussion

In this chapter, we will discuss how our system illustrates the first two chapters.

4.1 Composition Theoretic Analysis

How might a composition theorist analyze our music-making system? Because we are in the coincidental position of being both designer and analyzer, we can introspect on cognitive events giving rise to the design process. However, there is a world of cognition that arises when designing a system like this. We will only consider those cognitive events that objectively manifest in the system and can be observed by others. This especially means identifying stipulations and their rationale. We will set aside subjective considerations like artistic vision, poetic connotation, etc.

Our approach was top-down, stipulatory, and script-based. However, this belies many preceding months of exploration. There was a long gestation period of testing small ideas and evaluating their success, which is in fact bottom-up. (1) The first top-down stipulation was the germinative idea: the Lorenz attractor as a monotheme. This served as the primary constraint throughout the design process. Our selection of this chaotic dynamical system was because it is a paradigm of chaos. Thus, our selection took into account how well-studied the Lorenz attractor is, its easy solution, and its distinct character. This first stipulation entailed and underpinned many subsequent stipulations. It entailed self-similarity, multiscale design, how chaos is a major attribute of this music, and to an extent even granular synthesis as a multiscale time-domain synthesis technique. Self-similarity and multiscale design further led to masks taking on the shape of Lorenz trajectories. (2) Our second stipulation was the desire to optimize for a human experience. For example, this implied deliberately mapping to human hearing. In terms of constraints, this second stipulation can be seen as the adoption of an 'audience constraint'. This entailed how to best expose the Lorenz attractor's chaotic behavior, further leading to its 'fleshing out' with the stochastics of masking, as well as scaling to the 'acoustic fovea'. (3) Our third stipulation was the duration of about ten minutes. One can think of this as a 'sub-stipulation' arising from the audience constraint: presenting multiple scales too quickly would be unfeasible given the audience's nature, such as maximum limits on information processing. Thus, this third stipulation can be seen as underpinning segmentation, and to an extent our treatment of self-similarity and open form. (4) A fourth stipulation was full autonomy. This is what most directly led to open form, and the particularities of our system's stochastics and aleatorism. It is also what led to involving Laske's composition theory and a systems approach to software design.

These several stipulations are not entirely separate but can be related, as we saw with the second and third stipulations. Although this is a brief sketch, we can see how only a handful of elementary stipulations might logically shape the entire development of a system.

4.2 As an Illustration of Open Form

In our music-making system, the number and ordering of segments of parameter masks can optionally be specified by the user. However, the default is a randomized ordering and for the number of (equal-length) segments for each parameter to be mutually co-prime so that overlaps do not feel too sudden. For example, trigger rate has three segments, grain duration has five, amplitude has two, carrier frequency has thirteen, and so forth for all eight parameters. The particular number for each parameter was chosen based on the observation that the number of segments was proportional to how quickly we move on to a new segment, i.e. the variety of phrasing for that parameter. Thus, carrier frequency has thirteen segments so that we move on to a new segment at a faster rate for that parameter (approximately each forty-six seconds for a ten minute rendition). The time scale implied by the number of segments is significant: two to five segments may be heard more as sectioning, whereas thirteen or more segments may be heard more as phrasing (hence our use of the general term 'segment'). Furthermore, the audibility of segments depends on the randomization process: if two consecutive segments are similarly scaled, the result is a sense of a twice-as-long segment, whereas if two consecutive segments are drastically different, the result is a qualitative sense of contrast for that parameter. Note also that change of segment for parameters such as amplitude or carrier frequency may be heard more prominently than for parameters such as reverb or modulation index. Thus, the actually perceived phrasal structure should be analyzed post-rendition. In short, open form here applies to our randomized ordering of mask segments for each parameter. The result of the culminating parametric counterpoint is that the listener passes through macrocosms, which feel like phrases or sections.

It is valuable to distinguish between openness of form due to causes that are top-down (e.g. directly stipulated at the macroscale by rule) or bottom-up (e.g. emergent or self-organizing from microscale rules). These are not total opposites due to the inextricability of 'form' and 'content'. For example, the designer may stipulate a rule at the largest scale for organizing sections according to a Markov process. Alternatively, the designer may stipulate a grammar at the microscale for organizing small chunks of material so that they relate to one another syntactically and thereby produce emergent large scale structures. In our case, the act of segmentation was stipulated in the top-down design of this system. This was specifically to address macrostructural design, and arose in conjunction with the decision for this to be a lengthier piece (roughly ten minutes, or for an installation version, indefinite). Without the act of segmentation, this would arguably not be a piece of music at all, but merely a mathematical curiosity. A more sophisticated version of this system would involve multiscale design of both macrostructure and intermediate structure. These truly free (i.e. arbitrary), but musically intuited or justified stipulations often are what give algorithmic compositions their distinct character and style.

4.3 Reflections on Granular Synthesis

In this music-making system, we continue along the figurative, time-domain paradigm of granular synthesis. Constraints of elementariness and brevity have been relaxed, allowing for grains that can last for many seconds (> 5s), which is facilitated by SuperCollider's built-in functions. Thus, we are not using the granular synthesis of either Xenakis or Roads, which are microsound conceptions of the grain. Also, we only use a single stream of consecutively spaced (although sometimes overlapping) grains, as a kind of solo instrument monophony, again facilitated by SuperCollider. This is distinct from the concept of a cloud as a statistical aggregate. It is worth exploring this aspect from the perspective of music perception, such as the concept of good continuation (Deutsch 1999). Grains overlap at times, creating a continuous sonic transformation, and at other times separate, creating a staccato texture. As mentioned before, in the proposed future version of the synthesis subsystem, each grain would be 'Lorenz-shaped'; and so this fluctuation between overlap and separation would shift the listener's focus between the Lorenz-like collective behavior of clouds and the Lorenz-like shape of individual grains.

We have continued the now longstanding approach of using masks in conjunction with granular synthesis. An important reason is that masks are suggestive of grains—the shape of masks at the macroscale can be made similar to the shape of grains at the microscale, which is exactly a kind of self-similarity.[1] Laske brought this out in *Furies and Voices*: phrases begin to be conceived of as 'macroscale grains', which is why his three 3-minute 'movements', separated by silence, are in fact embedded within a single uninterrupted piece. We must remember that this implies a distinct approach from the original conception of the grain strictly as microsound, however. Ultimately, we will have actualized this idea upon specifically making the masks the same shape as the grains.

[1]This is suggested by Roads' *events* as well.

Thus, by relaxing the constraint of grain briefness and implementing masks, we find the theme of the Lorenz attractor in both the texture and the form. In this way, granular synthesis with masks reinforces the theme of self-similarity and is integral to this music-making system.

4.4 The Role of Chaos

Each rendition generated by this music-making system is a complete, unique musical experience. This is due both to stochastic and chaotic elements. We find pseudo-random processes at various stages of mask construction (e.g. random selection from a set, randomized coefficients, etc.) and also are faced with chaotic behavior of the sonic material as well as the phrase structure. The chaotic behavior is what ultimately stands out most prominently in that it characterizes the actual transformations we experience, whereas stochasticity merely prepares the way for chaotic material. For example, each rendition is unique despite keeping user inputted initial values constant: this is because we use stochastic processes to slightly perturb the initial state and coefficients of our initial Lorenz attractors, which results in distinct Lorenz trajectories in the long term. This would only be apprehensible upon listening to multiple renditions, however.

To better see the importance of chaos in this work, we might contrast it with Laske's *Furies and Voices* (1990). A critical observation, which can be made by either looking at the mask figures he provides in his 1992 paper or by looking at a spectrogram of the piece, is that the parametric counterpoint is mostly linear. This linear parametric counterpoint is apparent in other compositions of the era, such as Truax's *Riverrun* (1986). For example, we hear frequency and amplitude rising and falling according to linear envelopes. In contrast, there is a strong nonlinearity to the mask shapes used in this work. The result is a parametric counterpoint which feels more unpredictable and diverse. It is an interesting question whether, in fact, the accumulation of information over time via hearing (whether perceptually or due to memory) is linear or not. In other words, we might feel an exponential ramp is more interesting than a linear ramp. In any case, nonlinear or chaotic parametric counterpoint is much less predictable that linear, and begins to approximate aliveness.

4.5 As an Illustration of Self-Similarity

We have already discussed the self-similarity of strange attractors. What is the role of self-similarity specifically in this music-making system and the music it generates? In our system, the Lorenz attractor is used as control data at multiple time scales, namely the tendency mask (long-term phrasing) and selection function (short-term activity). The Lorenz attractor also serves as control data for various degrees of freedom, such as different synthesis parameters and update rates; and in future implementations with Lorenz-like grain envelopes, we will see it in the sound itself.

For this kind of situation, Pareyon (2011) provides many analytic tools. We will specifically draw on his notions of (1) *functional similarity*, meaning 'the order established between two systems of comparable relationships' and (2) *statistical similarity*, meaning 'the considerable degree of comparison between values which are not identical or equivalent' (Pareyon 2011). Based on these concepts, we suggest two possible ways of talking about self-similarity in our situation: (1) we can consider the music-making system and formulate a language for talking about recursion in concept design, e.g. devise a functional or 'conceptualistic' similarity measure of the recursive use of material in systems, and (2) we can consider the resultant musical sound signal and quantify its statistical properties, e.g. devise a measure of statistical self-similarity for such signals. We will discuss both points.

Functional Self-Similarity

The tendency masks are inherently as chaotic as the time series used to make them: this is because we have only performed well-behaved linear shifting and scaling operations to create the mask, which do not affect the fundamentally chaotic behavior here. When these masks are applied as an envelope to another Lorenz time series, what happens? The selection function Lorenz time series (at the audio rate) becomes globally shifted and scaled throughout the course of the mask, taking on a global Lorenz-like behavior. Again, because these distorting operations are linear, the selection function Lorenz time series also remains locally Lorenz-like after applying the mask. Furthermore, the update rate of the mask as it is sent from Python to SuperCollider as well as the grain trigger rate which samples the selection function Lorenz time series are both Lorenz-like time series. This means that the flow of time and amount of activity both feel Lorenz-like.[2]

Statistical Self-Similarity

What does functional self-similarity ultimately entail in the music itself? The resulting sound has interesting properties under time and frequency transformations. (a) For example, you can verify that the essential character of this music is invariant under time stretching and contraction (e.g. in Audacity), from about 1/3X to 3X speed. In other words, the transformed sound seems distinctly as though it comes from the same piece and is indistinguishable from some other potential section at 1X speed. It would ideally be infinitely time stretchable and contractible, but it is not due to our finitely many scales of design and the finiteness of digital audio. One explanation of this invariance is that the music is always characterized by periods of greater and lesser activity, but all activity is of the same character. Thus, periods of greater activity are simply very rapid Lorenz-like behavior, and vice versa. Time stretching causes a transformation between texture and structure, and when texture resembles structure we have invariance under time-stretching. (b) Likewise, shifting frequency (and the other parameters if this were more easily possible) within certain ranges yields a similar result. Again, this is because such shifting already happens throughout the piece in a regular fashion, and so a single shifting operation becomes inconsequential.

Ultimately, this statistical self-similarity gives the impression of many layers of structure, a subconscious sense of coherence between surface and deep structure, and unity in variety.

[2]Note that the use of FM granular synthesis in the current implementation is indeed a ‘breaking of symmetry’, unfortunately.

Composition Theoretic Analysis of Self-Similarity

Although Yadegari's focus in the 1992 work is on synthesis of sounds as a supplement to composition rather than design of systems for autonomously generating complete compositions, there are important analogies that can be made between the two. In both works, the explicit stipulation of self-similarity as a design principle entails parallel logical considerations. We can generalize a logical process of 'self-similar design' from both works, as an example of rule-based composition: (1) define the *unit* of the 'self' (what is invariant? a feature, shape, etc.?), (2) define the *scale* (informationally, conceptually, size, nesting, etc.?), (3) define a *similarity measure* for comparing units (where exact is considered a special case of statistical). In Yadegari's synthesis paradigm, (1) the unit is a user-provided sound called a *cell*, (2) scale is considered as both the time and frequency axes, i.e. in a spectrogram, the shape of cells are geometrically similar after scaling along the time-frequency axes, and (3) we in fact have exact self-similarity (fusing functional and statistical together). We must remember that these decisions are creative and, even if logical, are not preordained; the particularity of these decisions are what give rise to the identity of this sound. In our music-making system, (1) the unit is the Lorenz attractor, (2) scale is also considered as the time and frequency axes, and (3) we have both functional and statistical similarity but they remain distinct. Thus, by considering the logic of design from a composition theoretic perspective, we can articulate analogies between two different works. Note that the empirical study of composition methods is distinctively composition theoretic.

We can also speculate as to analogies between the cognitive processes: were there similar motivations in both situations, drawing both to self-similarity? To explicitly adopt self-similarity as a design principle suggests that it has some value to the artist, whether practically for musical or sonic coherence, or aesthetically for stylistic and philosophical reasons. Thus, when considering rule-based composition we can infer there is some rationale behind choice of rule—here being the choice to make self-similarity the unifying design principle in both works.

The significance of self-reference has been emphasized by many. We find such veins of thought in Yadegari (1992) who says that 'self-similarity should be thought of as a portrait of a self-referential entity' and Pareyon (2011) who discusses self-reference in chapter 3.6. It is notable that Laske entitled his 1980 paper *On Composition Theory as a Theory of Self-Reference*, proposing that 'self-reference is the crucial topic of a theory of composition'. Laske points out the paradox in attempting to 'state self-reference in allo-referential terms', where Laske defines an allo-referential system as one which 'refers to its observer or maker, but not, as a living system does, to itself'. We suggest that this has deep connections to Godel's incompleteness theorems.

Bibliography

References for Chapter 1

Andersen, D. R. (2020). The Limits of Indeterminacy: The Performance and Analysis of Selected Indeterminate Compositions by John Cage and Earle Brown.

Balaban, M., Ebcioğlu, K., & Laske, O. (Eds.). (1992). Understanding music with AI: perspectives on music cognition. MIT Press.

Collins, N. (2009). Musical form and algorithmic composition. Contemporary Music Review, 28(1), 103-114.

De Bièvre, G. (2012). Open, mobile and indeterminate forms (Doctoral dissertation, Brunel University School of Arts PhD Theses).

Dewey, J. (1934). Art as experience.

Dunn, T. L., & Risko, E. F. (2016). Toward a metacognitive account of cognitive offloading. Cognitive Science, 40(5), 1080-1127.

Goldman, A. H. (2013). The broad view of aesthetic experience. The Journal of Aesthetics and Art Criticism, 71(4), 323-333.

Kugel, P. (1990). Myhill's thesis: There's more than computing in musical thinking. Computer Music Journal, 14(3), 12-25.

Laske, O. (1980). On Composition Theory as a Theory of Self-Reference. Allos, 419-431.

Laske, O. (1981). Composition theory in Koenig's project one and project two. Computer Music Journal, 5(4), 54-65.

Laske, O. E. (1988). Introduction to cognitive musicology. Computer Music Journal, 12(1), 43-57.

Laske, O. (1989). Composition theory: An enrichment of music theory. Journal of New Music Research, 18(1-2), 45-59.

Laske, O. (1990). The computer as the artist's alter ego. Leonardo, 23(1), 53-66.

Laske, Otto E. (1992). "Furies and Voices: Composition-Theoretical Observations." Otto Laske: Navigating New Musical Horizons, Ed. Jerry Tabor. Greenwood Press, 1999.

Laske, O. E., & Tabor, J. (Eds.). (1999). Otto Laske: Navigating new musical horizons (No. 53). Praeger.

Lewin, D. (1986). Music theory, phenomenology, and modes of perception. Music Perception, 3(4), 327-392.

Lochhead, J. (2015). Reconceiving structure in contemporary music: New tools in music theory and analysis (Vol. 2). Routledge.

Nof, S. Y. (Ed.). (2009). Springer handbook of automation. Springer Science & Business Media.

Pritchett, J. (1993). Introduction to the music of John Cage.

Risko, E. F., & Gilbert, S. J. (2016). Cognitive offloading. Trends in cognitive sciences, 20(9), 676-688.

Roads, C. (1978). Interview with Gottfried Michael Koenig. Computer Music Journal, 2(3), 11-15.

Schoener, T. (2009). I.1 Ecological Niche. In S. Levin, S. Carpenter, H. Godfray, A. Kinzig, M. Loreau, J. Losos, B. Walker & D. Wilcove (Ed.), The Princeton Guide to Ecology (pp. 3-13). Princeton: Princeton University Press. https://doi.org/10.1515/9781400833023.3

Tegmark, M. The Mathematical Universe. Found Phys 38, 101–150 (2008). https://doi.org/10.1007/s10701-007-9186-9

Welsh, J. P. (1994). Open form and Earle Brown's modules I and II (1967). Perspectives of new music, 254-290.

References for Chapter 2

Abdallah, S., & Plumbley, M. (2009). Information dynamics: patterns of expectation and surprise in the perception of music. Connection Science, 21(2-3), 89-117.

Batterman, R. W. (1993). Defining chaos. Philosophy of Science, 60(1), 43-66.

Bishop, R. C. (2002). Chaos, indeterminism, and free will. In The Oxford handbook of free will

Danforth, Christopher M. (April 2013). "Chaos in an Atmosphere Hanging on a Wall". Mathematics of Planet Earth 2013. Retrieved 12 June 2018.

Garson, J. W. (1995). Chaos and free will. Philosophical Psychology, 8(4), 365-374.

Gleick, J. (1987). *Chaos: Making a New Science*. New York: Viking.

Koblyakov, A. (1995). Semantic aspects of self-similarity in music. Symmetry: culture and science, 6(2), 297-300.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2), 130-141.

Martin-Löf, P. (1966). The definition of random sequences. Information and control, 9(6), 602-619.

Motter, A. E., & Campbell, D. K. (2014). Chaos at fifty. In The Best Writing on Mathematics 2014 (pp. 270-287). Princeton University Press.

Ott, E. (2002). Chaos in dynamical systems. Cambridge university press.

Ott, E., Grebogi, C., & Yorke, J. A. (1990). Controlling chaos. Physical review letters, 64(11), 1196.

Pareyon, G. (2011). On Musical Self-Similarity: Intersemiosis as Synecdoche and Analogy (Doctoral dissertation, University of Helsinki).

Peitgen, H. O., Jürgens, H., & Saupe, D. (2006). Chaos and fractals: new frontiers of science. Springer Science & Business Media.

Peitgen, H. O., Jürgens, H., & Saupe, D. (2013). Fractals for the classroom: part one introduction to fractals and chaos. Springer Science & Business Media.

Salter, J. R. (2009). Chaos in music: Historical developments and applications to music theory and composition. The University of North Carolina at Greensboro.

Strogatz, S. H. (2015). Nonlinear Dynamics and Chaos. CRC press.

Sundbye, L. (2018). *Discrete Dynamical Systems, Chaos Theory and Fractals.*

Tang, Y., Kurths, J., Lin, W., Ott, E., & Kocarev, L. (2020). Introduction to focus issue: When machine learning meets complex systems: Networks, chaos, and nonlinear dynamics. Chaos: An Interdisciplinary Journal of Nonlinear Science, 30(6), 063151.

Veissière SPL, Constant A, Ramstead MJD, Friston KJ, Kirmayer LJ. (2020) Thinking through other minds: A variational approach to cognition and culture. Behavioral and Brain Sciences 43, e90: 1–75. doi:10.1017/ S0140525X19001213

Werndl, C. (2009). What are the new implications of chaos for unpredictability?. The British Journal for the Philosophy of Science, 60(1), 195-220.

Yadegari, S. D. (1991). Using Self-Similarity for Sound/Music Synthesis. In Proceedings of the International Computer Music Conference (pp. 423-423). International Computer Music Association.

Yadegari, S. D. (1992). Self-similar synthesis on the border between sound and music (Master's thesis, Massachusetts Institute of Technology).

Yadegari, S. D. (2004). The radif as a basis for a computer music model: union of philosophy and poetry through self-referentiality (Doctoral dissertation, University of California, San Diego).

Zuchowski, L. C. (2017). A philosophical analysis of chaos theory. Springer.

References for Granular Synthesis

Clarke, M. (1996) Composing at the intersection of time and frequency. Organised Sound, 1(2).

Deutsch, D. (1999). Grouping mechanisms in music. In The psychology of music (pp. 299-348). Academic Press.

Gabor, D. (1946). 'Theory of Communication' The Journal of the Institution of Electrical Engineers. London: Unwin Brothers. 93(3): 429-457.

Laske, Otto E. (1992). "Furies and Voices: Composition-Theoretical Observations." Otto Laske: Navigating New Musical Horizons, Ed. Jerry Tabor. Greenwood Press, 1999.

Roads, C. (1978). Automated Granular Synthesis of Sound. Computer Music Journal, 2(2), 61-62. doi:10.2307/3680222

Roads, C. (1988). 'Introduction to Granular Synthesis' Computer Music Journal. MIT Press 12(2):11-13.

Roads, C. (2002). Microsound. Book and audio compact disc. MIT Press. Cambridge, MA.

Roads, C. (2012, May). From grains to forms. In Proceedings of the international Symposium Xenakis. La musique electroacoustique/Xenakis. The electroacoustic music, M. Solomos, Ed., universite Paris (Vol. 8).

Truax, B. (1988). Real-time granular synthesis with a digital signal processor. Computer Music Journal, 12(2), 14-26.

Truax, B. (1990). Composing with Real-time Granular Sound. Perspectives of New Music. USA: Hamilton Printing Company. Pages 120-134.

Truax, B. (2003). Music and Science Meet at the Micro Level: Time-Frequency Methods and Granular Synthesis. Presented at the Musica Viva conference, Coimbra, Portugal, Sept. 2003

Xenakis, I. (1971). Formalized Music. Bloomington: Indiana University Press.