Hierarchical Music Structure Analysis, Modeling and Resynthesis: A Dynamical Systems and Signal Processing Approach.[a]

by

Víctor Gabriel Adán

B.M., Universidad Nacional Autónoma de México (2002)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September, 2005

Author . . . .
Program in Media Arts and Sciences
August 5, 2005

Certified by . . . .
Barry L. Vercoe
Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by . . . .
Andrew B. Lippman
Chairman, Departmental Committee on Graduate Students

[a] This copy contains minor corrections (grammatical and of *errata*) made after the thesis was submitted.

Hierarchical Music Structure Analysis, Modeling and Resynthesis: A Dynamical Systems and Signal Processing Approach.[a]

by Víctor Gabriel Adán

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on August 5, 2005,
in partial fulfillment of the requirements
for the degree of Master of Science

[a] This copy contains minor corrections (grammatical and of *errata*) made after the thesis was submitted.

Abstract

The problem of creating generative music systems has been approached in different ways, each guided by different goals, aesthetics, beliefs and biases. These generative systems can be divided into two categories: the first is an *ad hoc* definition of the generative algorithms, the second is based on the idea of modeling and generalizing from preexistent music for the subsequent generation of new pieces. Most inductive models developed in the past have been probabilistic, while the majority of the deductive approaches have been rule based, some of them with very strong assumptions about music. In addition, almost all models have been discrete, most probably influenced by the discrete nature of traditional music notation.

We approach the problem of inductive modeling of high level musical structures from a dynamical systems and signal processing perspective, focusing on motion *per se* independently of particular musical systems or styles. The point of departure is the construction of a state space that represents geometrically the motion characteristics of music. We address ways in which this state space can be modeled deterministically, as well as ways in which it can be transformed to generate new musical structures. Thus, in contrast to previous approaches to inductive music structure modeling, our models are continuous and mainly deterministic. We also address the problem of extracting a hierarchical representation of music from the state space and how a hierarchical decomposition can become a second source of generalization.

Thesis supervisor: Barry L. Vercoe, D.M.A.
Title: Professor of Media Arts and Sciences

Thesis Committee

Thesis supervisor..

Barry L. Vercoe
Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis reader..

Rosalind Picard
Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis reader..

Robert Rowe
Professor
New York University

Acknowledgments

I would first of all like to thank my advisor, Barry Vercoe, for taking me into his research group and providing me with a space of complete creative freedom. None of this work would have been possible without his kind support.

Many thanks to the people who took the time to read the drafts of this work and gave me invaluable criticisms, particularly my readers: Professors Rosalind Picard, Robert Rowe and again Barry Vercoe.

Many special thanks go to the other *Music, Mind and Machine* group members: Judy Brown, Wei Chai, John Harrison, Nyssim Lefford, and Brian Whitman for being such incredible sources of inspiration and knowledge.
Special thanks to Wei for her constant willingness to listen to my math questions and for her patience. To Brian for being my second “advisor”. His casual, almost accidental lessons gave me the widest map of the music-machine learning territory I could have wished for. Thanks to John for being a tremendous “officemate” and friend. His intelligence, skepticism and passion for music and technology made him a real pleasure to work with and a constant source of inspiration.

Many other Media Lab members deserve special mention as well: Thanks to Hugo Solis for helping me with my transition into the lab, doing everything from paper work to housing. I probably wouldn’t have come here without his help. Also thanks to all *45 Banks* people for being such wonderful friends and roommates: Tristan Jehan, Luke Ouko, Carla Gomez and again, Hugo.

In addition I have to thank all the authors cited in this text. This thesis wouldn’t have been without their work. I want to give special thanks to Bernd Schoner for his beautiful MIT theses. They were a principal source of inspiration and insight for the present work.

My greatest gratitude to my family. Their examples of perseverance and faith have been my wings and strength.

Last but certainly not least, I want to thank my dear wife "Cosi" for her love and support (and her editing skills). She is the most important rediscovery I have made in my search for the meaning of it all.

Table of Contents

CHAPTER ONE

Introduction and Background

1.1 Motivations

There are two motivations for the present work: the first motivation comes from my interest in music analysis. Several music theories have been developed as useful tools for analyzing and characterizing music. Most of these theories, such as Riemann's theory of tonal music, Schenkerian analysis [31] and Forte's atonal theory, are specific to particular styles or systems of composition. It would be interesting to see the development of an analytical method useful for any kind of music. This method would have to be based on features that are present in all music, the element of motion being the only constant characteristic. While the importance of motion in music may be obvious, no work that I know of has systematically studied music from this perspective.

The second motivation comes from my interest in understanding the behavior of my musical imagination. The process of composing frequently involves writing, or recording in some way, the imagined and invented sound evolutions.Multiple paths and combinations are explored during the process of imagining new music, so the score ultimately written is but an instance of the paths considered. Thus, this representation is like a photograph of the dynamic, ever-changing world of the imaginary. Why choose one sequence or combination over another? Is there a single best sequence, a better architecture? My personal answer to this question is no. This has led me to become interested in pursuing the creation of a meta-music: a pattern capable of generating the fixed notated music along with its other implied possibilities.

1.2 Algorithmic Composition

Algorithmic composition is commonly defined as the invention and explicit definition of music generating recipes commonly implemented as automata.[1] In algorithmic composition the composer does not decide on the musical events directly but rather creates an intermediary that will choose or generate the specific musical events for him. This intermediary can be a mechanical automaton or a mathematical description that constrains the possible musical events that can be generated. The broadness of the definition suggests the existence of a continuous spectrum in the levels of detachment between the composer and the actual sounds produced.

We find examples of algorithmic composition as early as 1029 in music theorist Guido D'Arezzo's *Micrologus.* Guido discusses a method for automatically composing melodies using any text by assigning each note in the pitch scale to a vowel [26]. Because there are more pitches than vowels, the composer is still free to choose between the multiple pitch options available. Similarly, Miranda takes preexisting algorithms from cellular automata and fractal theory for automatic music generation [28].

Today, algorithmic composition is as much an aesthetic paradigm, a way of thinking about composition, as it is a methodology. In algorithmic composition it's not so much the specific sequence of sounds or notes that matters as it is the algorithm that produces the sequence. i.e. it is as much (or more) about *how* a thing is generated as it is about *what* is generated. Music is a byproduct—and a measure—of the algorithm, which becomes the artifact of main interest.

While inventing algorithms for music composition can be a fascinating endeavor, in this paper we are interested in learning from musical examples and in developing a generative pattern from them.

1.3 Modeling

All models are wrong, but some are useful.

George E.P. Box

[1] For a complete definition of algorithm and a discussion of how they relate to music composition, see [26].

There are no best models *per se*. The most effective model will depend on the application and goal. Different goals suggest different approaches to modeling. As expressed by our motivations, our models intend to serve a double purpose: the first is to obtain some understanding about the inner workings of a given piece and, hopefully, gain insight into the composer's mind. The second is for the models to be a powerful composition tool. It is difficult, if not impossible, to come up with a model that achieves these two goals simultaneously for a variety of pieces because a generative model might not be the most adequate for analysis and vise versa. Music structure is so varied, so diverse, that it seems unlikely that a single modeling approach could be used successfully for all music and for all purposes.

What are the criteria for choosing a model? Is the model simple? The *Minimum Description Length* principle [33], which essentially defines the best model as that which is smallest with regards to both form and parameter values, is a measure of such a criterion. Other criteria to consider are:

Robustness: Does it lend itself well to a variety of data (e.g. musical pieces)?
Prediction: Can it accurately predict short term or long term events?
Insight: Does it provide new meaningful information about the data?
Flexibility: As a generative system, what is the range or variety of new data that the model can generate?

1.3.1 Models of Music

Deduction vs. Induction

Brooks et al. describe two contrasting approaches to machine modeling: the *inductive* and the *deductive* [4]. In deductive reasoning, one moves from a general theory to a particular conclusion. In inductive reasoning one goes form particular observations to a principle that applies generally.

In a deductive model we feed the machine a theory of music. We encode the rules and generalizations and the machine deduces the details to generate new examples. Generally speaking, algorithmic compositions are mostly deductive. In an inductive approach, the machine makes the generalization from examples. Given a piece (or set of pieces) of music, the machine analyzes, learns from the example(s), and develops a general model from which novel pieces can then be generated.

Attempting to have the machine automatically derive the structure and the generalization is a more interesting and challenging problem for us at this juncture. It seems to us that in a deductive approach we are biasing the data. We are telling the machine how to think about music and how to process it. In an inductive approach the attempt is to have the machine figure out what music is.

So we are back to music analysis! But this time we might analyze our music from the data itself, and then develop a model and/or a theory, rather than analyzing the data from a theoretical framework *a priori*. In other words we want to build a model from the data, and not simply generate data from a given model.

Continuous vs. Discontinuous

Most generative music systems and analysis methods assume a discrete musical space.[2] This is a natural assumption since the dominating musical components in western music have been represented with symbols for discrete values.[3] The implication is that most models of music part from the idea that a piece is a sequence of symbols taken from a finite alphabet. Almost all the generative systems we are aware of are discrete ([21][4][19][38][12][8][9][41][7][32][30][29][3]). In this paper, however, we explore the possibility of a continuous model of music that allows for the analysis and modelling of perceptually relevant data not encoded in a musical score.

Deterministic vs. Probabilistic

> *There is no way of proving the correctness of the position of 'determinism' or 'indeterminism'. Only if science were complete or demonstrably impossible could we decide such questions.*

[2] Since the classical period, notation for loudness has commonly included symbols for continuous transitions, but loudness is typically not considered important enough to be studied.

[3] Indeed, the recurrent inclusion of the continuum in notated pitch space (starting probably with Bartok, continuing with Varèse and Xenakis, and culminating in Estrada) has made it difficult for some music theoretic views to approach these kinds of music.

Mach (Knowledge and Error, Chapt XVI.11)

Should a music structure model be deterministic or stochastic? Consider two contrasting musical examples: on one extreme there is Steve Reich's *Piano Phase*. The whole piece consists of two identical periodic patterns with slightly different *tempi* (or frequencies).[4] *Piano Phase* can be straightforwardly understood as a simple linear deterministic system. On the other extreme we can place Xenakis' string quartet *ST/4, 1-080262*. It would make sense to model *ST/4, 1-080262* stochastically since we know it was composed in this way! In between these two extremes there are a huge variety of compositions with much more complex and intricate structures. There may be pieces that start with clearly perceivable repeating patterns and that gradually evolve into something apparently disorganized. A piece like this might best be modeled as a combination of deterministic and stochastic components that are a function of time. In addition, these combinations may occur at multiple time scales, in which case it would be better modeled as a combination of a deterministic component at one level, and a stochastic component at another. Thus, rather than trying to find a single global model for a whole piece, we might want to model a piece as a collection of multiple, possibly different, models.

Brief Thoughts on Markov Models

The majority of the *inductive* machine models of music are variations of discrete Markov models. Remember that an kth order Markov model of a time series is a probabilistic model where the probability of a value at time step $n + 1$ in a sequence s is given by the k previous values: $p(s_{n+1}|p(s_n, s_{n-1}, \ldots, s_{n-k+1})$. Why discrete and why Markovian? Several papers on Markov models of music are based on the problem of reducing the size and complexity of the conditional probability tables by the use of trees and variable orders([41][3][30]). Again, the discrete nature of the models most probably comes from the view of music as a sequence of discrete symbols taken from an alphabet. Why not model the probability density functions parametrically, as with mixtures of gaussians?

Are Markov models really good inductive models of music? What kind of generalization can they make? Most applications of discrete Markov

[4] The point of the piece is the perception of the evolution of the changing phase relationships between the two patterns.

models estimate the probability functions from the training data. Usually, zero probabilities are assigned to unobserved sequences. If this is the case, it is impossible for a simple kth order Markov model to generate any new sequences of length $k+1$. All new and original sequences will have to be of length $k+2$ or greater. Take for example the following simple sequence which we assume to be infinite:

$$1, 2, 3, 2, 1, 2, 3, 2, 1, 2, 3, 2, \ldots \tag{1.1}$$

A first order model of this sequence can be constructed statistically by counting the relative frequencies of each value and of each pair of values. The relative frequencies of all possible pairs derived from this sequence can be easily visualized in the following table, where each fraction in the table represents the number of times a row value is immediately followed by a column value, divided by the total number of consecutive value or sample pairs found in the sequence. From these statistics we can now derive the marginal probabilities of individual values, and by Bayes' rule we find the conditional probabilities:

Table 1.1: Relative frequency of all pairs of values found in sequence 1.1

	1	2	3
1	0	1/4	0
2	1/4	0	1/4
3	0	1/4	0

$$p(s_{n+1}|s_n) = \frac{p(s_{n+1}, s_n)}{p(s_n)}$$

From this table we can see that new sequences generated by this joint probability function will never produce pairs (1,1), (1,3), (3,1), (3,3), (2,2) or extrapolate to other values, such as (4,5). In other words, all length 2 sequences that this model can generate are strictly subsets of sequence 1.1, while those of length 3 or greater may or may not appear in the original sequence. To overcome this limitation one could give unobserved sequences a probability greater than zero, but this is essentially adding noise. Looking at generated sequence segments of length 3 or greater we may now wonder how these are related to the original training pieces. The limitation of this model soon becomes apparent. New sequences generated from this model will have some structural resemblance to the original only at lengths $l \leq k+1$ ($l \leq 2$ in this example), but not at greater lengths. From the first order model in the present example it is possible to obtain the following sequence:

$$1, 2, 3, 2, 3, 2, 1, 2, 3, 2, 1, 2, 1, 2, 1, 2, 1, 2, 3, 2, 3, 2, 1, 2, 3, 2, 1, \ldots$$

Because $p(3|2) = p(1|2) = 0.5$, the generated sequence will fluctuate between 1 and 3 with equal probabilities every time a 2 appears. This misses what seems to be the essential quality of the sequence: its periodicity. Increasing the order of the model to 2, we are able to capture the periodicity unambiguously. But what generalization exists? Now the model will generate the whole original training sequence exactly. This is the key idea we explore in this work. We want the model to be able to abstract the periodicity and generate new sequences that are different from the original but that have the same essentially periodic quality. This information can be obtained by looking for structure and patterns in the probability functions derived from the statistics, but in this work we will present a different approach.

One more problem is that of stationarity. In our example the sequence repeats indefinitely, making a static probabilistic model adequate. But music continuously evolves and is very seldomly static. Sequences generated with a simple model like the one discussed above may resemble the original at short time scales, but the large scale structures will be lost. This problem could be addressed by dynamically changing the probability functions, i.e. with a stochastic model. Modeling non-stationary signals can also be done through a hierarchical representation, where the conditional probabilities are estimated not only sequentially, but also vertically across hierarchies. Some concrete applications of this approach can be found in image processing [10], and to our knowledge, there have not been similar approaches in music modeling. Similar problems have been observed using other methods such as neural networks [29], where the output sequences resembled the original training sequence only locally due to the note-to-note approach to modeling.

The Markov model example we have given here is certainly extremely simplistic, but hopefully it makes clear some of the problems of this approach to music modeling for the purpose of generating novel pieces.

1.3.2 Our Approach to Music Structure Modeling

As suggested in our motivations, our main interest is the abstraction of the essential qualities of the dynamic properties of music and their use as sources for the generation of novel pieces. By dynamics we mean the qualitative aspects of motion in music, rather than the loudness component as is usually used by musicians. Here we approach musical sequences in a similar way as a physicists would approach the motion of physical objects.

The notion of the dynamical properties of a musical sequences is rather abstract, and we hope to clarify it with a concrete example. First, consider the following question: why can we recognize a tune even when replacing its pitch scale (e.g. changing from major to minor), or when the pitch sequence is inverted? What are the constants that are preserved that allow us to recognize the tune?

Take for example the first 64 notes (8 measures) of Bach's Prelude from his Cello Suite no.1 (Figure 1-1). There are multiple elements that we

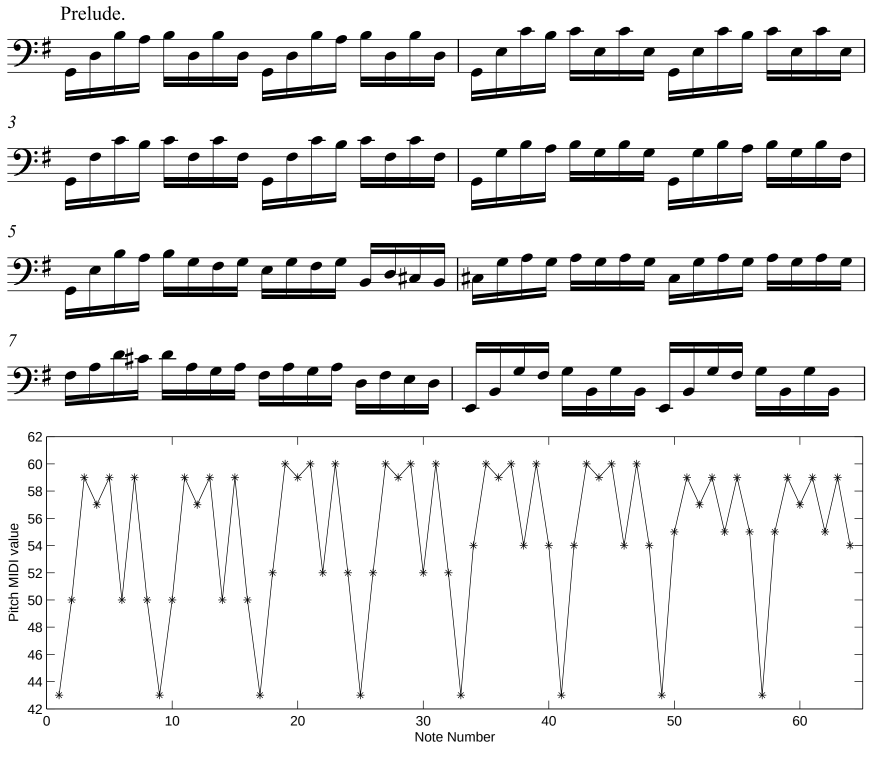


Figure 1-1: ***Top*: First 8 measures of Bach's Prelude from Cello Suite no.1. *Bottom*: Alternative notation of the same 8 measures of Bach's Prelude. Here the note sequence is plotted as connected dots to make the contour and the cyclic structure of the sequence more apparent.**

can identify in the series: the pitches, the durations, the scales and harmonies the pitches imply, the pitch intervals and the temporal relationships between these elements. How can we characterize these temporal relationships? Can we say something about the regularity/irregularity of the sequence, the changes of velocity, the contour? A reduced representation of the motion in Bach's Prelude might look something like (up, up, down, up, down, up, down, down, ...) repeating several times. This is a useful but coarse approximation, preserving only the ordering of intervals. We might also want to preserve some information about the size of the intervals, their durations and the quality of the motion from one point to the next: is it continuous, is it discrete? Whatever the case, we see from Figure 1-1 (*Bottom*) that this coarse sequence of ups and downs is a principal structure in the Prelude fragment, repeating and transforming gradually to match some harmonic sieve that changes every two measures. Every two measures the sequence is slightly different, but in essence the type of motion of the eight measures is the same. Thus, we can consider the sequence as being composed of two independent structures: the dynamics and the sieves though which these are filtered or quantized.

In this thesis we focus on the analysis and modeling of these dynamic properties. We address ways in which we can decompose, transform, represent and generalize the dynamics for the purpose of generating new ones. Rather than trying to extend on the Markovian model as suggested above, we approach the problem from a deterministic perspective. We explore the use of a method that allows for a geometric representation of time series, and discuss different ways to model and generalize the geometry deterministically.

CHAPTER TWO
Musical Representation

Structure in music spans about seven orders of magnitude, from approximately 0.0001 seconds ($\frac{1}{10kHz}$) to about $1,000$ seconds (≈ 17 min.) [11]. In the present work we focus on the upper four orders (between 0.1 secs. and 1,000 secs.) for two reasons: First, these orders correspond to those typically represented in musical scores. Second, there is a clear difference between the way we perceive sound below and above approximately 20Hz.

Below this threshold, we hear pitch and timbre, and above it we hear rhythm and sound events or streams. The perception can be so clearly different that they feel like two totally independent things: a high-level control signal driving high-speed pressure changes. These high-level control signals are what we are interested in modeling and transforming. Thus, the musical representations we will use are not those of recorded sound (either in time or frequency domain), but the more abstract representations of these high-level signals.

As of today, it is still very difficult to extract these high level control signals from the actual audio signal. Good progress has been made in tracking the pitch of individual monophonic instruments and in some special cases from polyphonic textures. But the technology is still far from achieving the audio segregation we would like. Therefore, except for simple cases where the evolution of pitch, loudness and some timbral features can be relatively well extracted (such as simple monophonic pieces), our point of departure is the musical score.

Traditionally, two basic types of score exist. In the first, the score represents the evolution of perceptual components, such as pitch, loudness and timbre–thus, the perception of these components is usually a requirement for their performance. In the second, traditionally called *tablature* notation, the score is a representation of the mechanics involved in the production of the music. In essence they are both control signals driving some salient feature of sound or some mechanism for its production. The following are a few ways in which these high level control signals can be represented before being modeled and transformed. Some representations will be more appropriate than others depending on their use and the type of music to be studied.

Representation of Time

The simplest and most common way of representing a digitized scalar signal is as a succession of values at equal time intervals:

$$s[n] = \quad s_0, \quad s_1, \quad \ldots, \quad s_n \tag{2.1}$$

Since it is assumed that all samples share the same duration, this information need not be included in the series.

The same is true for a multidimensional signal where each dimension describes the evolution of a musical component. For example, a series describing the evolution of pitch, loudness and brightness might look like this:

$$\mathbf{s}[n] = \begin{bmatrix} p_0, & p_1, & \ldots, & p_n \\ l_0, & l_1, & \ldots, & l_n \\ b_0, & b_1, & \ldots, & b_n \end{bmatrix} \tag{2.2}$$

For the specific case where there is more than one instrument or sound source, as in a three voice fugue, a chorale or even possibly an entire orchestra, the series can again be extended as:

$$\mathbf{s}[n] = \begin{bmatrix} p_{a,0}, & p_{a,1}, & \ldots, & p_{a,n} \\ l_{a,0}, & l_{a,1}, & \ldots, & l_{a,n} \\ b_{a,0}, & b_{a,1}, & \ldots, & b_{a,n} \\ & & & \\ p_{b,0}, & p_{b,1}, & \ldots, & p_{b,n} \\ l_{b,0}, & l_{b,1}, & \ldots, & l_{b,n} \\ b_{b,0}, & b_{b,1}, & \ldots, & b_{b,n} \\ \vdots & & & \\ p_{m,0}, & p_{m,1}, & \ldots, & p_{m,n} \\ l_{m,0}, & l_{m,1}, & \ldots, & l_{m,n} \\ b_{m,0}, & b_{m,1}, & \ldots, & b_{m,n} \end{bmatrix} \tag{2.3}$$

While this is a simple representation scheme, it is not necessarily the best in all cases. The problem with this representation has to do primarily with the temporal overlapping of events. Consider a polyphonic instrument such as the piano. With this instrument it is possible to articulate multiple notes at the same time. How should we represent a sequence of pitches that overlap, starting and ending at different times? One way is to have a series that has as many dimensions as the number or keys in the piano, where each dimension represents the evolution of the velocity of the attack and release of each key:

$$\mathbf{s}[n] = \begin{bmatrix} v_{1,0}, & v_{1,1}, & \ldots, & v_{1,n} \\ v_{2,0}, & v_{2,1}, & \ldots, & v_{2,n} \\ \vdots & & & \\ v_{m,0}, & v_{m,1}, & \ldots, & v_{m,n} \end{bmatrix} \tag{2.4}$$

How could we reduce the number of dimensions and still have a meaningful representation? A tempting idea might be to separate a piano piece into multiple monophonic voices and to assign each voice to a dimension in a multidimensional polymelodic series as in 2.3. But in addition to being an arbitrary decision in most cases (not all piano pieces are conceived as a counterpoint of melodies), the main problem is that even in single melodic lines there may still be overlaping notes through *legatissimo* articulation.

An alternative representation is the Standard Midi File (SMF) approach, where time is included explicitly in the representation by indicating the absolute position of each event in time or the time difference: the Inter Onset Interval (IOI). In addition to this time information there are the duration of each event and the component values. The most economical form of this representation, Format 0, combines all separate sources or instruments into a few dimensions, one of which specifies the instrument to which the event corresponds. The following is an example of this format, where *ioi* stands for Inter Onset Interval, *d* for duration, *p* for pitch, *v* for velocity (or loudness) and *ch* for channel (or instrument):

$$\mathbf{s}[n] = \begin{bmatrix} ioi_0, & ioi_1, & \ldots, & ioi_n \\ d_0, & d_1, & \ldots, & d_n \\ p_0, & p_1, & \ldots, & p_n \\ v_0, & v_1, & \ldots, & v_n \\ ch_0, & ch_1, & \ldots, & ch_n \end{bmatrix} \tag{2.5}$$

With this format we can now represent overlapping events with only a few dimensions. This is an adequate representation for the piano, due to the discrete nature and limited control possibilities of the instrument. Yet, it is far from being a good numeric replacement for a traditional music score in general, the main problem being the loss of independence between the multiple parameters. Because the explicit representation of time intervals can be useful, we can still take advantage of this feature by defining pairs of dimensions for each parameter: one for the parameter values and another for their durations. Besides providing some insight into the rhythmic structure of a sequence, it can also significantly reduce the number of data samples:

$$\mathbf{s}[n] = \begin{bmatrix} p_0, & p_1, & \ldots, & p_n \\ d_0, & d_1, & \ldots, & d_n \\ & & & \\ v_0, & v_1, & \ldots, & v_n \\ d_0, & d_1, & \ldots, & d_n \end{bmatrix} \tag{2.6}$$

While this representation lends itself best for discrete data, we can assume some kind of interpolation between key points and still make it useful for continuous data.

Summary

There are multiple ways in which musical data can be represented. Different representations have different properties: some provide compact representations, while others are more flexible. In addition, some may be more informative about certain aspects of the music than others. The choice of representation will also depend on the type of transformation we are interested in applying to the data. In the present work we use these three basic representations (constant sampling rate representation, variable sampling rate representation, and SMF Format 0) in different situations for the purpose of analysing and generating new musical sequences.

CHAPTER THREE
State Space Reconstruction and Modeling

3.1 State Spaces

One of the most important concepts in this work is that of *state space* (or *phase-space*). A state space is the set of all possible states or configurations available to a system. For example, a six sided die can be in one of six possible states, where each state corresponds to a face of the die. Here the state space is the set of all six faces. As a musical example consider a piano keyboard with 88 keys. All possible combinations of chords (composed from 1 to 88 keys) in this keyboard constitute the state space, and each chord is a state. These two examples constitute finite state spaces because they have a finite number of states. Some systems may have an infinite number of states; for example a rotating height adjustable chair. A chair that can rotate on its axis and that can be raised or lowered continuously has an infinite number of possible positions. Yet, in practical terms, many of these positions are so close to each another that sometimes it makes sense to discretize the space and make it finite (for example, by grouping all rotations within a range of $\frac{\pi}{8}$ radians into one state). While this system has an infinite number of possible states, it has only two degrees of freedom: up-down motion and azimuth rotation.

State spaces can be represented geometrically as multidimensional spaces where each point corresponds to one and only one state of the system. In the chair example, since only two degrees of freedom exist, we can represent the whole state space in a two dimensional plane. Yet, we might like to keep the relations of proximity between states in the representation as well. Therefore we embed the two dimensional state space of the rotating chair in a three dimensional space in such a way that the states that are physically close to each other are also close in state space. This results in a cylinder, and is a "natural" way of configuring the points in the state space of the chair.

An analogous musical example is that of pitch space. This one-dimensional space can be represented with a line, ranging from the lowest perceivable pitch to the highest. But this straight line is not a good representation of human perception of pitch in terms of similarity. Certain frequency ratios like the octave (2:1) are perceived to be equivalent or more closely related to each other than, for example, minor seconds. In 1855 Drobisch proposed representing pitch space as a helix, placing octaves closer to each other than perceptually more distant intervals [35]. Thus, we have a similar case to that of the chair, where a low dimensional space is embedded in a higher dimensional euclidean space to represent similar states by proximity (Figure 3-1).

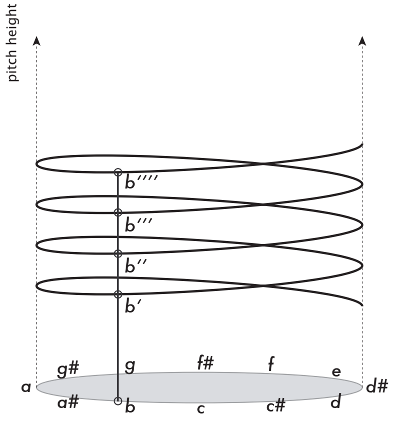


Figure 3-1: **Helical representation of pitch space. A one-dimensional line is embedded in a three-dimensional euclidean space for a better representation of pitch perception.**

3.2 Dynamical Systems

We have discussed the notion of state space and have given a few examples of their representation. But for the case of music, it is the temporal relationships between states that we are most interested in (i.e. their dynamics), rather than the states themselves. Given this interest, it comes as no surprise that our first approach to studying the qualitative characteristics of motion in music comes from the long tradition of the study of dynamical systems in physics. We will not discuss here the history of this vast discipline, but only present some key ideas and important results that will help us analyze, model and generalize musical structures from which we will hopefully generate novel pieces of music.

3.2.1 Deterministic Dynamical Systems

> *We ought then to regard the present state of the universe as the effect of its anterior state and as the cause of the one which is to follow. Given for one instant an intelligence which could comprehend all the forces by which nature is animated and the respective situation of the beings who compose it –an intelligence sufficiently vast to submit these data to analysis– it would embrace in the same formula the movements of the greatest bodies of the universe and those of the lightest atom; for it, nothing would be uncertain and the future, as the past, would be present to its eyes.*
>
> Pierre-Simon Laplace (Philosophical Essay on Probabilities,
> II. Concerning Probability)

A deterministic dynamical system is one where all future states are known with absolute certainty given a state at an instant in time. If the future state of a system depends only on the present state, independently of the time at which the state is found, then the system is said to be *autonomous*. Thus, the dynamics of an autonomous system are defined only in terms of the states themselves and not in terms of time.

In the case of an m-dimensional state space, the dynamics of an autonomous deterministic system can be defined as a set of m first-order ordinary differential equations for the continuous case (also called a *flow*) [22]:

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t)), \quad t \in \mathbb{R} \tag{3.1}$$

or as an m-dimensional *map* for the discrete case:

$$\mathbf{x}_{n+1} = F(\mathbf{x}_n), \quad n \in \mathbb{Z} \tag{3.2}$$

3.2.2 State Space Reconstruction

What if we know nothing about the nature of a system, and all we have access to is one of its observables? Consider for example the flame of a candle as our system of interest. Suppose that, similarly to Plato's cave of shadows from his *Republic, Book VII*, we do not see the candle directly, but only the motion of shadows of objects projected on the wall. Thus, we have a limited amount of information about the candle's motion and all its degrees of freedom. Can we infer the motion of the candle's flame in its entirety from the motion of the shadows?

Suppose now that our observation is a piece of music; it is the shadow moving on the wall. We now want to know what the nature of the "system" that generated the given piece of music is. Can we automatically infer the hidden structure of the mind responsible for the generation of the piece and, from this structure, generate other musical possibilities?[1]

We now state the question more formally. Given a series of observations $s(t) = h(\mathbf{w}(t))$ that are a function of the system $\mathbf{W}$, is it possible to reconstruct the dynamics of the state space of the system from these observations? Can we find $h^{-1}(s(t))$? If $s(t)$ is a scalar observation sequence, is it possible to recover the high dimensional state space that generated the observation? Takens' theorem [40] states that it is possible to reconstruct a space $\mathbf{X} \in \mathbb{R}^m$ that is a smooth differomorfism (topologically equivalent) of the true state space $\mathbf{W} \in \mathbb{R}^d$ by the *method of delays*. This method consists of constructing a high dimensional embedding by taking multiple delays of $s(t)$ and assigning each delay to a dimension in the higher dimensional space X:

$$\begin{aligned}
\mathbf{x}(t) &= (s(t), s(t-\tau), \ldots, s(t-(m-1)\tau)) \\
&= (h(\mathbf{w}(t)), h(\mathbf{w}(t-\tau)), \ldots, h(\mathbf{w}(t-(m-1)\tau)) \\
&= H(\mathbf{w}(t))
\end{aligned}$$

The theorem states that this is true if the following conditions are met:

1. System $\mathbf{W}$ is continuous with a compact invariant smooth manifold A of dimension d, such that A contains only a finite number of equilibria, contains no periodic orbits of period τ or 2τ and contains only a finite number of periodic orbits of period $p\tau$, with $3 \leq p \leq m$.

[1] Evidently, in the context of this work, this example should not be taken literally, but you get the idea.

2. $m \geq 2d + 1$.
3. The measurement function $h(\mathbf{w}(t))$ is C^2.
4. The measurement function couples all degrees of freedom.

The condition that $m \geq 2d + 1$ guarantees that the embedded manifold does not intersect itself. In many cases a value of m smaller than $2d + 1$ will work because the number of effective degrees of freedom of the system may be smaller due to dissipation or correlation between the dimensions in state space [37].

3.2.3 Calculating the Appropriate Embedding Dimension

How do we decide on a good dimension m for the embedding? How do we know if the dimensionality we've chosen for the embedding is high enough to capture all the degrees of freedom of the system? For deterministic time series we want each state to have a unique velocity associated with it, i.e. if $\mathbf{x}_n = \mathbf{x}_k$ then $F(\mathbf{x}_n) = F(\mathbf{x}_k)$. This implies that for the case of continuous time series, trajectories should not cross in state space. In other words, points that are arbitrarily close to each other, both in space and time, will have very similar velocities.

Correlation Sum

A simple way of estimating a good embedding dimension for state space reconstruction of deterministic systems is to find points that overlap or that are closer than ϵ to each other. The correlation sum simply counts the number of distances smaller than ϵ between all pairs of points $(\mathbf{x}_i, \mathbf{x}_j)$.

$$C(\epsilon) = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Theta\left(\epsilon - \|\mathbf{x}_i - \mathbf{x}_j\|\right) \tag{3.3}$$

Here $\Theta(x)$ is the Heaviside function, which returns 1 if $x > 0$ and 0 if $x \leq 0$. Throughout the text we use the Euclidean distance metric: $\|\mathbf{x} - \mathbf{y}\| = \sqrt{(x_1 - y_1)^2 + \ldots + (x_m - y_m)^2}$.

A good embedding dimension will be one where the correlation sum is very small in proportion to the total number of points. Because it is not clear what a good choice of ϵ is, the correlation sum is evaluated for different values of ϵ and different embedding dimensions m. Specifically for discrete series, with ϵ smaller than the quantization resolution δ we can count the number of points that fall in the same place. Thus, when we want a strictly bijective relation between the embedded trajectory and the observation sequence, we choose the smallest embedding dimension where the correlation sum equals zero, for $\epsilon < \delta$. i.e. $C(\epsilon < \delta) = 0$.

False Nearest Neighbors

A more sophisticated method for estimating the embedding dimension is the *false nearest neighbors* method [23]. This algorithm consists of comparing the distance between each point and its nearest neighbor in an m-dimensional lag space to the distance of the same pair of points in an $m+1$-dimensional space. If the distance between the pair of points grows beyond a certain threshold after changing the dimensionality of the space, then we have false nearest neighors. Kantz [22] gives the following equation for their estimation:

$$X_{fnn}(r) = \frac{\sum_{\text{all } n} \Theta\left(\frac{\|\mathbf{x}_n^{(m+1)} - \mathbf{x}_{(1)n}^{(m+1)}\|}{\|\mathbf{x}_n^{(m)} - \mathbf{x}_{(1)n}^{(m)}\|} - r\right) \Theta\left(\frac{\sigma}{r} - \|\mathbf{x}_n^{(m)} - \mathbf{x}_{(1)n}^{(m)}\|\right)}{\sum_{\text{all } n} \Theta\left(\frac{\sigma}{r} - \|\mathbf{x}_n^{(m)} - \mathbf{x}_{(1)n}^{(m)}\|\right)} \tag{3.4}$$

Where $\mathbf{x}_{(1)n}^{(m)}$ is the closest neighbor of $\mathbf{x}_n^{(m)}$ in m dimensions and σ is the standard deviation of the data. The first Heaviside function in the numerator counts the neighbors whose distance grows beyond the threshold r, while the second one discards those whose distance was already greater than σ/r. Cao [5] proposed a more elegant measure of false nearest neighbors. He further eliminates threshold value r by considering only the average of all the changes in distance between all points as the dimension increases, and then taking the ratio of these averages. He defines the mean

$$E[m] = \frac{1}{N} \sum_{\text{all } n} \frac{\|\mathbf{x}_n^{(m+1)} - \mathbf{x}_{(1)n}^{(m+1)}\|}{\|\mathbf{x}_n^{(m)} - \mathbf{x}_{(1)n}^{(m)}\|} \tag{3.5}$$

and $E1(m) = E[m+1]/E[m]$ as the ratio between the averages of consecutive dimensions. The ratio function $E1(m)$ describes a curve that grows slower as m increases, asymptotically reaching 1. The value of m where the growth of the curve is very small is the embedding dimension we are looking for.[2] Typically the value selected is slightly above the knee of the curve (Figure 3-2).

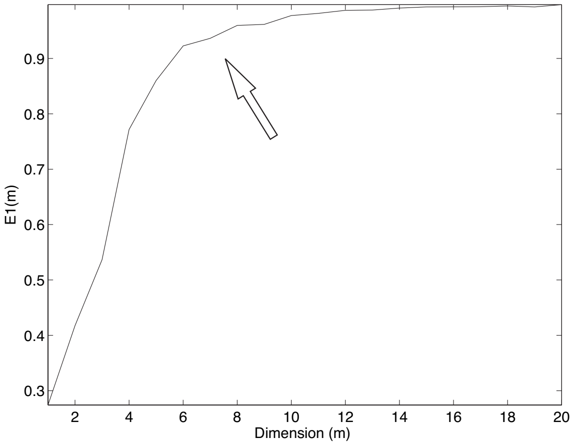


Figure 3-2: **Example of a typical behavior of E1(m). The function reaches 1 asymptotically, and a good choice for the embedding dimension is above the knee of the curve, where its velocity decreases substantially.**

3.2.4 Calculating the Time Lag τ

While Taken's theorem tells us what the minimum embedding dimension for state space reconstruction should be, it says nothing about how to choose the delay τ for the embedding. In theory, the choice of τ is irrelevant since the reconstructed state space is topologically equivalent to the system's manifold. In practice, though, factors such as observation noise and quantization make it necessary to find a good choice of τ for model estimation and prediction.

To better understand what a good choice of τ would be, we take as example one of the most popular systems in nonlinear dynamics: the

[2] This assumes that the time series comes from an attractor. If the series is noise for example, $E1(m)$ will never converge.

Lorenz attractor. The dynamics of the system are defined as

$$\begin{aligned}\dot{x} &= \sigma(y - x)\\ \dot{y} &= \rho x - y - xz\\ \dot{z} &= -\beta z + xy\end{aligned}$$

Figure 3-3 shows 4000 points of the trajectory of this attractor, with parameters $\sigma = -10$, $\rho = 28$, $\beta = -2.666\ldots$, and initial position $x = 0$, $y = 0.01$ and $z = -0.01$. Suppose our only observable is the x axis of the system (Figure 3-4). We construct the lag space from this observable as $\mathbf{x}_n = (x_n, x_{n-\tau}, x_{n-2\tau})$. Figure 3-5 shows the reconstructed state space from x for different values of τ. For $\tau = 1$, all the points in the reconstructed space fall along the diagonal. $\tau = 1$ is clearly small because the coordinates that compose each point are highly correlated and almost identical. As τ is increased, the embedding unfolds to reveal the structure of the attractor, which is clearest for values of τ between 8 and 14. As τ continues to increase, the geometry gradually distorts to an overcomplicated structure. While visual inspection is a good way of estimating the delay parameter, in many cases the dimensionality of the state space must be greater than three (apart from the fact that we ignore the shape of the real state spaces that generated the observation). Thus, a quantitative measure to estimate the appropriate value of τ is required. The method we will use is based on the above observation about the correlation that exists between coordinates in the reconstructed state

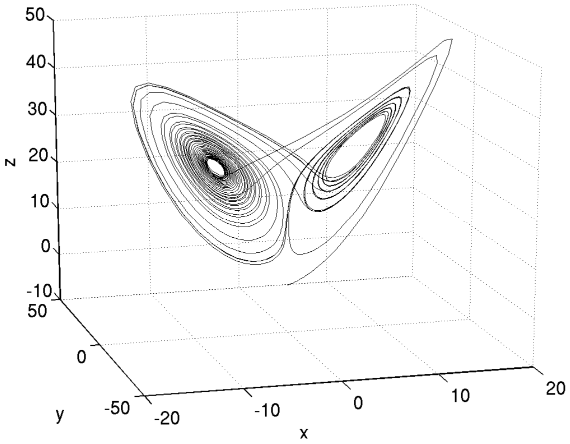


Figure 3-3: **Lorenz attractor with parameters $\sigma = -10$, $\rho = 28$, $\beta = -2.666\ldots$.**

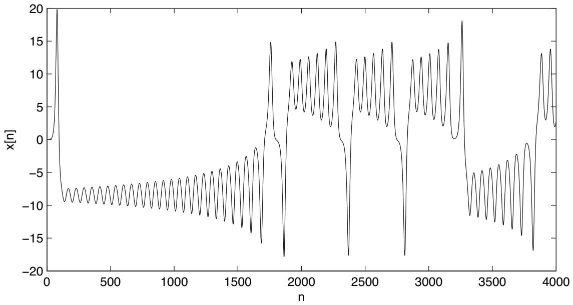


Figure 3-4: **Evolution of the x dimension (our observation) of the Lorenz system. The abscissa shows the sample steps of the sequences while the ordinate shows the actual value of the x axis of the Lorenz attractor.**

space. If coordinates are highly correlated, then the reconstructed trajectory will fall along the main diagonal of the space. This means that we want to find a τ large enough to make the correlations minimal yet small enough to avoid distortion of the reconstructed trajectory. A measure of the *mutual information* between a series and itself delayed by τ provides us with a reasonable answer to this question [24]. Let $p_{s_n}(i)$ be the probability that the series $s[n]$ takes the value i at any time n, and $p_{s_n,s_{n-\tau}}(i,j)$ be the probability that the series $s[n]$ takes the values i at time n and j at time $n-\tau$ for all n. The mutual information is then:

$$I(s_n; s_{n-\tau}) = \sum_i \sum_j p_{s_n,s_{n-\tau}}(i,j) \log_2 \frac{p_{s_n,s_{n-\tau}}(i,j)}{p_{s_n}(i)^2}. \tag{3.6}$$

Figure 3-6 is a plot of the mutual information of the observation x of the Lorenz system as a function of τ. The first minimum of the mutual information function is our choice of τ.

3.3 Modeling

3.3.1 Spatial Interpolation ≡ Temporal Extrapolation

A trajectory in the reconstructed state space informs us about states visited by the system, as well as their temporal relationships. How do we generate new trajectories (new state sequences) that have similar dynamics to the reconstructed state space trajectory? What other state sequences are implied by this trajectory? Given a new state $\mathbf{x}_n$ not found

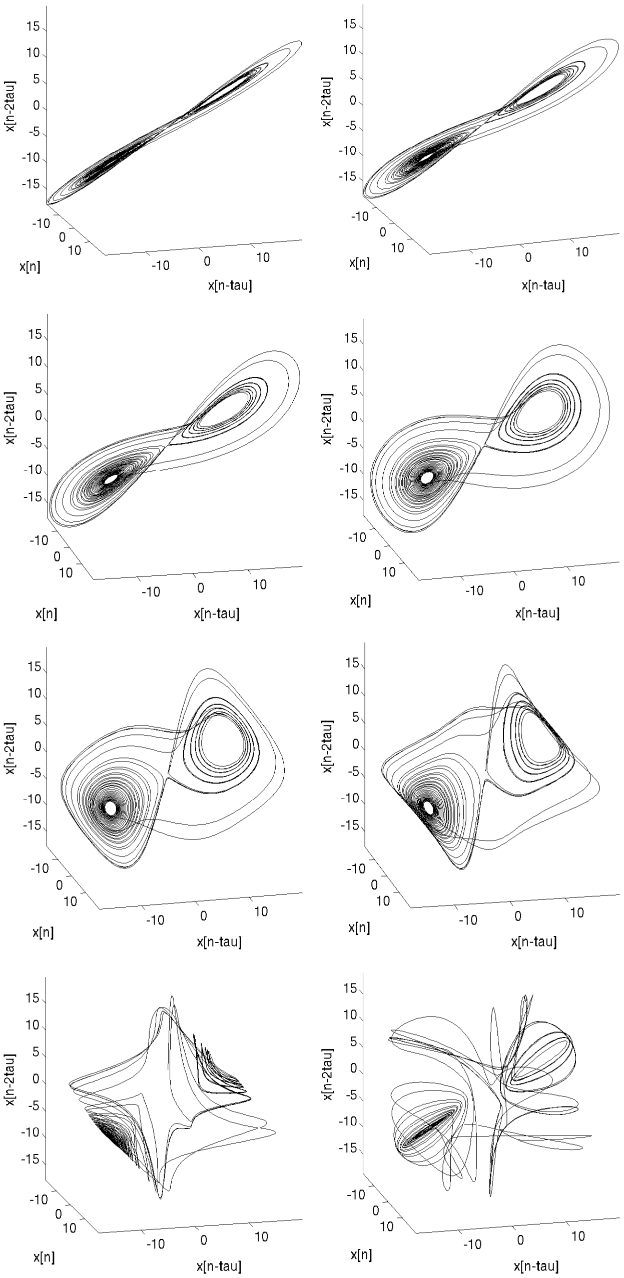


Figure 3-5: **Three-dimensional embedding of the observable $x[n]$ by the method of delays with different values of τ. From top to bottom, left to right: $\tau = 1, \tau = 2, \tau = 4, \tau = 8, \tau = 14, \tau = 20, \tau = 32, \tau = 64$.**

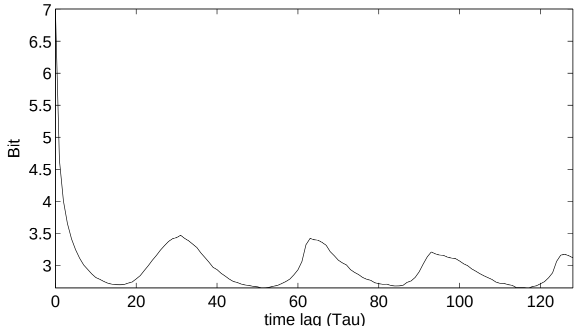


Figure 3-6: **Estimation of the value for τ by mutual information. The first local minimum is the best estimate of τ.**

in the original state space trajectory, what would be the "natural" or implied sequence of states $\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \mathbf{x}_{n+3} \ldots$? Paraphrasing in more musical terms, if our state space is a reconstruction of an actual piece of music, how do we generate new pieces of music that have similar motion characteristics to the original piece, yet different at the same time? What other musical sequences are implied by the structure of the embedding of the original piece?

These questions are intimately related to the problem of *prediction*, and in a reconstructed state space prediction becomes a problem of interpolation [39]. There are multiple ways to interpolate the space and different modeling techniques can be used. We can consider a single *global predictor* valid for all $\mathbf{x}_n$, or a series of *local predictors* which are valid only locally in the vicinity of the point of interest.

Local Linear Models

Probably the simplest interpolation method is the *method of analogues* proposed by Lorenz in 1969 [25]. Let $\mathbf{x}_{(i)n}$ be the ith nearest neighbor of $\mathbf{x}_n$. The *method of analogues* consists of finding the nearest neighbor $\mathbf{x}_{(1)n}$ to $\mathbf{x}_n$, and equating $\mathbf{x}_{n+1}$ to $\mathbf{x}_{(1)n+1}$. In other words, $F(\mathbf{x}_n) = F(\mathbf{x}_{(1)n})$.An obvious improvement over this method is to calculate $\mathbf{x}_{n+1}$ from a number of nearest neighbors. The number of neighbors can be defined by a radius ϵ around the point $\mathbf{x}_n$ (in which case this number would vary depending on the density of the state space around $\mathbf{x}_n$), or by choosing a fixed number on neighbors N. For both cases, the combination of the neighbor points of $\mathbf{x}_n$ can be weighted by their distances to $\mathbf{x}_n$. An estimation of $\mathbf{x}_{n+1}$ from the weighted average of the

N nearest neighbors of $\mathbf{x}_n$ can be expressed as

$$\hat{F}(\mathbf{x}_n) = \sum_{i=1}^{N} F(\mathbf{x}_{(i)n})\phi(\|\mathbf{x}_n - \mathbf{x}_{(i)n}\|) \tag{3.7}$$

where ϕ is a weighting function [25]. In our present implementation ϕ is defined as

$$\phi_i = \frac{\delta_i}{\sum_{j=1}^{N}\delta_j}, \quad \text{with} \quad \delta_i = \frac{\sum_{j=1}^{N}\|\mathbf{x}_n - \mathbf{x}_{(j)n}\|^p}{\|\mathbf{x}_n - \mathbf{x}_{(i)n}\|^p} \tag{3.8}$$

where p is a parameter used to vary the weight ratios of the states $\mathbf{x}_{(i)n}$ involved.

$$\hat{F}(\mathbf{x}_n) = \phi_1 F(\mathbf{x}_{(1)n}) + \phi_2 F(\mathbf{x}_{(2)n}) + \ldots + \phi_N F(\mathbf{x}_{(N)n})$$

$\phi(\cdot)$ returns values between 0 and 1, and $\phi_{(1)} + \phi_{(2)} + \ldots + \phi_{(N)} = 1$. In other words, only the relative distances between point x_n and its N nearest neighbors are considered, not their absolute distance. As a simple example of how this method performs, we calculate the *flow* of the entire state space by applying this formula to equally spaced points in the space. The two dimensional state space was constructed from a sinusoid $\sin(2\pi 40n)10 + 13$ using the method of delays, with $\tau = 6$. Figure 3-7 shows the original time series $s[n]$ and Figure 3-8 the result of the interpolation of the reconstructed state space. In this figure we see that all the points in state space converge to the original trajectory. In terms of the generation of new trajectories with similar characteristics to the original, this result is not useful. We want states in our reconstructed space to behave similarly to their nearest neighbors, not to be followed by the same states as their neighbors. Therefore, we modify our function

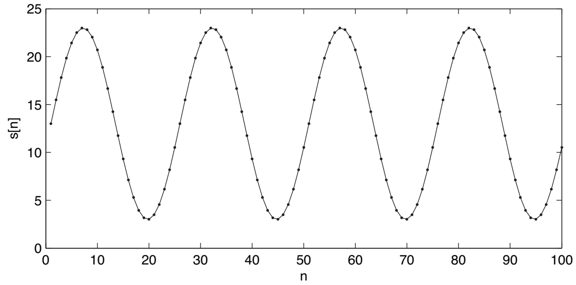


Figure 3-7: **Series $s[n] = \sin(2\pi 40n)10 + 13$ with 25 samples per cycle.**

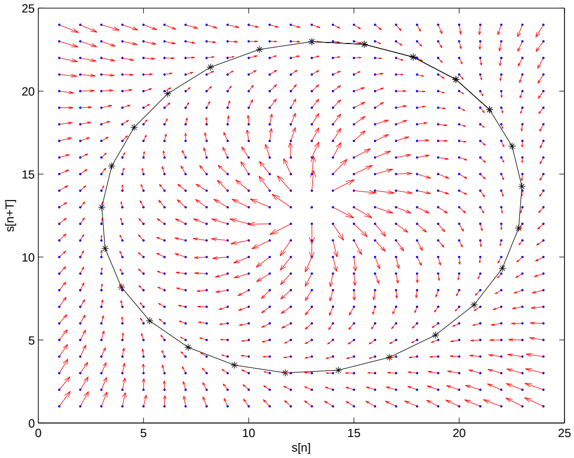


Figure 3-8: **Flow estimation of state space by averaging the predictions of the nearest neighbors. Parameters: $N = 3$, $p = 2$, $\tau = 6$.**

by replacing $F(\mathbf{x}_{(i)n})$ with $\dot{\mathbf{x}}_{(i)n} \equiv F(\mathbf{x}_{(i)n}) - \mathbf{x}_{(i)n}$. Thus we calculate the velocity of point $\mathbf{x}_n$ as the weighted average of the velocities of the N closest points to it.

$$\hat{F}(\mathbf{x}_n) = \mathbf{x}_n + \sum_{i=1}^{N} \dot{\mathbf{x}}_{(i)n} \phi(\|\mathbf{x}_n - \mathbf{x}_{(i)n}\|) \tag{3.9}$$

Figure 3-9 shows the estimated *flow* from this method.

When the observable signal $s[n]$ is by nature discontinuous (Figure 3-10), so will the reconstructed state space trajectory. In this case, using a large N will smooth out the interpolated space, distorting the characteristic angularity of the original trajectory (Figure 3-11). The interpolation function is an averaging filter smoothing the state space dynamics. It is basically a Moving Average (MA) filter with coefficients changing at each new estimation point. Obviously, using only one nearest neighbor will preserve the angularity of the trajectory since no filtering takes place (Figure 3-12). A nice middle point between a totally curved (smooth) space and a linear one is the use of a large number of nearest neighbors N together with an equally high p exponent. This results in a generally straight *flow* but with rounded corners (Figure 3-13).

The signals used in these examples are distant from actual music, but their simplicity makes the understanding and visualization of the concepts presented easier. In Chapter 5 we will use real music; for now just

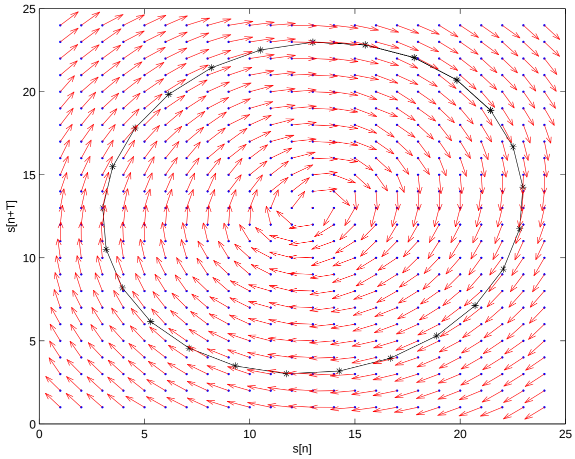


Figure 3-9: **Flow estimation for the state space by weighted average of nearest neighbor velocities. Parameters:** $N = 3$, $p = 2$, $\tau = 6$.

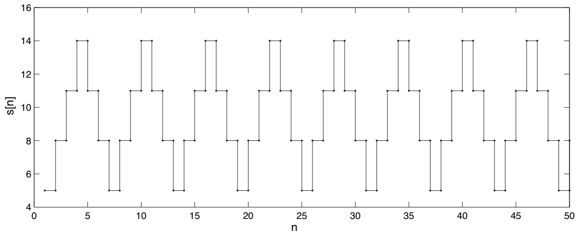


Figure 3-10: **Signal** $s[n] = 5, 8, 11, 14, 11, 8, 5, 8, 11, 14, 11, 8, 5 \ldots$

keep in mind that the observation signal $\mathbf{s}[n]$ can be any musical component such as pitch, loudness, IOI, etc., or, as discussed in the previous chapter, even multidimensional signals composed of all these parameters simultaneously.

We have focused on a particular implementation of local linear models as a way of generalizing the reconstructed state space dynamics. We have also discussed the effect different parameter values have on the state space interpolation. The implications these have on reconstructed spaces from actual music will be discussed in Chapter 5.

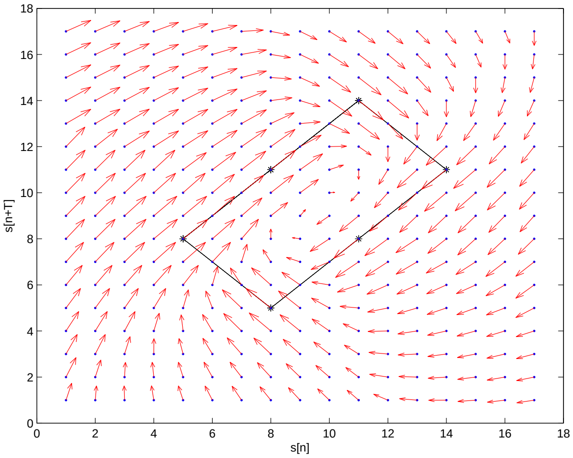


Figure 3-11: **Flow estimation of state space by nearest neighbor velocities. Parameters :** $N = 8$, $p = 2$, $\tau = 1$.

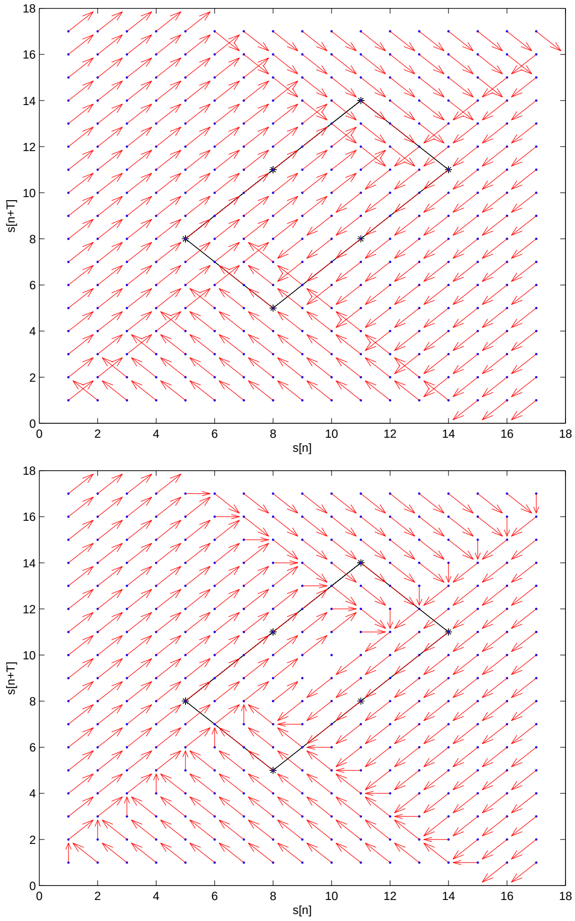


Figure 3-12: **Flow estimation of state space by nearest neighbor velocities.** ***Top:*** **Parameters :** $N = 1$, $\tau = 1$.
Bottom: **Parameters:** $N = 2$, $p = 2$, $\tau = 1$.

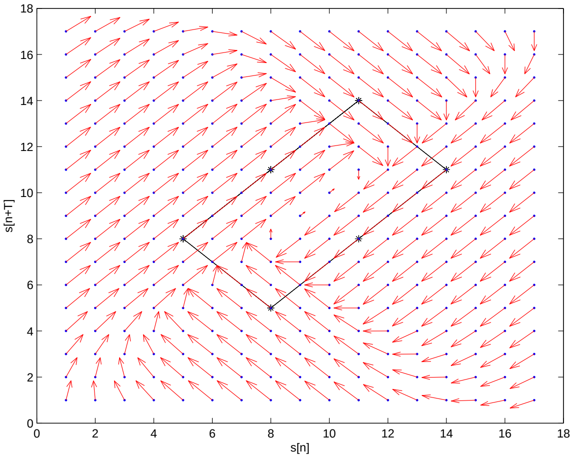


Figure 3-13: **Flow estimation of state space by nearest neighbor velocities. Parameters :** $N = 8$, $p = 10$, $\tau = 1$.

CHAPTER FOUR
Hierarchical Signal Decomposition

4.1 Divide and Conquer

Signal analysis can be characterized as the science (and art) of decomposition: how to represent complex signals as a combination of multiple simple signals. From a musicological perspective, it can be thought of as reverse engineering composition. The power and usefulness of having a signal represented as a combination of simpler entities is easy to see. Once a decomposition is achieved, we can modify each component independently and recompose novel pieces that may share similar characteristics with the original. Just as with modeling, the best decomposition depends on its purpose. For the purpose of musical analysis, we would like the decomposition to be informative of the signal at hand and for it to be perceptually meaningful. For the purpose of re-composition, we would like the decomposition to allow for flexible and novel transformations.

How can we decompose our signal in a meaningful way? Different characteristics of a signal may suggest different decomposition procedures. Frequently, a signal can be separated into a deterministic and a stochastic component. More specific components might be a *trend*, which is the long term evolution of the signal, or a *seasonal*: the periodic long term oscillation[6]. Thus, multiple hierarchical overlapping structures may occur simultaneously. Particularly within the seven orders of magnitude spanned by music (see Chapter 2), these structures can vary tremendously from one time scale to the next.

Because of the relevance of hierarchic structure to music and music perception, this chapter will focus on decomposition methods that reveal structure at different time scales. Before discussing these methods, we give a brief summary of some preliminary developments.

4.1.1 Fourier Transform

By far the most popular representation of a signal as a combination of simple components is the Fourier transform. This transform consists of representing a signal as a combination of sinusoids of different frequency, amplitude and phase. While it is a powerful linear transformation, it's not without its limitations. Because the basis functions of this transformation are the complex exponentials ($e^{i\omega t}$) defined for $-\infty < t < \infty$, the transform results in a representation of the relative correlation of each of the sinusoids over the whole signal, with no information about how they might be distributed over time. This is fine for stationary signals, but music is almost never stationary. Thus, the Fourier transform is not an optimal representation of music.

4.1.2 Short Time Fourier Transform (STFT)

In 1946 Gabor proposed an alternative representation by defining elementary time-frequency atoms. In essence, his idea was to localize the sinusoidal functions in order to preserve temporal information while still obtaining the frequency representation offered by the Fourier transform. This localization in time is accomplished by applying a windowing function to the complex exponential

$$g_{u,\xi(t)} = g(t-u)e^{i\xi t}. \tag{4.1}$$

Gabor's *windowed Fouried transform* then becomes

$$Sf(u,\xi) = \int_{-\infty}^{+\infty} f(t)g(t-u)e^{-i\xi t}dt. \tag{4.2}$$

The energy spread of the atom $g_{u,\xi}$ can be represented by the Heisenberg rectangle in the time-frequency plane (Figure 4-1). $g_{u,\xi}$ has time width σ_t and frequency witdh σ_ξ, which are the corresponding standard deviations [27]. While one would like the area of the atoms to be arbitrarily small in order to attain the highest possible time-frequency resolution, the uncertainty principle puts a limit to the minimum area of the atoms at

$$\sigma_t \sigma_\omega \geq \frac{1}{2}.$$

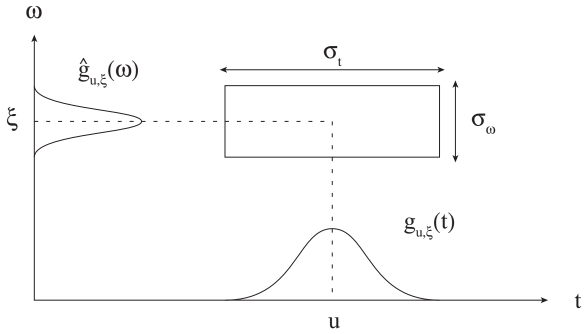


Figure 4-1: **A Heisenberg rectangle representing the spread of a Gabor atom in the time-frequency plane.**

4.1.3 Wavelet Transform

While the STFT offers a solution to the problem of decomposing non-stationary signals, it still isn't the most optimal representation of our data for the following reasons:

1. Many musical high level signals are discontinuous by nature. The choice of a continuous basis is not necessarily the optimal choice.
2. At high levels of musical structure, the concept of frequency has little or no meaning.
3. The STFT does not give us information about the structure of a signal at different time scales.

The development of a method for multi-resolution analysis is necessary when dealing with perceptually relevant data. Our hearing apparatus has evolved to extract multi-scale time structures from sound. Inspired by the function of the cochlea, Vercoe implemented multi-resolution sound analysis in Csound [42] by exponentially spacing Fourier matchings. In the early 1900s, Schenker proposed what is arguably his main contribution to music analysis: the abstraction of musical structures from different *Schichten*, or layers at multiple scales [31]. Yet, Schenker's analysis defines the multi-scale structures in terms of tonal harmonic functions and certain specific voice leadings. Here we would like to extract the multi-scale representation more generally in terms of motion and, in addition, automatically. Thus, we import the general purpose tools traditionally used in the micro-world of sound to the macro-world of form.

As a natural extension of the STFT, the wavelet transform was developed as a way to obtain a more useful multi-resolution representation. Like the STFT, the wavelet transform correlates a time-localized *wavelet* function $\psi(t)$ with a signal $f(t)$ at different points in time, but in addition the correlation is measured for different scales of ψ to achieve a multi-scale representation. Thus, the wavelet transform of $f(t)$ at a scale s and position u is given by

$$Wf(u,s) = \int_{-\infty}^{+\infty} f(t)\psi^* \left(\frac{t-u}{s}\right) dt. \tag{4.3}$$

Wavelets are equivalent to hierarchical low-pass and high-pass filterbanks called *quadrature mirror filters* [18]. A signal is passed through both filters. Then the output of the low-pass filter is again passed though another pair of low-pass and high-pass filters and so on recursively (Figure 4-2). The outputs of the high-pass filters are called the *details* of the signal and the outputs of the low-pass filters are called the *approximations*.[1] The number of filter pairs used in this recursive process defines the number of scales in which the signal is represented.

While the "Gabor atoms" of a STFT are windowed complex exponentials, the development of the wavelet transform has introduced a wide variety of alternative basis functions. The first mention of wavelets is found in a thesis by Haar (1909)[20]. The *Haar* wavelet is a simple piecewise function

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \le t < 1/2 \\ -1 & \text{if } 1/2 \le t < 1 \\ 0 & \text{otherwise} \end{cases}$$

that when scaled and translated generates an orthonormal basis:

$$\left\{ \psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi \left(\frac{t - 2^j n}{2^j} \right) \right\}_{(j,n)\in\mathbb{Z}^2}$$

[1] For a detailed discussion about the relationship between *quadrature mirror filters* and wavelets, see [27].

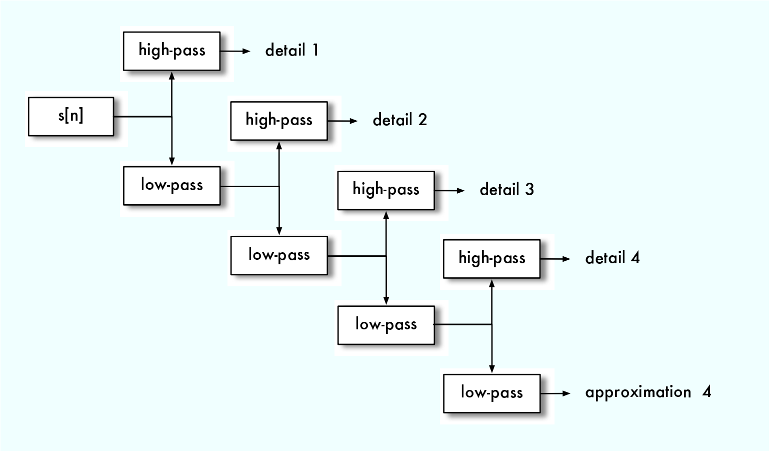


Figure 4-2: **Recursive filtering of signal $s[n]$ through pairs of high-pass and low-pass quadrature mirror filters.**

Because of the discontinuous character of the Haar wavelet, smooth functions are not well approximated. Its limitation started the investigation of alternative wavelets, and many varieties with different properties have been invented since. Probably the most popular are the *Daubechies* family of wavelets. The *Daubechies 1* wavelet basis is the same as Haar's, and the family becomes progressively smoother as its number increases.

How then do we choose a wavelet type? What is the best set of wavelet basis? Evidently, the definition of "best" depends on our goal. If our goal were compact representation, then we would want a basis that used the least number of wavelets without losing much information. The usefulness of this is obvious for compression and noise reduction, but the choice of a wavelet basis that provides the most economical representation may also help us understand the nature and intrinsic properties of a given signal. A common measure of how well a limited set of wavelets describes a given signal is by a linear approximation. Given a wavelet basis $\mathcal{B} = \{\phi_n\}$, a linear approximation s_M of a signal s is given by the M larger scale wavelets [27]:

$$s_M = \sum_{n=0}^{M-1} \langle s, \phi_n \rangle \phi_n \tag{4.4}$$

The accuracy of the approximation is typically measured as the squared norm of the difference between the original signal and the approximation:

$$\epsilon[M] = \|s - s_M\|^2 = \sum_{n=M}^{+\infty} |\langle s, \phi_n \rangle|^2 \tag{4.5}$$

Figures 4-3 and 4-4 show wavelet decompositions of Bach's Prelude from Cello Suite no. 1. Both show the largest approximation and the details at all four levels. For Figure 4-3 we used *Daubechies 1* wavelet, and *Daubechies 8* for Figure 4-4. Measuring the approximation for $s_M = a_4$ in both cases with equation 4.5, we get $\epsilon = 117.2$ with *Daubechies 8* and $\epsilon = 122.69$ with *Daubechies 1.* Thus, Bach's prelude is better approximated with the *Daubechies 8* wavelet. Notice though that the decompositions of the signal using *Daubechies 8* are always smooth, while those with *Daubechies 1* are not. If the discontinuous character of the prelude is important to preserve, then *Daubechies 1* is a better choice of wavelet. Because our goal is to generate new pieces and not to compress them, this qualitative criterion is certainly more useful to us. In the next chapter we will exploit the continuous quality of *Daubechies 8* to obtain musical variations with similarly smooth characteristics.

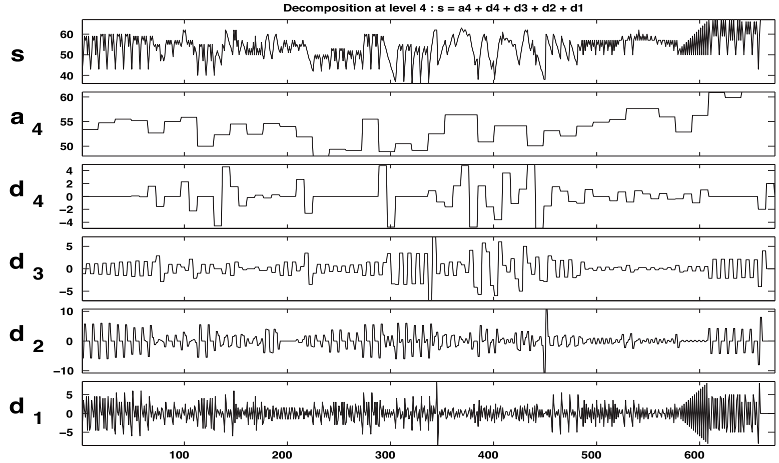


Figure 4-3: **Pitch sequence of Bach's Prelude from Cello Suite no. 1 and its wavelet decomposition using *Daubechies 1* wavelets. s is the original pitch sequence, a_4 the approximation at level 4 and d_n are the details at level n. The sequence is sampled every sixteenth note.**

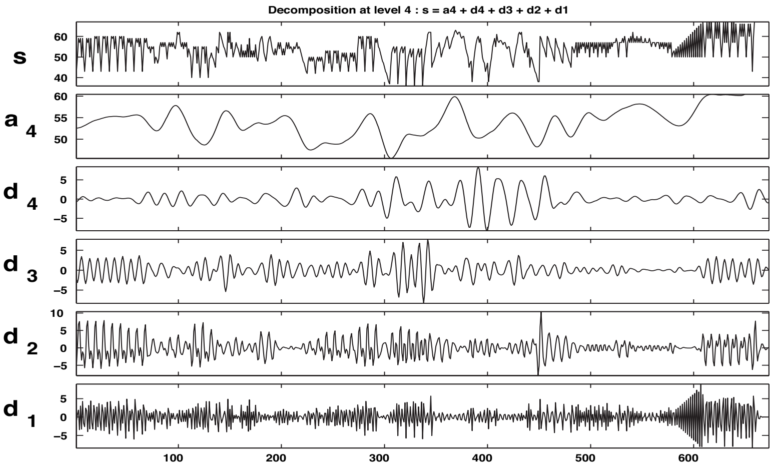


Figure 4-4: **Pitch sequence of Bach's Prelude from Cello Suite no. 1 and its wavelet decomposition using *Daubechies 8* wavelets. s is the original pitch sequence, a_4 the approximation at level 4 and d_n are the details at level n. The sequence is sampled every sixteenth note.**

4.1.4 Data Specific Basis Functions

As we saw in Chapter 3, state space reconstruction by the method of delays allows us to uniquely represent each step in a given sequence as a function of multiple variables: $s_n = h(\mathbf{x}_n) = h(x_1, x_2, \ldots, x_n)$. $h(\mathbf{x}_n)$ must then be defined as a combination of the components of $\mathbf{x}_n$, for example $s_n = a_1(x_n) + a_2(x_n) + \ldots + a_n(x_n)$ or $a_1(x_n)a_2(x_n)\ldots a_n(x_n)$ or may have one of many other forms. Can the structure of the manifold that results from the embedding tell us something about the structure of $h(\mathbf{x}_n)$ and its variables, and as a consequence on the observation signal s_n? It turns out that similarly to the wavelet transform, a lag space representation automatically recovers structure at different time scales. This is one of the most important observations regarding state space reconstruction by the method of delays. As an example, consider the following signal: $\sin(2\pi 10t) + \sin(2\pi 47t)/4$ (Figure 4-5). This signal can be seen as a fast oscillation "riding" on a slow oscillation. In a three-dimensional lag space (Figure 4-6), this simple linear combination of two sinusoids becomes a torus, and the two levels of motion (a global cycle and a local cycle) are clearly distinguishable. In Chapter 3 we assumed all along that the systems we were dealing with were autonomous, yet a state space trajectory such as this one can be interpreted as an autonomous dynamical system being driven by another system. How do we separate these two systems? Because the state space is reconstructed by assigning

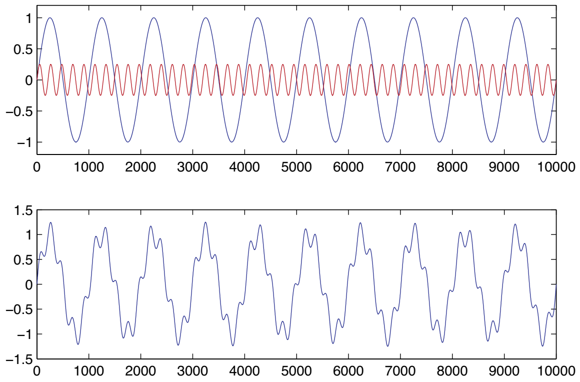


Figure 4-5: **Top:** $\sin(2\pi 10t)$ **and** $\sin(2\pi 47t)/4$. **Bottom:** $\sin(2\pi 10t) + \sin(2\pi 47t)/4$.

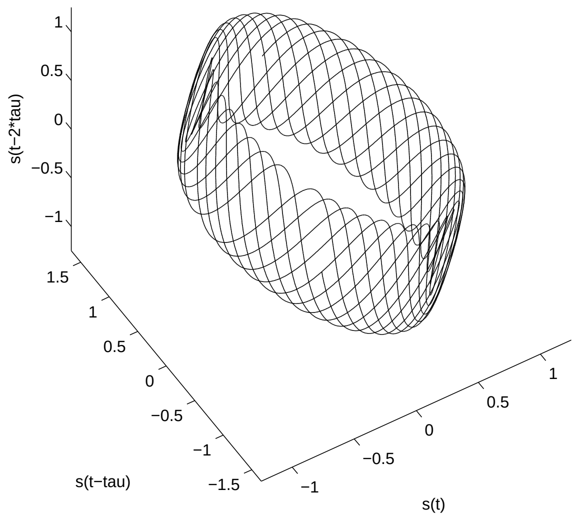


Figure 4-6: **Three dimensional state space reconstruction of** $\sin(2\pi 10t)+\sin(2\pi 47t)/4$**, with** $\tau = 220$**.**

delayed versions of a single signal to each coordinate, they all have the same information delayed by some τ time. In other words, projecting the embedding onto any of the three dimensions will yield the same signal. A careful observation of the torus in three-dimensions reveals that by rotating the structure in such a way that the maximal variances are aligned with the axes of the space, the previously redundant dimensions become decorrelated, separating the two sinusoids almost completely (Figure 4-7). Figure 4-8 plots the two signals resulting from the projection on two of the three orthogonal axes after rotating the torus. The method of

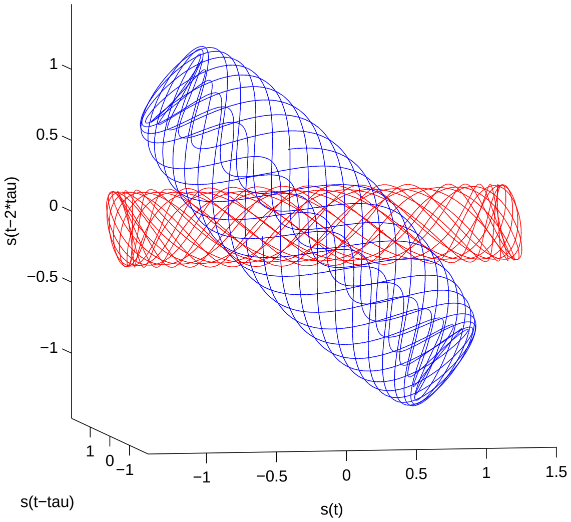


Figure 4-7: **PCA on the torus resulting from the state space reconstruction of** $\sin(2\pi 10t) + \sin(2\pi 47t)/4$**, with** $\tau = 220$**.**

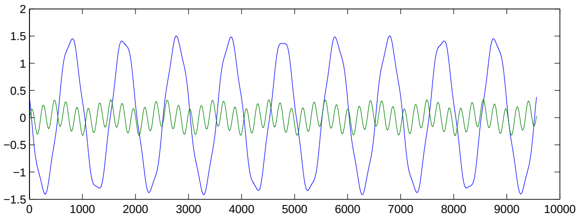


Figure 4-8: **Two component dimensions of the state space reconstruction of** $\sin(2\pi 10t) + \sin(2\pi 47t)/4$ **(with** $\tau = 220$**) after PCA transformation.**

delays therefore provides a means for decomposing our signal into a "natural" set of hierarchical basis functions. In other words, the functions are derived from the data rather than selected *a priori*. The transformation that rotates the state space trajectory so that dimensions become decorrelated is called Principal Component Analysis. In Chapter 3 we saw that the interpolation of the state space manifold "sketched" by the reconstructed trajectory is a way of generalizing the dynamics of a signal. These recovered basis functions are the other half of the generalization of the structure. These building blocks are independent entities that can be modified and recombined to generate new state space geometries while preserving its identity (its "torusness"). For example, multiplying the amplitude of the first sinusoid by 5 results in the state space trajectory shown in Figure 4-9. This is a simple example by which we illustrate concepts and tools for music structure generalization. In the next chapter we will use them to generate new pieces of music.

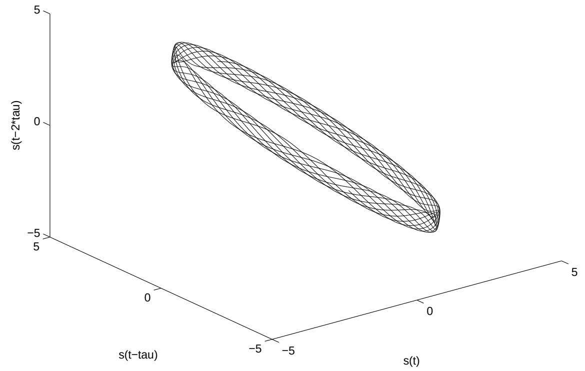


Figure 4-9: **Three dimensional state space reconstruction of** $\sin(2\pi 10t)5 + \sin(2\pi 47t)/4$**, with** $\tau = 220$**.**

For completeness, we compare PCA with the wavelet transform using again Bach's Prelude. We arbitrarily choose eight dimensions for the embedding of the Prelude, and $\tau = 1$. Figure 4-10 shows the bases resulting from applying PCA to the lag space and projecting the trajectory on each of the dimensions.

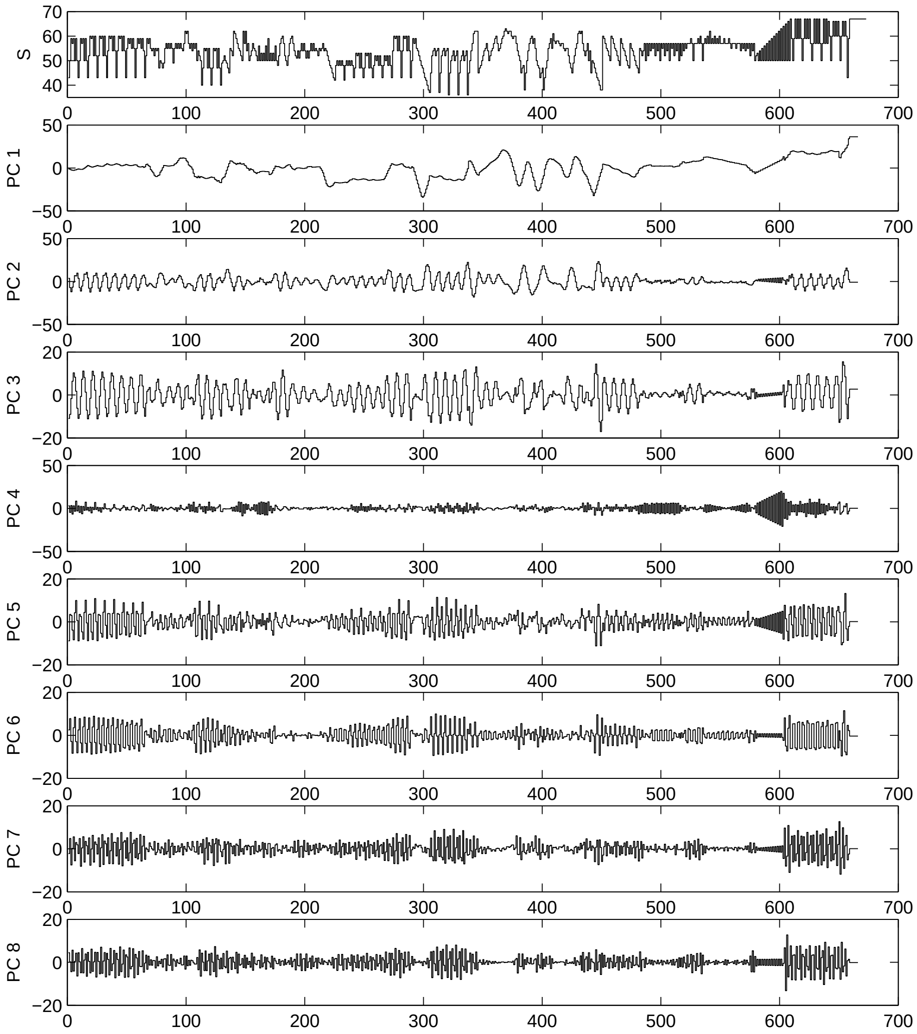


Figure 4-10: **Pitch sequence of Bach's Prelude from Cello Suite no.1 and each of the eight principal component extracted from an eight dimensional lag space.** s **is the original pitch sequence,** PCn **the** n**th principal component. The sequence is sampled every sixteenth note.**

4.1.5 Principal Component Analysis (PCA)

The main purpose of Principal Component Analysis is that of decorrelating a set of correlated variables. As already suggested, the variables we are interested in decorrelating are the dimensions of a lag space. We want each of the dimensions to become as independent as possible so that each provides unique information about the state space trajectory.

We have seen that a careful rotation of the lag space can separate the different scale dynamics of a signal $s[n]$. But how do we do this mathematically? Essentially what we want is that the m dimensions that make up the lag space trajectory be as independent and as informative as possible. PCA is the process of obtaining *linear* independence between the m vectors $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m$, i.e. if $a_1\mathbf{z}_1 + a_2\mathbf{z}_2 + \ldots + a_m\mathbf{z}_m = 0$ then $a_i = 0$ for all i. Statistically, m variables are linearly independent if their covariances equal zero. Given that the m length observation vectors $\mathbf{x}$ are vertical, their covariance matrix is defined as:

$$C_{\mathbf{x}} \equiv E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T]$$

We want a transformation $\mathbf{y} = \mathbf{M}\mathbf{x}$ such that $\mathbf{C}_{\mathbf{y}}$ becomes the identity matrix.

In relation to $\mathbf{x}$, the covariance matrix of $\mathbf{y}$ is

$$\begin{aligned}
\mathbf{C}_{\mathbf{y}} &= E[(\mathbf{y} - E[\mathbf{y}])(\mathbf{y} - E[\mathbf{y}])^T] \\
&= E[(\mathbf{M}(\mathbf{x} - E[\mathbf{x}]))(\mathbf{M}(\mathbf{x} - E[\mathbf{x}]))^T] \\
&= E[(\mathbf{M}(\mathbf{x} - E[\mathbf{x}]))((\mathbf{x} - E[\mathbf{x}])^T\mathbf{M}^T) \\
&= \mathbf{M}E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T]\mathbf{M}^T \\
&= \mathbf{M}\mathbf{C}_{\mathbf{x}}\mathbf{M}^T
\end{aligned}$$

Choosing $\mathbf{M}$ to be the eigenvectors of $\mathbf{C}_{\mathbf{x}}$ is the transformation that makes $\mathbf{C}_{\mathbf{y}}$ the identity matrix, rendering the dimensions of the lag space linearly independent.

Yet, it is important to be aware of the assumptions and limitations of this trasformation[36].

1. *Linearity.* The state space manifold resulting from a time lag embedding may (and many times is) curved and twisted. In order to avoid redundancy, a nonlinear transformation would be required prior to applying PCA . In the sum of sinusoids example, the com-

ponents were by definition linearly independent, which allowed us to separate them well. Unfortunately, this is rarely the case.

2. *The principal components are orthogonal.* Related to the problem of linearity, it is possible too that the points in lag space result in distributions whose axis are not perpendicular to each other. Thus, while PCA may decorrelate some axes, it may not decorrelate others.

3. *Large variances reflect important dynamics.* For the purpose of dimensionality reduction, PCA assumes that the important information is in the parameters (or dimensions) with the greatest variance, while those with the least variance are discarded as insignificant or noise. In the Bach example, the large scale dynamics are the ones with the greatest variance, but this is not necessarily always the case (although it usually is).

While PCA has its own limitations, it is still a useful transformation that can allow us to extract and modify dynamics at different time scales. Other more robust transformations attempt to achieve statistical independence between all the variables ($p(x_i, x_j) = p(x_i)p(x_j)$), rather than just linear independence. The approaches that fall into this category are called ICA: Independent Component Analysis. We will not deal with this family of transformations here.

4.2 Summary

The possibility of representing a signal as a combination of simpler basis functions is a second level of generalization. This decomposition allows us to transform the state space trajectory with much more flexibility and, by implication, the original observable sequence as well.

Applying PCA on the lag space results in a set of useful basis functions that reveal hierarchical structures in a similar way the wavelet transform does. The simplicity of PCA over that of wavelets is another attractive feature. On the other hand, having an analytical description of the wavelet basis can give us more control over the type of decompositions and transformations we may perform. In the following chapter we will apply both transformations to concrete musical examples.

CHAPTER FIVE
Musical Applications

We have presented some theoretical background, tools and methods useful for the analysis, modeling and resynthesis of music, and are ready to give them some concrete applications. We have also mentioned how each of the methods provides some aspect of the generalization necessary for an inductive model. We now summarize the three types of generalization discussed:

1. **State space interpolation** (Chapter 3): The recovered state space trajectory by the method of delays serves as a kind of "wireframe" that suggests the existence a full hyper-surface or manifold that can be estimated by interpolation. New trajectories that fall on the manifold and follow its *flow* constitute the generalization. By following the flow, new pieces with similar dynamics and states can be generated.
2. **Abstraction: states vs. dynamics** (Chapter 3): The recovered state space trajectory provides information about the states visited by the trajectory and the dynamic relationships between these states. Each of these two pieces of information can be separated and varied independently for the generation of new music.
3. **Decomposition (wavelets, PCA, ICA)** (Chapter 4): Either through wavelets or through PCA or ICA, it is possible to decompose a musical sequence into hierarchical, simpler, and meaningful components. These decompositions reveal embedded dynamics and provides an additional level of understanding of musical structure and flexibility for subsequent transformations.

All three aspects can be used as tools in music analysis and synthesis, and may be combined in many different ways. Here, we can only give a few examples of their use and suggest some other possibilities. Sound files from the examples discussed in this chapter as well as additional relevant material can be found at `http://www.media.mit.edu/~vadan/msthesis`.

5.1 Music Structure Modeling and Resynthesis

Combined Components vs. Independent Components in Multidimensional Signals

When considering the multiple components of a piece of music, we can opt to model their dynamics independently or collectively. Keeping things separate gives us more control over the variety of transformations we can apply to music, but modeling all components collectively allows us to generate new trajectories that preserve their aggregate dynamics in the original training piece. The state space reconstruction of these two approaches is depicted in Figure 5-1.

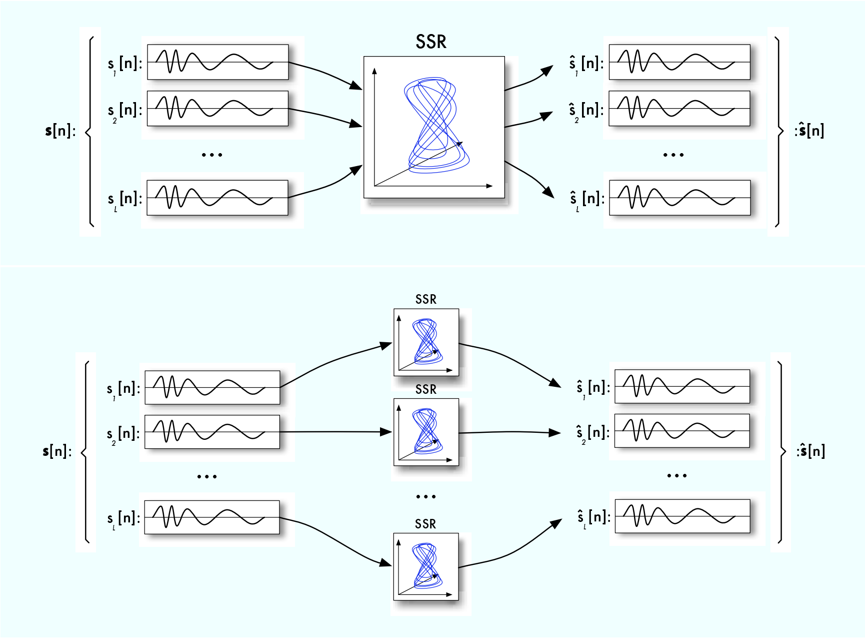


Figure 5-1: ***Top*: The aggregate dynamics of all the musical components such as pitch, loudness and IOI are modeled jointly by constructing a single state space. *Bottom:* Multiple state spaces are reconstructed and modeled from each component independently.**

For the purpose of state space reconstruction, a multidimensional signal

$$\mathbf{s}[n] = \begin{bmatrix} s_{1,0}, & s_{1,1}, & \ldots, & s_{1,n} \\ s_{2,0}, & s_{2,1}, & \ldots, & s_{2,n} \\ \vdots & & & \\ s_{l,0}, & s_{l,1}, & \ldots, & s_{l,n} \end{bmatrix} \tag{5.1}$$

can be treated in essentially the same way as a scalar signal. Each point in state space is defined as a set of delays of each of the signal's components:

$$\begin{aligned} \mathbf{x}_n \;\; = \;\; & (s_{1,n}, s_{1,n-\tau}, \ldots, s_{1,n-(m-1)\tau}, s_{2,n}, s_{2,n-\tau}, \ldots, \\ & s_{2,n-(m-1)\tau}, \ldots, s_{l,n}, s_{l,n-\tau}, \ldots, s_{l,n-(m-1)\tau}). \end{aligned} \tag{5.2}$$

The reconstructed state space has dimension lm. As can be seen in the previous equation, the resulting state space is an aggregate of the individual state spaces of each component in the signal $\mathbf{s}[n]$. Thus, a different τ can be chosen for each component $s_i[n]$. Since the value ranges of the components $s_i[n]$ will usually differ widely, it is important to *whiten* the reconstructed state space to avoid having dimensions that are too flat relative to others. Having different variances for the different dimensions will distort the map or flow estimation since our weighting and averaging functions rely on euclidian distances. After estimating a new trajectory, we must de-whiten the data.

5.1.1 Interpolation

As we reviewed in Chapter 3, the type of interpolation we perform on the reconstructed state space can greatly affect the qualitative characteristics of the estimated flow. Thus, a careful choice of the state space model parameters can be crucial to obtain good musical output. For the case of the local linear models discussed in Chapter 3, there are two parameters to consider: the number of nearest neighbors and the p exponent in Equation 3.8 (i.e. the weight of each neighbor as a function of their distance). In addition to whatever modeling technique we use, there is the choice of embedding dimension and the initial state $\mathbf{x}_0$ for the estimation of a new trajectory.

As a concrete example of how the different parameters affect the interpolation and thus the outcome of new musical pieces we use Ligeti's Piano Etude no.4 Book 1 (Figure B-1). We combine pitch, loudness, IOI and duration in a single lag space and estimate a new trajectory for multiple combinations of the parameters already discussed. For each new trajectory estimated we systematically vary one parameter value at a time. The embedding dimensions explored are 2, 4, 6, 8 per parameter. So, given that we are embedding four parameters together, the lag spaces actually have 8, 16, 24, and 32 dimensions. The number of nearest neighbors used are 2, 4, 6, and 8, and values of the exponent p are 2, 16, 64, and 128. As the initial condition we always use the mean of the space. Figure B.3 in Appendix B shows the new generated sequences in pianoroll representation. All sequences are 500 notes long.

Number of Neighbors and the p Exponent From the pianoroll plots of the generated pieces we see that for very high values of p, particularly $p = 128$, the resulting sequences are almost identical to some fragment of the original Etude. This is because for such high values of p the nearest neighbor to the point being estimated will have much greater weight in defining its behavior than the more distant points. Thus, no matter how many neighbor points are considered in the estimation, having $p = 128$ is practically the same as considering only one nearest neighbor. The resulting trajectories are then transposed copies of the original trajectory. The other problem with using only one neighbor or too high a value for p is that, particularly for embeddings with insufficient dimensions, it is very likely that the estimated trajectory will fall in an infinite loop. In Figure B.3 we see several instances of this. The second half of panels two and three (from left to right, top to bottom), as well as panel eight are examples of this. Ideally we want to use several neighbors for the interpolation to be more accurate. But, as discussed in Chapter 3, the use of more than one nearest neighbor results in smoothed out maps, distorting the originally angular quality of discontinuous dynamics. Thus, a careful balance between the number of nearest neighbors and the values of the exponent p must be found in order to produce novel yet sharp interpolations. Naturally, if the original series where continuous, then the choice of number of neighbors would not be a problem.

Dimensionality of the Embedding In Chapter 3, we discussed the requirement of finding a high enough dimension for the embedding to be able to capture all the degrees of freedom of a continuous autonomous system, and for a bijective relation to exist between the observation sequence s and the state space trajectory $\mathbf{x}$. What are the implications of the embedding by the method of delays on discontinuous series? Mainly that trajectories may cross without violating the bijective relation between the state space trajectory and the observation sequence because trajectory crossing of discontinuous series does not imply an overlap of points. Thus, the embedding dimension for discontinuous sequences can be smaller and still be good for the generation of new sequences. From the plots in Figure B.3 we can see that, in general, large scale dynamics from low dimensional embeddings (2 or 4) are rather chaotic compared to those of high dimensional ones.[1] To understand why this is the case, consider the nature of the reconstruction by the method of delays. Each state in lag space is defined as a set of ordered values taken from the observation sequence. If the embedding dimension $m = 2$, we define each state as a set of two values in the observation. Setting $\tau = 1$, for example, defines each state as two successive values of the observation sequence. Choosing a four dimensional embedding would give us a map of all the four value combinations found in the sequence. Evidently, one state might be visited more than once if the dimensionality is not high enough. It should be clear that the higher the dimensionality of the space, the sparser it will be. If the dimension is too high, very few points will occupy the space, and different trajectory segments will be at a big distance from each other, making their interaction for the generation of new trajectories more difficult. If the dimensionality is small, the space will be very dense and points may fall on top of each other. Here, new trajectories will be estimated from several trajectory segments that are close to each other in the embedding, but not necessarily close temporally in the original sequence. Thus, these points may have very different velocities.

Initial State This is the most difficult parameter to estimate in terms of the predictability of the output. Choosing any of the points that fall in the embedded trajectory as initial state will result in the original music sequence from that point on. The actual estimation of new states begins when the end of the trajectory is reached. In time series prediction this is the natural point to begin the estimation of a new trajectory. But for the purpose of generating novel trajectories without forecasting motivations, any point in the state space that does not fall on the original trajectory will do. A reasonable thing to do might be to select the initial

[1] This is similar to what happens with Markov models, where small order models result in sequences that are locally similar to the original but globally unrelated, while higher order models yield sequences that preserve the structures of the original piece at higher scales.

state at random, but for the purpose of evaluating parameter values and dimension sizes, it is best to choose a constant initial state.

There is only so much we can do by modeling all the components together. The lack of control on the independent parameters and the limited number of variables in the local linear models used make it difficult to produce novel pieces that are clearly different from the training piece. Because the training piece in its entirety is modified by only a few parameters, the new generated pieces are coarse variations on the original. In order to produce more refined transformations, we can make use of the decomposition methods discussed in the previous chapter.

5.1.2 Abstracting Dynamics from States

Rotations

A geometric interpretation of the reconstructed state space suggests a variety of transformations of the musical data. Simple transformations like rotations have been used as generators of musical variation [43, 16, 1]. In fact, the classical transformations of inversion, retrograde and retrograde inversion found in western music since the 16th century can be understood as four simple space-time transformations: 180° rotations in two-dimensional lag space and temporal reversing.

In Chapter 4 we reviewed how PCA on a reconstructed state space is nothing more than a special purpose rotation for decorrelating the component dimensions. But more generally, rotations can be interpreted as a way of changing the states visited by a dynamical system while keeping the dynamics unchanged. Thus, a state space trajectory defines a set of possible state sequences that share the same dynamics.

If the purpose of the embedding is the application of geometric transformations exclusively, then the requirements for a proper embedding for state space reconstruction (see Chapter 3) can be ignored, and all dimensions $m \geq 2$ are useful. In this interpretation the dimension of the embedding defines the possible degrees of transformation. The number of degrees of freedom of a rotation is defined by the dimensionality of the space. For an m-dimensional space, there are $C\binom{m}{2}$ planes of rotation [2]. Thus, higher dimensional spaces have the potential for a greater variety of transformations.

A rotation is a linear combination of the m elements defining each point in lag space. For example, given a series $s[n]$, we embed it by the *method of delays* with a delay of τ and dimension $m = 3$. We get the embedded trajectory $\mathbf{x}[n]$, so that each point in the m-dimensional space is $\mathbf{x}_n = (s_n, s_{n-\tau}, s_{n-2\tau})$. If we apply a rotation of angle θ to the $s_{n-2\tau}$ axis of the embedding space, we get:

$$\begin{aligned} \hat{\mathbf{x}}_n &= \begin{bmatrix} s_n & s_{n-\tau} & s_{n-2\tau} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= (\cos\theta s_n - \sin\theta s_{n-\tau}, \sin\theta s_n + \cos\theta s_{n-\tau}, s_{n-2\tau}) \end{aligned} \tag{5.3}$$

These rotations can be applied to the lag space of any musical component independently or to that of the combined components. In this case the components will be combined, so that pitch, for example, will be a combination of pitch, rhythm, etc. As an example of rotating the lag space of a single component we take the first six measures of Bach's Courante from Cello suite no.4. We consider only one component: the Inter Onset Interval (which in this case is the same as the duration of each note).[2] We embed the sequence in three-dimensional lag space (Figure 5-3). We explore all 64 combinations of 90° rotations in this three-dimensional space and plot the projections of each rotation. As can be seen in Figure 5-4, many resulting patterns are identical. Only six patterns are actually different (Figure 5-5). Thus, with 90° rotations we obtain only five new sequences out of a total of 64 rotations. We are curious to know what other patterns might be obtained from other rotations, so we compare the projections of all 512 45° rotations. Out

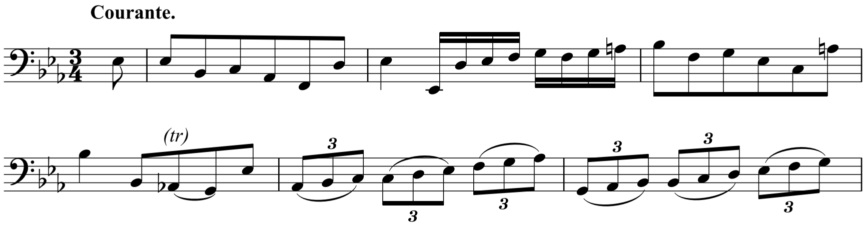


Figure 5-2: **First six measures of Bach's Courante from Cello Suite no.4.**

[2] While the trill "embellishment" in measure 4 is not written out explicitly, it is an important element of the rhythmic sequence. Thus, in the electronic score we write it out as 32nd notes.

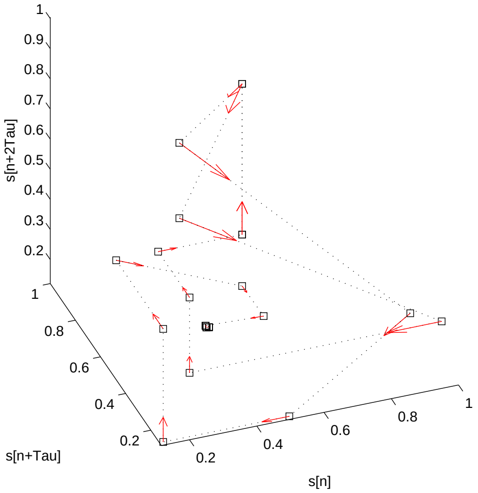


Figure 5-3: **Three-dimensional embedding of IOI of the first six measures of Bach's Courante from Cello Suite no.4.**

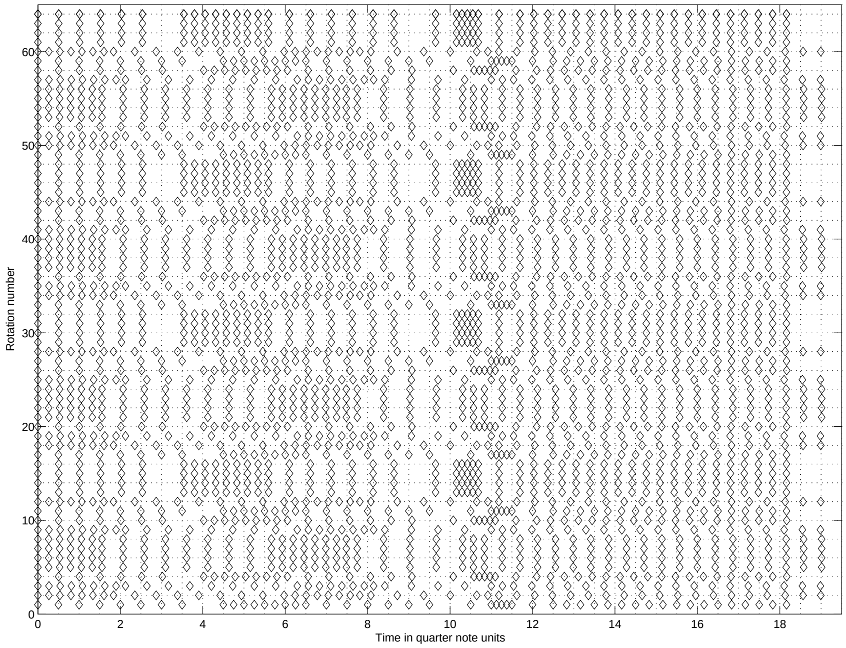


Figure 5-4: **Projections of all 64 90° rotations of the three-dimensional embedding of the IOI from Bach's Courante.**

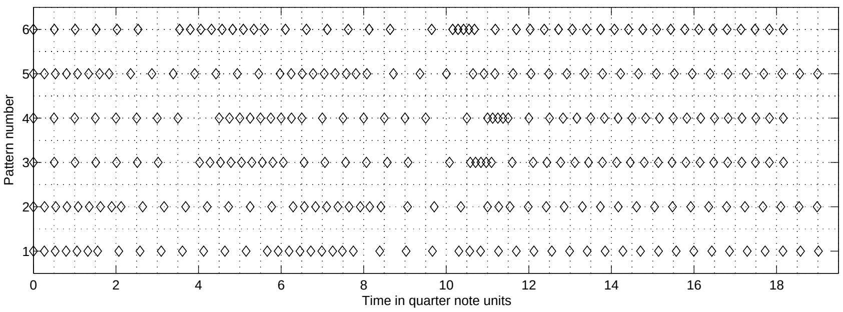


Figure 5-5: **The six non-repeating patterns resulting from** 90° **rotations of the embedding. Pattern number 4 is Bach's original rhythmic sequence.**

of 512, 26 (including the original) are distinct. Figure 5-6 is a plot of all 26 non-repeating patterns. Figure 5-7 shows patterns no.1 and no.25 in musical notation. They are approximations quantized to the 64th note, particularly in cases where no clear rational proportion was found.

From our interpretation, the rhythmic pattern in Bach's Courante is

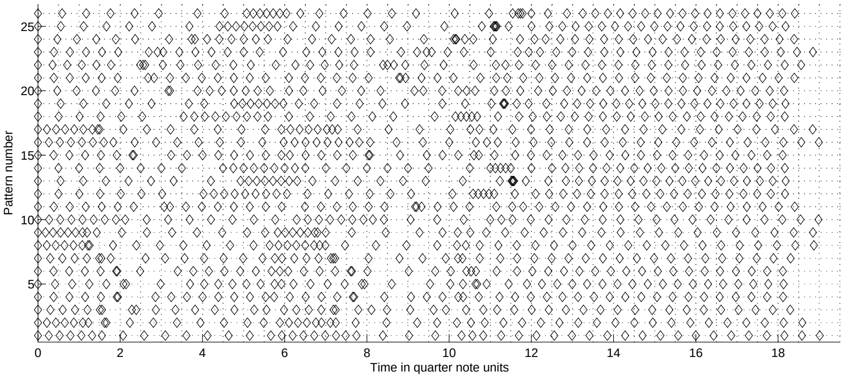


Figure 5-6: **Non-repeating patterns resulting from all 512** 45° **rotations of the three-dimensional IOI embedding. Pattern number 14 is Bach's original rhythmic sequence.**

one of a family of patterns that share a common structure. Here we see one of the advantages of considering musical space as a continuum rather than a discrete alphabet. We have generated new durations not present in the original sequence, as well as new sequences that have a similar structure to the original. Because these new sequences are combinations of the values of an original sequence, this generative approach

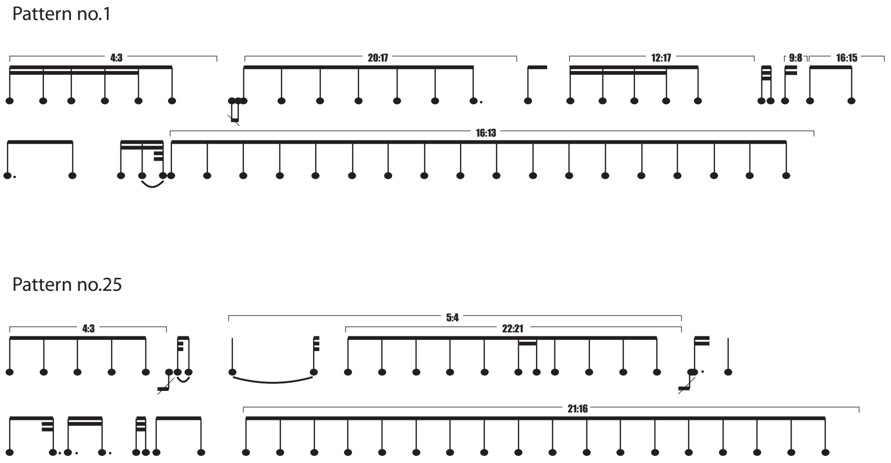


Figure 5-7: **Musical notation of patterns no.1 and no.25 from the 26 non-repeating patterns resulting from 45° rotations of the embedded IOIs.**

can be called *combinational* as opposed to systems based on discrete Markov models such as the one we discussed in Chapter 1, which are typically *permutational*. For an additional example in pitch space see `http://www.media.mit.edu/~vadan/msthesis`.

5.1.3 Decompositions

The top portion of Figure 5-8 shows a fragment of Ligeti's Etude no.4 book 1 in pianoroll representation.[3] The Etude is composed of two layers. One consists of an ostinato characterized by an ascending pattern with a regular rhythm. The other is a homorhythmic polyphonic texture. Throughout the piece, the ostinato is transposed up and down in jumps of one or more octaves. We want to transform Ligeti's Etude so that its global pitch dynamics become continuous. For this we wavelet transform the pitch sequence of Ligeti's Etude using *Daubechies 8* wavelet. The decomposition is done at 7 levels. After decomposing the signal we perform a 4-dimensional state space reconstruction of the largest scale approximation and rotate it by π radians on each of the six rotation planes. We then project the rotated trajectory back to a single dimension and inverse wavelet transform it to recover the complete pitch sequence of the Etude. Figure 5-9 depicts this process. Because the *Daubechies 8*

[3] This pianoroll plot was generated using the MATLAB MIDI toolbox.[13]

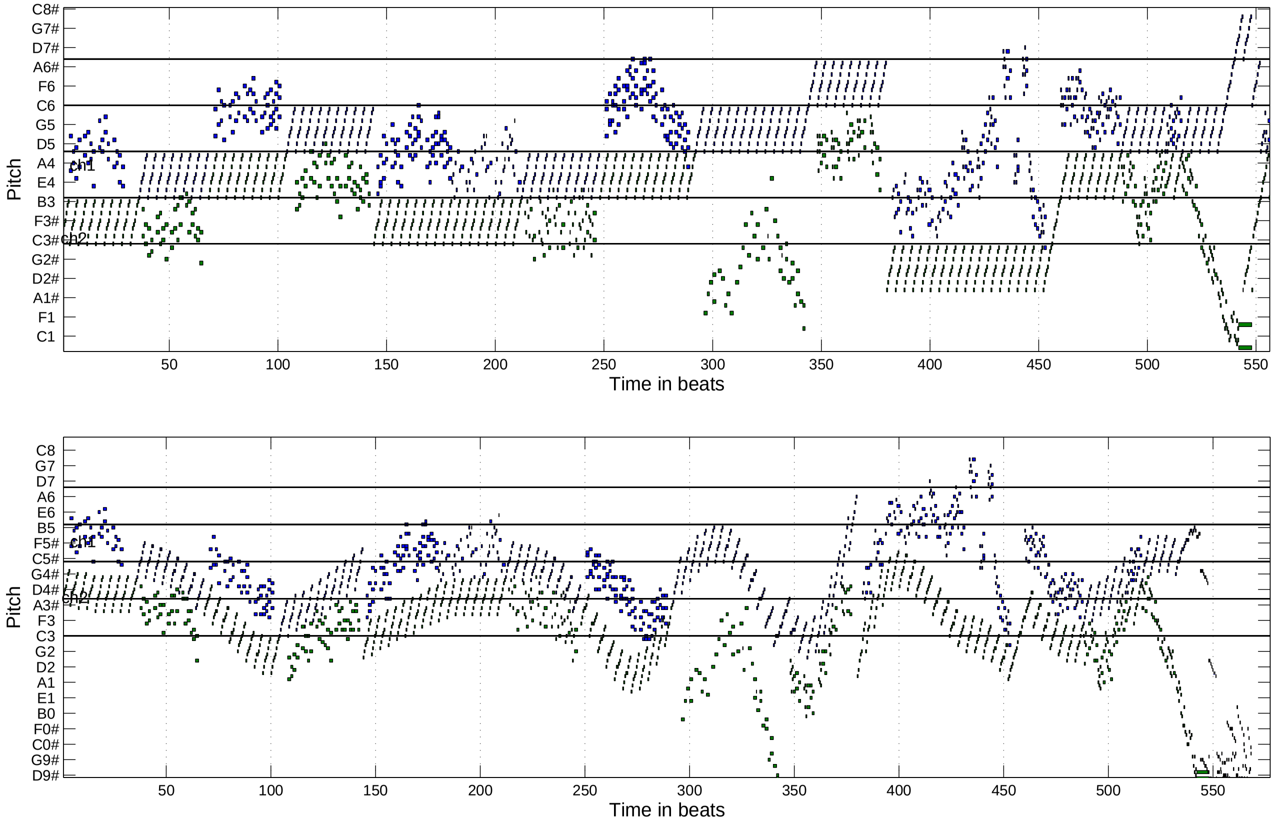


Figure 5-8: ***Top:*** **Fragment from Ligeti's Piano Etude no.4, Book 1.** ***Bottom:*** **Etude no.4 after a π radians rotation of the state space reconstruction of the 7th level approximation using *Daubechies 8* wavelet.**

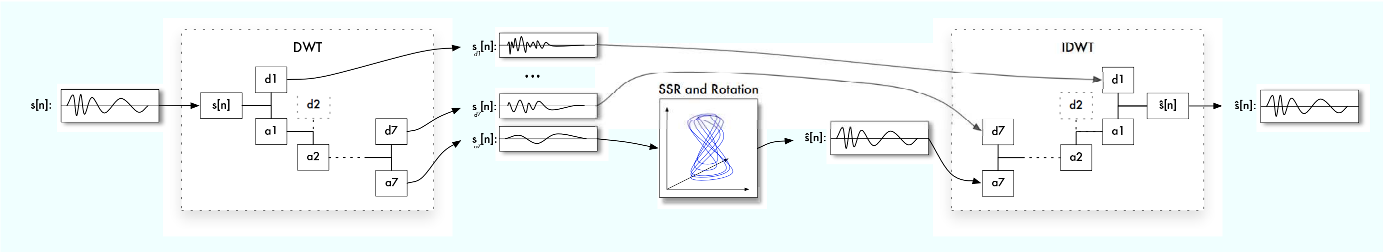


Figure 5-9: **Diagram of the transformation process used to vary and render continuous the large scale pitch contour of Ligeti's Etude. First, the pitch sequence is discrete wavelet transformed. Then, the largest scale approximation is embedded in three-dimensional lag space and rotated to obtain a variation of the original trajectory. Finally, after replacing the original approximation with the variation, the decomposition is inverse discrete wavelet transformed.**

wavelet is smooth, the resulting transformation is smooth as well. The bottom of Figure 5-8 depicts the result of the transformation.

5.2 Outside-Time Structures

In this work we have focused our attention on the dynamic properties of music, and have pointed out the advantage of considering the musical space as a continuum rather than a discrete alphabet. But in addition to the temporal aspects of music, a lot of the quality of a piece comes from its "outside time" structures [43] as well. These consist of the sieves through which the different musical components are discretized. For example, a piece sieved through a major scale feels very different from the same piece sieved though a Phrygian scale. While a hard distinction between in-time and outside-time may sometimes be inaccurate (modes are a case where the sieve is dynamically defined), the distinction provides a simple way of modeling music structure.

5.2.1 Sieve Estimation and Fitting

Both for practical as well as perceptual reasons, it is many times useful to consider scales as cyclic structures. In Chapter 3 we mentioned how the helical pitch space is a better representation of pitch perception than a straight line. By collapsing the helix into a circle we can reduce the infinite number of steps in the helical staircase to a compact set of equivalent pitches classes. Indeed, in musical pitch theories such as Forte's [17] or Estrada's [14], there is the notion of *octave equivalence*. This notion allows the reduction of the pitch space into a small set of pitch classes. This is simply a modulo operation on the pitch space, and mod 12 is the modulo typically used for the twelve tone equal tempered scale. Similarly, we may define a time sieve in the rhythmic dimension (IOIs) via a modulo. If there exists a clear beat or higher rhythmic structure like a meter, then we can reduce all the IOI to a time sieve modulo the meter. Thus, the same helical structure used for pitch space can be used to represent similarity (or equivalence) between events in relation to their placement in time. Figure 5-10 depicts an example of this representation for a triple-time meter.

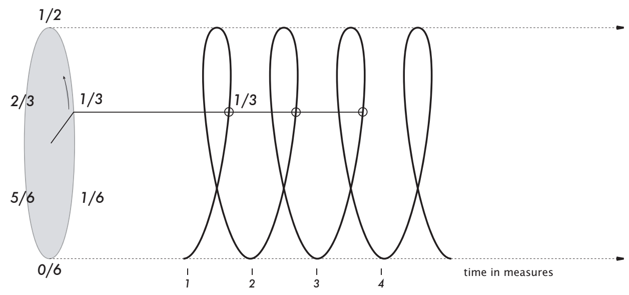


Figure 5-10: **A helical time line representing equivalent points in a measure in triple-time.**

Here we take a simple statistical approach to sieve modeling by computing a histogram of the values found in each of the musical components.[4] Essentially, the histogram of each component is the model of the sieve. To sieve a newly generated trajectory we take two measurements into consideration:

1. The relative frequencies $rf(E_k)$ and $rf(E_{k+1})$ of each of the two scale values E_k and E_{k+1} surrounding point $\hat{s}_n$ in the new series. i.e. $E_k \leqslant \hat{s}_n \leqslant E_{k+1}$.

2. The relative distance between the point to be fitted, $\hat{s}_n$, and each of its surrounding scale values. i.e. $\|E_k - \hat{s}_n\|$ and $\|E_{k+1} - \hat{s}_n\|$.

The complete equation used to replace the estimated value by that of the sieve is:

$$\frac{rf(E_k)^p}{\|E_k - \hat{s}_n\|} \quad \overset{E_k}{\underset{E_{k+1}}{\gtrless}} \quad \frac{rf(E_{k+1})^p}{\|E_{k+1} - \hat{s}_n\|}$$

We use the exponent p to adjust the weights of the relative frequencies versus the distances. If the distance between the generated point $\hat{s}_n$ and its surrounding sieve points is considered to be more important than their relative frequencies, then we use a low value for p and vice versa.

[4] Contrary to traditional tonal music theory, where a scale in pitch space is defined by the underlying functional harmony, all pitches found in a piece are considered part of the scale.

5.3 Musical Chimaeras: Combining Multiple Spaces

An interesting problem in automatic music generation is that of combining two or more pieces to generate a new one. This is not a trivial task, and in the context of the present work the most natural thing to do is to combine the reconstructed state spaces. There are three basic ways of doing this:

Method 1: Generate a new trajectory from the average of the flows estimated for each of the embeddings:

$$\mathbf{z}_{n+1} = aF(\mathbf{x}_n) + bF(\mathbf{y}_n). \tag{5.4}$$

In this approach we estimate the behavior of a point in each of the separately reconstructed state spaces. The estimated velocities are then averaged to obtain the behavior of the new point resulting from the combination. The nature of this method suggests considering the combination of two pieces as a mixture where we can control the percentages of each piece in the mix. For example, in a two piece mixture, we could combine 90% of piece 1 and 10% of piece 2 simply by weight the estimated velocity vectors for each piece: $\mathbf{z}_{n+1} = 0.9F(\mathbf{x}_n){+}0.1F(\mathbf{y}_n)$. As in method 1, all the embeddings must have the same number of dimensions.

Method 2: Estimate a new trajectory from the superposition of the embedded trajectories:

$$\mathbf{z}_{n+1} = F(\mathbf{x}_n, \mathbf{y}_n). \tag{5.5}$$

All spaces must have the same dimensionality. The dimensionality necessary to capture the degrees of freedom of one piece will usually be smaller than that necessary for all the pieces combined. Thus, we estimate the embedding dimension necessary for all the combined pieces by taking them as a single sequence. Once the embedding is made, we iteratively compute $\mathbf{z}_{n+1}$.

For methods 1 and 2 we can ask the following questions: how should the spaces be overlapped? Should they be overlapped without alteration or should they be transformed in some way; maybe translated so that they share the same centroid or rotated using PCA so that their principal components are aligned? Again, the goal of merging two or more pieces is an important determinant of the approach to be taken. If our goal is to keep each of the training pieces as clearly perceivable as possible, then the geometric transformations just mentioned must be avoided or kept to a minimum.

Method 3: Construct a state space from both pieces directly:

$$\begin{aligned}\mathbf{x}_n &= (s1_{n-(k-1)\tau}, s2_{n-(l-1)\tau}, s1_{n-(k-2)\tau}, s2_{n-(l-2)\tau}, \ldots, \\ &\quad s1_{n-\tau}, s2_{n-\tau}, s1_n, s2_n) \qquad (5.6)\end{aligned}$$

In this approach instead of superposing two m-dimensional spaces, we create a new $k+l$-dimensional space by combining an k and a l-dimensional spaces. Thus, in this method the number of dimensions for the reconstructed state space of each piece may be different. While in methods 1 and 2 each lag space dimension is a combination of both training pieces, in method 3 half of the dimensions are defined by one piece and half by the other. Thus, each point in the lag space is defined by both pieces. The difference is important because our new observation will be a projection of the lag space onto one dimension per component. What dimension should this be? Should it belong to training piece 1 or piece 2? The piece to which the projection dimension belongs to will dominate the new output. This can be seen as the carrier piece, while the other is the modulator. As discussed in Chapter 4, this approach can be interpreted as two systems driving each other.

To show how each of these methods performs, we combine two pieces with each of the three methods. We are particularly interested now in trying to preserve clearly perceivable characteristics from the two training pieces. Thus, we do not alter the embeddings of any of the pieces in any way prior to their combination. The pieces we use are Ligeti's Piano Etudes 4 and 6 from Book 1. Again, in order to see how these three methods behave under different parameter configurations, we systematically explore several values for the three parameters: embedding dimension, the number of nearest neighbors, and the the weight of the neighbors relative to their distance (the p exponent). We generate 500 points (notes) for each test, again using the mean of the reconstructed trajectory as the initial point.

Method 1 This method seems to be the one less likely to work. Looking at Figure B.4 and listening to the sequences generated with this method we see that there is little resemblance between these and the original Etudes. The pieces generated are noisy. In other words, there is little structure and the music sounds quite random. This is especially true for low dimensional embeddings. As the embedding dimension increases, though, the global dynamics become more varied and interesting. Yet, the local dynamics are usually very irregular and the source pieces imperceptible. Imagine embedding many pieces of music, each in their own state space. What is the velocity of a single point in each of the spaces? We will most likely discover these velocities to be very different. Averaging these estimated velocities will then result in unpredictable dynamics and possibly neutralization due to opposing velocities.

Method 2 The choice of the number of nearest neighbors in the lag space interpolation is important in defining the mixture of the pieces. If only one neighbor is used, the dynamics of each point will be defined by only one of the pieces being combined. The greater the number of nearest neighbors, the greater the chances that the behavior of a point will be defined by points from the two training pieces. The p exponent equally affects the mixture. As we have seen, too high a value of p will have practically the same result as using only one nearest neighbor since the weights of distant points will be very small compared to those of nearby points.

Figure 5-11 is a pianoroll plot of a 500 notes sequence generated from the mixture. The parameters used to generate this result were $p = 200$, number of neighbors=11, and embedding dimension=3. What is characteristic about this method is the fact that it tends to behave like a *collage*, deserving the chimaera alias (this can clearly be seen in Figure 5-11). This is because in the superposition of the two state spaces, the

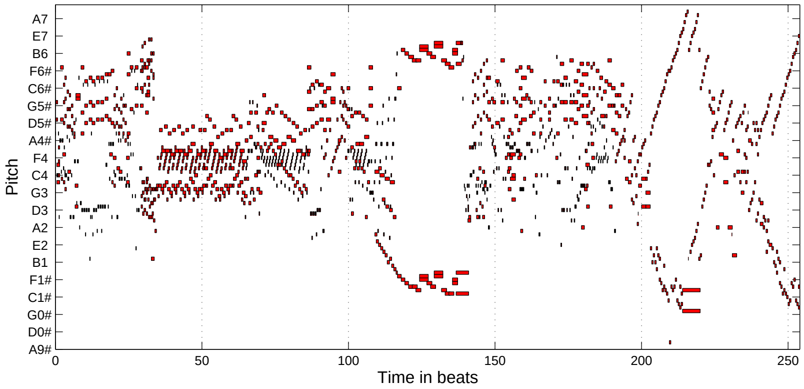


Figure 5-11: **Mixture of Ligeti's Etudes 4 and 6 Book 1 using *method 2*, with parameters $p = 200$, number of neighbors=11, and embedding dimension=3.**

trajectories from both pieces will not always be close to each other. At some point in the combined space, trajectories will diverge, while at others they will cross or pass close to each other. Thus, at some moments the new estimated trajectory will be dominated by one piece; at places where the trajectories of the training pieces meet there will be a mixture and/or a shift of dominance in the estimation from one piece to the other, making the resulting sequence a kind of splice and mix, and reordering of fragments of the training pieces. Naturally, the degree to which pieces will be combined depends on their degree of similarity. Similar states between pieces will be close to each other, while states that are not similar will be apart. It is important, then, to find a representation that will make the spaces overlap as much as possible, but, again, without altering them since we want to be able to hear the original training pieces in the new piece. Therefore, we modify the components of the sequence (pitch, loudness, IOI, duration) of the training pieces so that they match as much as possible without altering perceptual features. For example, a histogram of the IOI of Etude 4 shows that the most frequent IOI values is 0.5 (a quarter note), while that of Etude 6 is 0.25 (an eight note) (Figure 5-12). Because the difference can be interpreted simply as that of *tempo*, (which generally is not a defining characteristic of the piece unless the difference between *tempi* is too great) we scale the IOI of one of the Etudes to equal that of the other. A similar transformation can be applied to the pitch component. If the register is not something we may consider very important, we can use the sequence of pitch intervals (the Inter Pitch Interval) simply by taking the difference of the pitch sequence. We don't do this in the present example though, keeping the original pitch sequences unchanged.

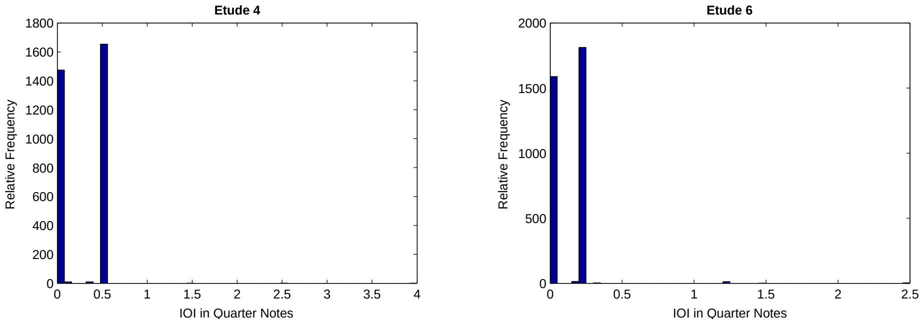


Figure 5-12: **Histogram of IOI of Ligeti's Etudes 4 and 6.**

The dimensionality of the superimposed spaces is also an important factor defining the degree to which the new estimated trajectory will be a combination of the two training trajectories. Since each point is defined by a sequence of notes in the training pieces, the higher the dimensionality of the state spaces, the less likely it will be that points from different pieces will be close to each other. Thus, the best results are obtained when the dimensionality of the embedding is low. On the other hand, as we saw before, choosing too low a dimensionality will make the high scale dynamics of the resulting trajectory more chaotic.

Method 3 The third method is probably the most intriguing. In lag space, two signals embedded together can be interpreted as two signals driving each other, in a similar way as planets pull and affect each other's motion.

As in method 1, the generated sequences resulting from the combination of both pieces into a single state space are generally noisy. Figure B-6 in Appendix B shows the result of combining Ligeti's Etudes with this method for multiple parameter values. Clelarly, the resulting trajectories from this mixture are overly noisy. But rather than embedding all the parameters of both pieces in a single space, we can embed each of the four components (pitch, loudness, IOI and duration) independently (see bottom of Figure 5-1) and combine the spaces corresponding to the same component (figure 5-13). Figure B-7 shows several 500 note sequences generated with this configuration for the same parameter combination as with methods 1 and 2.

The choice on the exponent p greatly affects the output of the interpolation using method 3. From the multiple panels in the figures it can be seen that no matter what the choice of dimension or number of nearest neighbors is, when $p = 2$ the resulting sequence is noisier than with higher exponents. In method 3, p is clearly the most important factor determining the degree to which the points interact, and thus, the degree to which the pieces mix. It could be labeled the “mixing level knob” of the method.

What state spaces Are Made Of

In addition to the alternative ways of combining state spaces, there are also multiple ways of defining the spaces to be combined. We have jointly modeled the dynamics of all the components of a piece (e.g. pitch, loud-

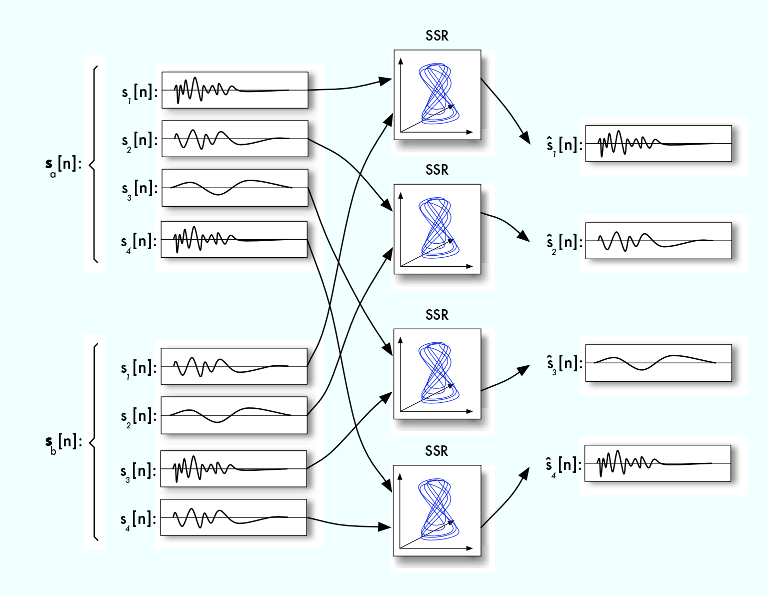


Figure 5-13: **Each component is embedded separately**

ness, brightness, IOI), and each component independently. We've also seen that we can decompose each component and model each of its subcomponents independently. In addition, we can reconstruct a state space from combinations of subcomponents derived from corresponding components of different pieces. For example, the large scale approximations of pitch from various pieces on one side, and their corresponding details on another (Figure 5-14). We could also take the high scale dynamics from a component in one piece and the low scale dynamics from another, or even combine different components from different pieces. These more intricate and experimental approaches may result in unique variations, but the original training pieces will most certainly be lost perceptually.

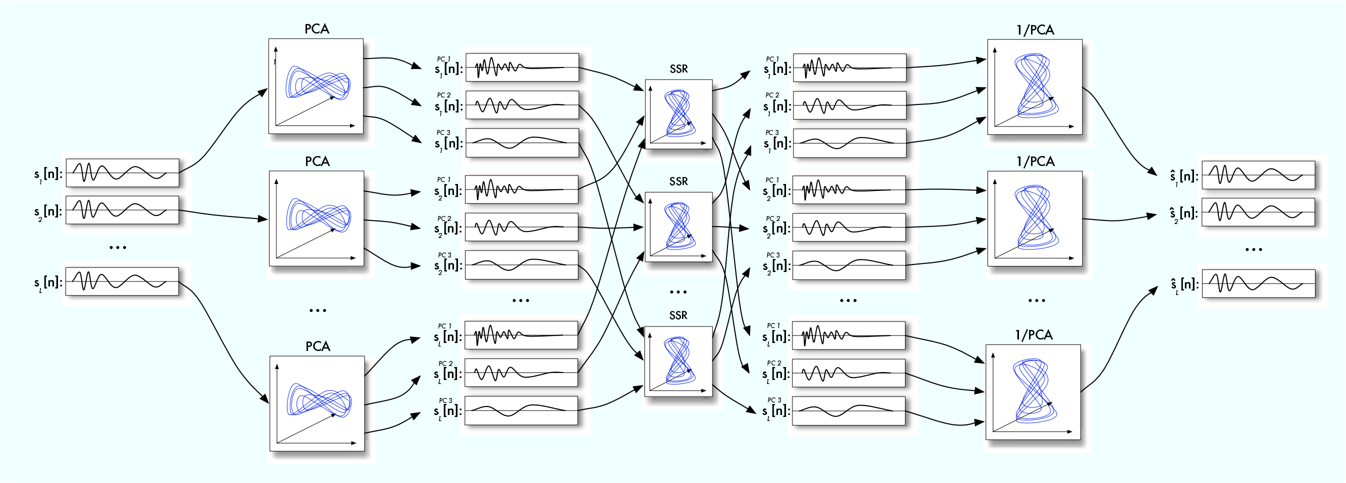


Figure 5-14: **Corresponding components from different pieces are combined at different scales, but each scale is independent from the others.**

CHAPTER SIX
Experiments and Conclusions

6.1 Experiments

In this work we have presented methods to generalize and extrapolate musical structures from existing pieces for the purpose of generating novel music. Thus, what we might evaluate is the novelty of the pieces generated with these methods in relation to the source or training pieces. But testing whether a piece generated by the system is different from the training piece is trivial. As we showed in the previous chapter we can generate completely new sequences with enough transformations, however arbitrary these may be. What we are actually looking for is that ambiguous middle ground where the generated pieces share structural characteristics with the training piece(s) while at the same time are as novel as possible. Clearly, this is as much a test on the system as it is on the system's user, since the art still is, after all, in the creative use of the tools.

6.1.1 Merging Two Pieces

In the previous chapter we discussed three ways of combining two pieces of music to generate new ones. In this chapter we present the results of an experiment performed to evaluate how well each of the three music-merging methods is able to generate novel musical pieces while still preserving structural characteristics of the original training pieces.

Experiment Setup

In this experiment subjects were presented with 10 excerpts: 5 pairs, each generated with a different method:

Pair 1 Using method 1 with a single state space reconstruction.

Pair 2 Using method 2 with a single state space reconstruction.

Pair 3 Using method 3 with a single state space reconstruction.

Pair 4 Using method 3 with components modeled independently.

Pair 5 Randomly, using a gaussian distribution for each musical component. The parameters of the gaussian distributions used are shown in Table 6.1.

Table 6.1: Parameters of the Gaussian distribution used for the randomly generated musical sequences.

	μ	σ
IOI	0	0.25
duration	0.25	0.5
pitch	60	10
loudness	80	10

We used two excerpts per generation method because each method can generate a wide variety of musical sequences. Using only one excerpt may give misleading results that depend more on the particular example than on the generation method. On the other hand, using more excerpts might have made the experiment too long for subjects to remain interested and attentive during the experiment.

The two pieces used as training data were Ligeti's Etudes 4 and 6, Book I. In all cases the generated sequences (including the random ones) were quantized to fit the union of the sieves of the two training pieces (see Section 5.2). The reason for including randomly generated excerpts was to test whether listeners confused some of the generated pieces with noise. As we pointed out in the previous chapter, method 3 and to a lesser degree method 1 (both with a single state space reconstruction combining all musical components) generate noisy sequences. All excerpts were between 30 and 40 seconds in duration except the training pieces, which were presented complete.

The experiment was divided in two parts: in the first part, subjects were asked to rate the complexity and their preference for each excerpt. They were also asked to cluster the excerpts into groups on the basis of similarity. Subjects were free to create any number of clusters between 1 and 5, and their choice depended on their ability to perceived distinct groups. Evidently, the perfect ear would have perceived the five groups corresponding to the five generation methods used. In the second part they were asked to evaluate the similarity between each of the excerpts and each of the training pieces. The excerpts were labeled [F1], [F2], [F3], etc. in the forms used and were organized as shown in Table 6.2. The forms used in the experiment can be found in Appendix C.

Table 6.2: Labels used for the excerpts in the experiment form and their corresponding Pair type (i.e. production method).

F1	excerpt from Pair 1
F2	excerpt from Pair 2
F3	excerpt from Pair 3
F4	excerpt from Pair 5
F5	excerpt from Pair 4
F6	excerpt from Pair 5
F7	excerpt from Pair 1
F8	excerpt from Pair 2
F9	excerpt from Pair 4
F10	excerpt from Pair 3

Experiment Part 1 Results

None of the subjects perceived five distinct groups: 30.77% of the subjects distinguished four groups, 46.15% distinguished three, and 23.077% distinguished two. How were the excerpts grouped? Were the two excerpts generated with the same method (and thus belonging to the same Pair) actually clustered together? Table 6.3 shows the answer for each subject to this question. For convenience, the bottom line in the table shows the number of clusters found by each subject. Because all of the subjects perceived four clusters or less, there is necessarily some overlap of Pair types in every subject. Clearly, the fewer the number of clusters found, the greater the chances that two Pairs will be clustered together. Subject no.2, for example, correctly clustered togethered the excerpts belonging to Pairs 3, 4 and 5, but only defined two clusters total. This means that at least two of these Pairs were clustered together. Which pairs were clustered together? More formally, if Pair $P = \{p_1, p_2\}$ and

Table 6.3: Were the two excerpts generated with the same method clustered together (1=yes, 0=no)?

Subject no.:	1	2	3	4	5	6	7	8	9	10	11	12	13
Pair 1:	1	0	1	1	0	1	0	0	0	0	0	1	0
Pair 2:	1	0	1	0	0	1	0	0	0	1	1	0	0
Pair 3:	0	1	0	0	0	1	0	1	0	0	0	0	1
Pair 4:	1	1	1	0	0	0	1	1	0	1	1	0	0
Pair 5:	1	1	0	1	0	1	0	0	0	1	0	1	0
No. of Clusters:	4	2	3	4	4	3	3	2	3	4	3	3	2

Pair $Q = \{q_1, q_2\}$, the question then is $(p_1 = q_1) \wedge (p_1 = q_2) \wedge (p_2 = q_2)$?. Table 6.4 shows the results of this operation applied to every pair of Pairs for each subject. In this Table we see that subject no.2 clustered together Pairs 3 and 5. We can see that very few Pairs were consistently clustered together: Pairs 3 and 4 for subject no.2 as we have just seen, and Pairs 1, 2 and 3 for subject no.6.

Table 6.4: Which of the Pairs were consistently clustered together (1=yes, 0=no)?

Subject no.:	1	2	3	4	5	6	7	8	9	10	11	12	13
Pairs 1 and 2:	0	0	0	0	0	1	0	0	0	0	0	0	0
Pairs 1 and 3:	0	0	0	0	0	1	0	0	0	0	0	0	0
Pairs 1 and 4:	0	0	0	0	0	0	0	0	0	0	0	0	0
Pairs 1 and 5:	0	0	0	0	0	0	0	0	0	0	0	0	0
Pairs 2 and 3:	0	0	0	0	0	1	0	0	0	0	0	0	0
Pairs 2 and 4:	0	0	0	0	0	0	0	0	0	0	0	0	0
Pairs 2 and 5:	0	0	0	0	0	0	0	0	0	0	0	0	0
Pairs 3 and 4:	0	0	0	0	0	0	0	0	0	0	0	0	0
Pairs 3 and 5:	0	1	0	0	0	0	0	0	0	0	0	0	0
Pairs 4 and 5:	0	0	0	0	0	0	0	0	0	0	0	0	0
No. of Clusters:	4	2	3	4	4	3	3	2	3	4	3	3	2

To see to what degree the clusters defined by the subjects were a combination of excerpts belonging to different Pairs, we asked the question:

Is any of the two excerpts belonging to Pair $P = \{p_1, p_2\}$ clustered with any of the two excerpts belonging to Pair $Q = \{q_1, q_2\}$? More formally, $(p_1 = q_1) \vee (p_1 = q_2) \vee (p_2 = q_1) \vee (p_2 = q_2)$?. Table 6.5 shows the results. Looking at the Tables we can see the variety of the clustering results across subjects. Figure 6-1 shows graphically the clusters made by a few subjects. On one extreme we have subject no.1 who found four clusters and correctly grouped the excerpts of Pairs 1, 2, 4 and 5. He/She correctly isolated Pairs 4 and 2, but clustered one of the excerpts belonging to Pair 3 with Pair 1, and the other one with Pair 5. On the other extreme we see Subject no.13. He/She found only two clusters, and only correctly clustered together the excerpts belonging to Pair 3. The excerpts belonging to the rest of the Pairs were divided in separate clusters. Subject no.10 also did a good job in clustering the excerpts. He/She was able to discriminate the noise excerpts (Pair 5) from the rest, but confused excerpts from Pair 1 with those of Pair 2, and those from Pair 3 with Pair 4.

Table 6.5: Is any of the two excerpts belonging to Pair $P = \{p_1, p_2\}$ clustered with any of the two excerpts belonging to Pair $Q = \{q_1, q_2\}$?

Subject no.:	1	2	3	4	5	6	7	8	9	10	11	12	13
Pairs 1 and 2:	0	1	0	1	1	1	1	1	1	1	1	1	1
Pairs 1 and 3:	1	1	0	1	1	1	1	1	1	1	1	1	1
Pairs 1 and 4:	0	1	0	1	0	1	1	1	1	0	0	1	1
Pairs 1 and 5:	0	1	1	0	1	0	1	1	1	0	1	0	1
Pairs 2 and 3:	0	1	1	1	1	1	1	1	1	0	1	1	1
Pairs 2 and 4:	0	1	0	1	1	1	1	1	1	0	0	1	1
Pairs 2 and 5:	0	1	1	0	1	0	1	1	1	0	0	0	1
Pairs 3 and 4:	0	0	1	1	0	1	0	0	1	1	1	1	1
Pairs 3 and 5:	1	1	1	0	1	0	1	1	1	0	1	0	1
Pairs 4 and 5:	0	0	0	0	1	0	1	1	1	0	1	0	1
No. of Clusters:	4	2	3	4	4	3	3	2	3	4	3	3	2

From our own observations of each of the three music-merging methods presented in the previous chapter, we were expecting subjects to frequently confuse the randomly generated excerpts (Pair 5) and those generated with method 3 on a single state space reconstruction (Pair 3). Adding across subjects in Table 6.5 we see that Pair 5 was clustered with Pair 1 by 8 subjects, with Pair 2 by 7 subjects, with Pair 3 by 9 subjects

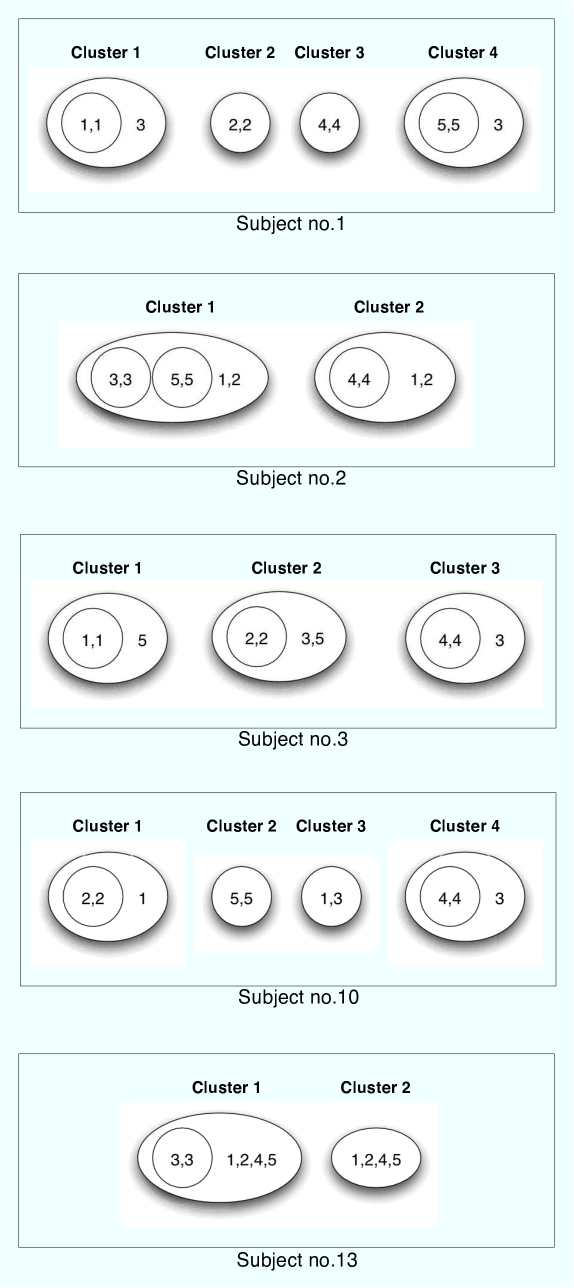


Figure 6-1: **Clusters found by several Subjects.**

and with Pair 4 by 6 subjects. These results match our hypothesis: Pair 5 was most confused with Pair 3. However, the confusion between the gaussian noise excerpts (Pair 5) and those belonging to the rest of the Pairs was more frequent and homogeneous than what we expected.

We also expected subjects to clearly distinguish the excerpts generated with method 3 with components modeled independently (Pair 4) from the rest of the groups. Again, adding across subjects in Table 6.5 we see that Pair 4 was clustered together with Pair 1 by 8 subjects, with Pair 2 by 9 subjects, with Pair 3 by 8 subjects and with Pair 5 by 6 subjects. Thus, the confusion of Pair 4 with the rest of the Pairs was also very homogeneous.

Why did we obtain such fuzzy results? Of course, since we discretized all excerpts with the same sieves, they all share the same “outside time” structure. This common structure no doubt plays a big role in the subjects perception of similarities between the noise excerpts and the training based ones. It is likely that the longer the musical excerpts, the more subjects rely on these “outside time” structures (which are actually statistics abstracted from temporal placement) for evaluating structural similarities. Thus, it is likely that the sieves have a greater weight on the perception of similarity between the excerpts than what we thought. How would the results change if we used shorter excerpts? What if we had not sieved the gaussian excerpts? How distinguishable would they be in this case? These questions will have to be left open for future experiments.

The less relevant but nonetheless interesting results were subject’s perception of complexity and preference of each of the generated excerpts. Figure 6-2 shows the resulting means (bars) and standard deviations (lines) of complexity and preference across subjects. The most preferred excerpts were 5, 8 and 9, which belong to Pairs 4, 2 and 4 respectively. Even so, there is a wide variance among subjects in their perception of both complexity and preference. There isn’t a clear relationship between preference and complexity, but its curious that the most preferred is also the most complex.

Experiment Part 2 Results

To evaluate how well each of the generated excerpts preserved structural characteristics of the original training pieces, subjects were asked to rate the similarity between each of the generated excerpts and each of the

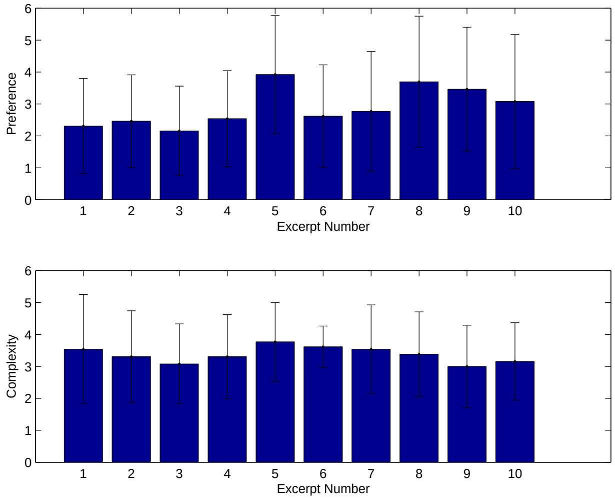


Figure 6-2: **Means and standard deviations across subjects of their evaluation of Complexity and Preference for each of the excerpts.**

two training pieces. Figure 6-3 shows the means (bars) and standard deviations (lines) of the estimates across subjects.

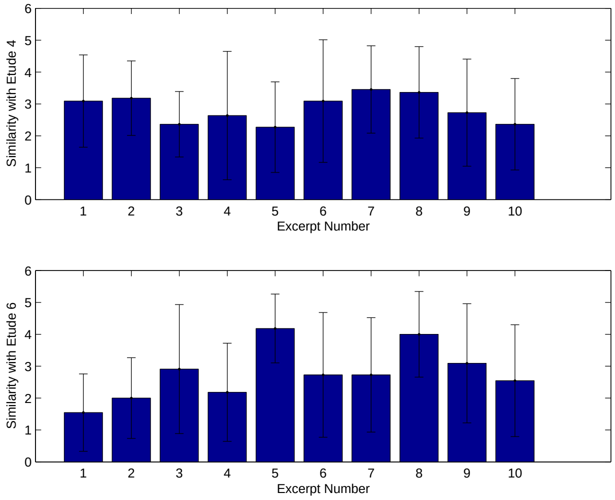


Figure 6-3: **Means and standard deviations across subjects of the similarity estimates between each of the excerpts and each of the training pieces: Ligeti's Etude 4 and Etude 6.**

The similarity estimates between the generated excerpts and Ligeti's Etude 6 are fairly clear. With a relatively small variance, excerpt number 5 comes first with the highest similarity estimate followed closely by excerpt 8. As we see in Table 6.2, excerpt 5 belongs to Pair 4 which was generated with method 3, parameters modeled independently. Excerpt 8 belongs to Pair 2 which was generated with method 2, components modeled jointly. It is interesting to note that these two excerpts coincide with the two most preferred. How were the other excerpts belonging to these pairs rated? The second excerpt belonging to Pair 4 is excerpt number 9, which is the third most similar to Etude 6. Strangely, the second excerpt belonging to Pair 2 (excerpt number 2) has the second lowest similarity to Etude 6.

How can we explain these results? As discussed in Chapter 5, in contrast to methods 2 and 1, method 3 is characterized by a dominance of one piece. How one piece dominates over another depends on the state space dimensions chosen for the projection of the novel trajectory. For the generation of the excerpts using this method we projected the newly generated trajectory onto the dimensions defined by the components of Etude 6. It is not strange then that Pair 4 resulted in the highest similarity estimate to Etude 6. Method 2 can best be described as a "collage" method because many times a new state space trajectory will be defined by one of the pieces for a period of time until an intersection of trajectories belonging to different pieces is reached, in which case there may be a shift from one piece's dominance to the other. The reason why excerpt 2 may have done so poorly is that the initial condition in the generation of this excerpt may have fallen close to the state space trajectory of Etude 4 and never shifted to the trajectory of Etude 6. In other words, this excerpt fell in a basin of attraction of Etude 4 and never left it.

The similarity estimates between the generated excerpts and Ligeti's Etude 4 are not so clearly different. The highest similarity scores were given to excerpts 7 and 8, which belong to Pairs 1 and 2, followed by excerpt 2 belonging also to Pair 2. The lowest similarity estimates were given for excerpts 3 and 5, which came high in similarity with Etude 6. Excerpt 3 was generated with method 3 components modeled jointly, and excerpt 5 with method 3 components modeled independently.

Which excerpt is the most similar to both training pieces? To answer this question we calculated the mean across subjects of the product of the similarity estimates between a given excerpt and $Etude4$, and the same excerpt and $Etude6$. More formally, let $sim(a, b)$ be the similarity estimate between a and b. Then, the similarity between excerpt $\mathbf{x}$ and both training pieces is $sim(\mathbf{x}, Etude4) \cdot sim(\mathbf{x}, Etude6)$. Figure 6-4 shows these results. Excerpt 8 is clearly the excerpt with the most similarity to both Etude 4 and Etude 6. It belongs to Pair 2, and was thus generated with method 2.

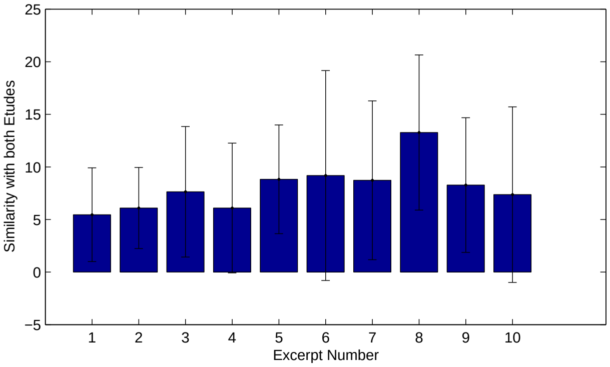


Figure 6-4: **Means and standard deviations across subjects of the similarity estimates between each of the excerpts and both training pieces.**

It is not easy to state definite unambiguous conclusions given the results obtained. From the results presented we could conclude that method 2 is the most successful in terms of preserving structural characteristics of both training pieces because excerpt 8 scored the highest in the similarity estimates of both pieces combined. Yet, excerpt 2 (also generated with method 2) obtained a very low combined similarity to both training pieces. As we mentioned in the previous chapter, a small change in the initial condition of a state space interpolation can greatly alter the evolution of the resulting observation sequence. Thus, while method 2 may be an excellent method for combining two pieces, an optimal result may depend on the careful choice of model parameters and initial conditions.

Method 3 with components modeled independently strongly preserves structure from one of the training pieces. Thus, rather than as a method of combining two pieces, it might be better understood as a method of modulating one piece with another.

The similarity ratings between the gaussian noise excerpts (Pair 5) and the two training pieces were surprisingly high. How is it possible that these excerpts were rated with comparable similarity values to those generated with some of our methods, particularly methods 1 and 3 with components modeled jointly? In our own perception, any of the excerpts generated with either method 1, 2 or 3 are clearly better at preserving structure from the training pieces than the gaussian noise examples. Even the excerpts generated with method 3 with components modeled jointly (which we saw in Chapter 5 generates noisy sequences) have large scale evolutions that the gaussian noise excerpts don't, thus potentially making them distinguishable. Could it be that we have overestimated our methods, or could it be that we have overestimated the subjects' ability to perceive music structure, or could it be both?

Most subjects who commented on the experiment expressed their difficulty in distinguishing clear differences between the excerpts. Indeed, in general the results from both the clustering and the similarity experiments reveal poorer aural discriminability than what we expected. But the large variances in all the results also reveal a wide range of ears.

Some subjects also talked about their experience with judging similarity. A non-musician said she noticed mood changes from excerpt to excerpt, so this was her main criterion for judging similarity. A professional musician said that he noticed his criteria for judging similarity changed from pair to pair. First he started rating the similarity between excerpts without thinking much about it. Later, as he rationalized the task, he realized that his criteria for similarity were many, and that the weight of each criterion changed from excerpt to excerpt. Sometimes some features were more salient than others, thus influencing more his decision about what was similar.

There are clearly a large variety of perceptual approaches between different listeners. Different people focus on different aspects of a piece, and therefore judge similarity in different terms. Each individual's perception of the musical examples given is also influenced by their experience with music similar to the pieces used, and with music in general. It is noteworthy that subjects no.1 and no.10 (who had the best discrimination results) are professional musicians. In addition, subject no.1 was acquainted with both training pieces. Indeed, two Mozart sonatas may sound very different to a listener acquainted with his music, while they may sound as the same piece to an "untrained" person. Humans, just like machines, are trained to perceive the world in different ways, and the training data individuals feed on can be quite different. One man's music is another man's noise, and we see this in the arts continuously. For humans, too, perception requires learning.

6.2 Conclusions and Future Work

In this work we have presented an approach to the problem of inductive music structure modeling from a dynamical systems and signal processing perspective which focuses on the dynamic properties of music. The point of departure of the approach was the reconstruction of a state space from which to obtain essential characteristics of music's dynamics, such as its number of degrees of freedom. This multidimensional representation allowed us to obtain generalizations about the structure of a give piece of music, from which we generated novel musical sequences. We presented three main types of generalization strategies (Chapter 5): **State space Interpolation**, **Abstraction** of the dynamics from the states and **Decomposition** of the musical signals using wavelets and PCA.

We used local linear models to model the reconstructed state space structure deterministically because of the method's ability to generate a variety of transformations, its speed and simplicity. There are, of course, other ways of modeling the state space's reconstructed structure with global methods, such as using polynomials, neural networks or radial basis functions. Our explorations of radial basis functions were brief, so we did not document them here. Nonetheless, it is worth mentioning that we abandoned this method because, in contrast to the local linear models used, the newly generated sequences using this method tended to fall in attractors and thus generated infinite cycles. In addition, unless one uses almost as many radial basis functions as points in state space, the resulting interpolations are smooth. Therefore, this method is not very useful unless one is dealing with continuous sequences only. Other modeling strategies remain to be explored. Of particular importance is the incorporation of stochastic methods into the overall system.

We discussed the tight relationship between states and dynamics, and how a collection of states only allows a variety of state space paths (dynamics) and *vice versa*. We demonstrated the use of rotations of the state space to generate novel state sequences that preserve the same dynamics. However, we did not explore the reverse: keeping the same states while changing their temporal relationships. The classical (and probably only frequently used) transformation in this respect is reversing the sequence, but evidently other reorderings are possible. These remain to be explored.

By decomposing musical sequences hierarchically we were able to achieve an additional level of flexibility that allowed us to perform transformations at different time scales. We discussed two decomposition methods: wavelets and PCA; but we did not explore the more robust family of transformations known as ICA. Particularly for the case of music, where nonlinear relationships frequently exist, PCA offers a limited solution to the problem of obtaining non-redundant decompositions. Thus, future iterations of this work will explore more robust decomposition methods. Related to this problem is that of stream segregation discussed below.

6.2.1 System's Strengths, Limitations and Possible Refinements

In our interest to be as unbiased as possible towards music, we have tried to make our approach to music structure modeling very general. In doing so we have payed the price of generality. Many of the sequences that we have generated are rather coarse approximations that frequently miss some perceptually important characteristics. As we stated in Chapter 1, it is really not possible to model all kinds of music with a single approach, so while the approach presented here may be a reasonable point of departure for any musical piece, the particularities of each work can only be faithfully captured with more specific and more refined methods.

What refinements could we implemented that would still be applicable to a broad variety of music? We presented methods of decomposing musical signals hierarchically, making it possible to transform the details or the large scale approximations of the dynamics of music independently. But we did not discuss or attempt to extract repeating patterns such as "themes" or motives that could be treated independently form other musical materials. Thus, a further refinement of the system would be to try to find these kinds of structures, and to segment the music in more detail. Another more ambitious task would be to segregate streams that may be running in parallel in a single musical component such as pitch. Musical multiplexing is quite common in monophonic pieces, and the separation of perceivable streams or *voices* would allow greater control and refinement of the possible transformations applied to music.

While we briefly mentioned the possibility of modeling the dynamic properties of sieves, in all the examples presented and experiments performed we modeled the sieves as a stationary structure spanning the entirety of the piece(s). A more refined yet still general modeling approach would consider sieves not just as stationary "outside-time" structures, but as dynamic ones as well.

As can be seen and heard, the approach to music structure modeling we have presented is quite good at capturing the large scale dynamics of music and generating novel sequences at these large temporal scales. But due to the filtering made by the local linear models, we have had to sacrifice some of the high energy dynamics which are so clearly perceivable and thus fundamental, frequently making the short time-scale sequences sound unrelated to the training piece.[1] Thus, we have a reverse tradeoff to the one typically found in Markov models and in some music-generative applications of neural networks [29]. A possible solution to this problem might be a more in-depth use of hierarchical decomposition of the musical sequences presented in Chapter 4. Because the high frequency content is removed from the large scale dynamics (e.g. wavelet *approximations*), we could safely use many neighbors in the state space interpolation without causing any additional filtering. Meanwhile, the state space reconstruction of the *details* would be interpolated with very few neighbors to avoid filtering high frequencies in the manner we have discussed in the previous chapter. By combining these two models it might be possible to obtain good small scale variations as well as large scale ones.

[1] Remember that, in order to reduce the filtering, we typically had to reduce the number of points in the velocity estimates or increase the exponent p in Equation 3.8.

Appendix A
Notation

x		Scalar variable.
$\mathbf{x}$		Vector.
$s[n]$		Discrete scalar signal.
$\mathbf{s}[n]$		Discrete multi-dimensional signal.
$s(t)$		Continuous scalar signal.
$\mathbf{s}(t)$		Continuous multi-dimensional signal.
$\langle a, b\rangle$		Inner product of a and b.
ab^T		Outer product of a and b, where a and b are vertical vectors.
$\|\mathbf{x}-\mathbf{y}\|$	$=$	$\sqrt{(x_1-y_1)^2+\ldots+(x_m-y_m)^2}$
$C\binom{m}{2}$	$=$	$\frac{m!}{2!(m-2)!}$
C^2		Two times continuous differentiable function.
$\wedge$		Logical AND.
$\vee$		Logical OR.
$E[x]$		The expected value of x.

Appendix B
Pianorolls

B.1 Etude 4

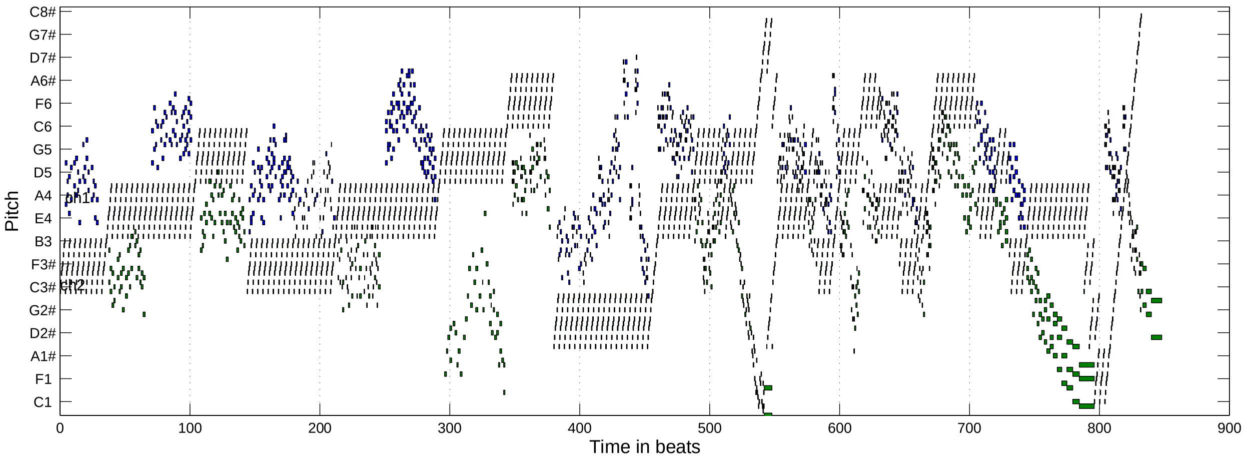


Figure B-1: **Ligeti's Etude 4, Book1.**

B.2 Etude 6

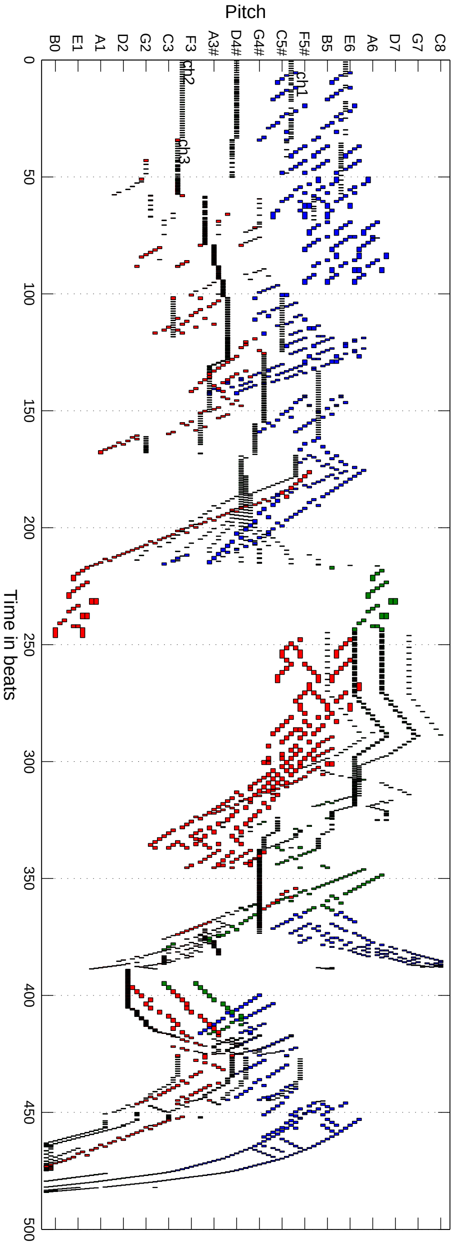


Figure B-2: Ligeti's Etude 6, Book1.

B.3 Interpolation: Etude 4

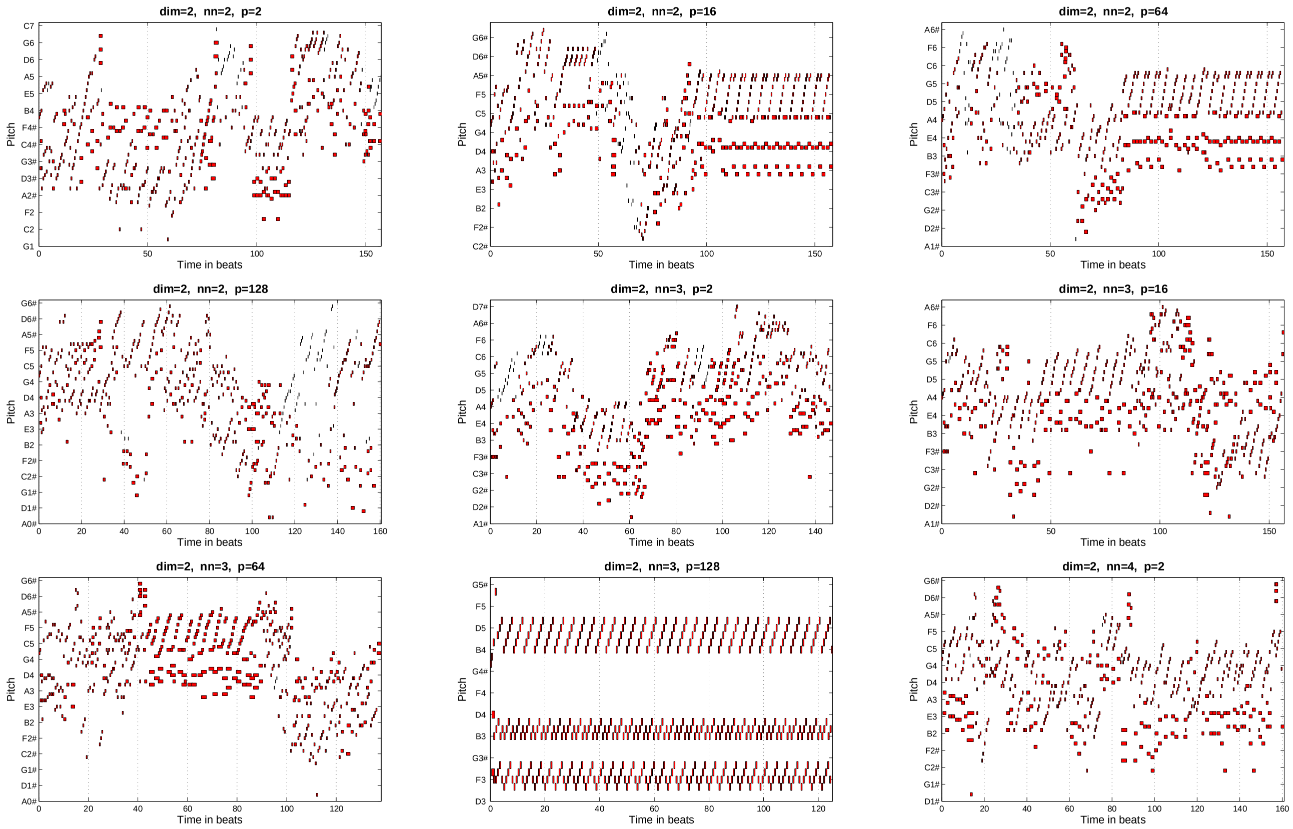
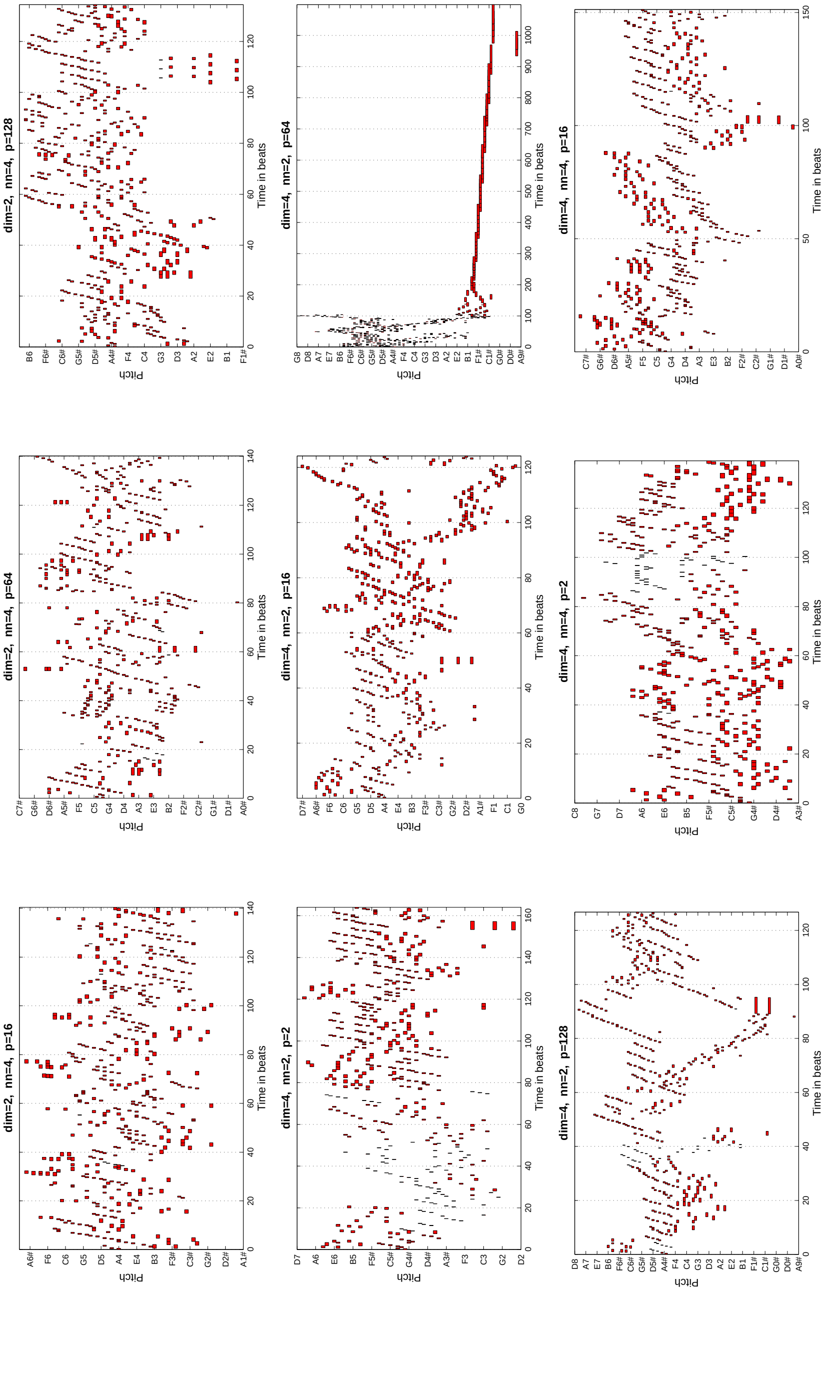
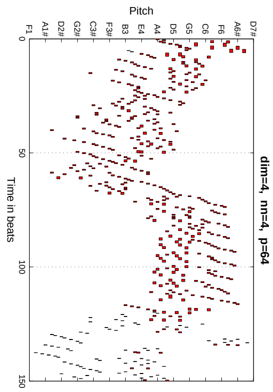
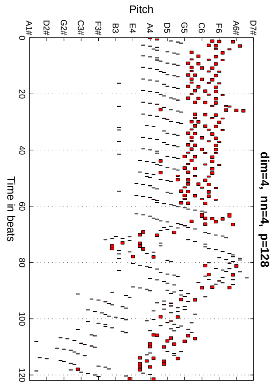
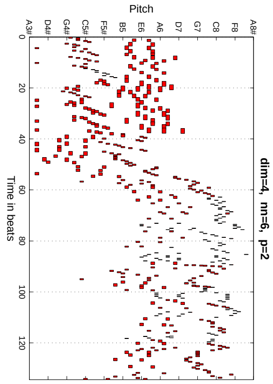


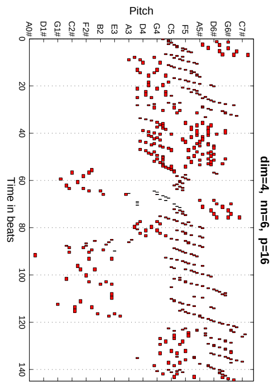
Figure B-3: **Selection of generated sequences for different model parameters. The training piece is Ligeti's Etude no.4 Book1.**

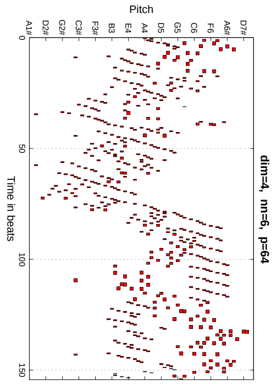

dim=2, nn=4, p=16
dim=2, nn=4, p=64
dim=2, nn=4, p=128
dim=4, nn=2, p=2
dim=4, nn=2, p=16
dim=4, nn=2, p=64
dim=4, nn=2, p=128
dim=4, nn=4, p=2
dim=4, nn=4, p=16
Pitch
Time in beats

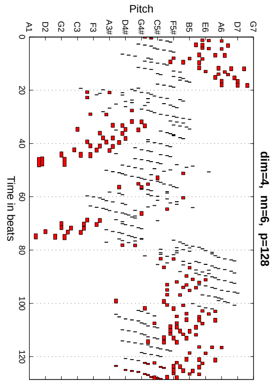

dim=4, nn=4, p=64

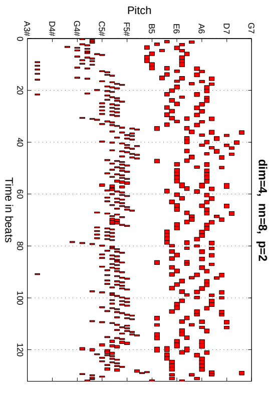

dim=4, nn=4, p=128

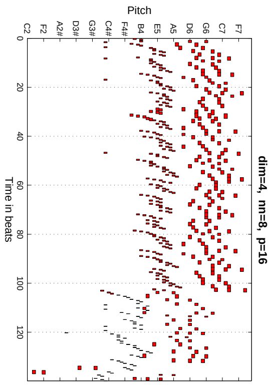

dim=4, nn=6, p=2

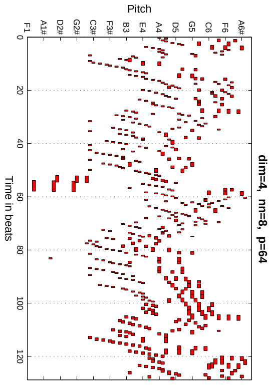

dim=4, nn=6, p=16

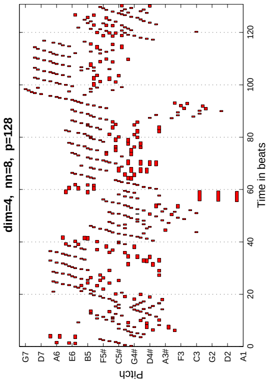

dim=4, nn=6, p=64

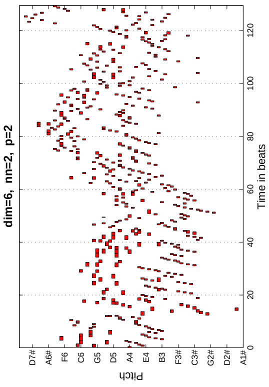

dim=4, nn=6, p=128

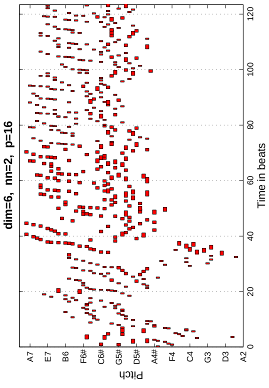

dim=4, nn=8, p=2

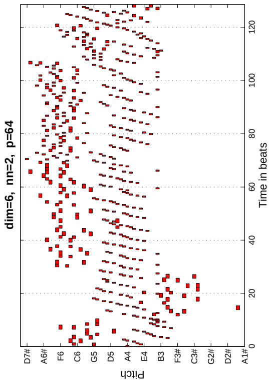

dim=4, nn=8, p=16

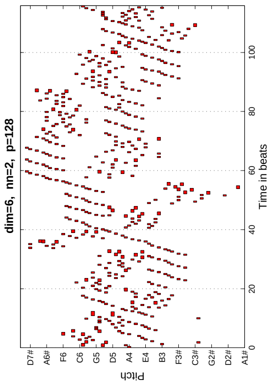

dim=4, nn=8, p=64

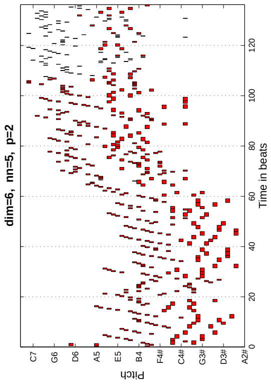

dim=4, nn=8, p=128
Pitch
G7
D7
A6
E6
B5
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
A1
Time in beats
0
20
40
60
80
100
120

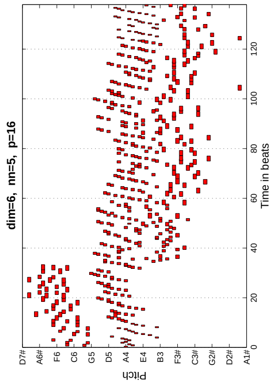

dim=6, nn=2, p=2
Pitch
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
20
40
60
80
100
120

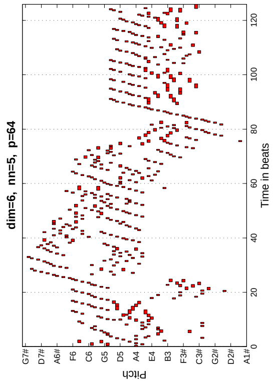

dim=6, nn=2, p=16
Pitch
A7
E7
B6
F6#
C6#
G5#
D5#
A4#
F4
C4
G3
D3
A2
Time in beats
0
20
40
60
80
100
120

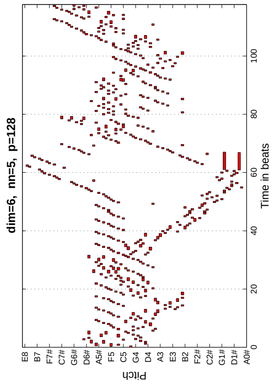

dim=6, nn=2, p=64
Pitch
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
20
40
60
80
100
120

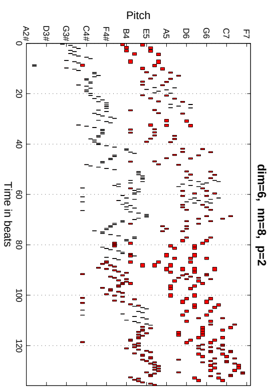

dim=6, nn=2, p=128
Pitch
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
20
40
60
80
100

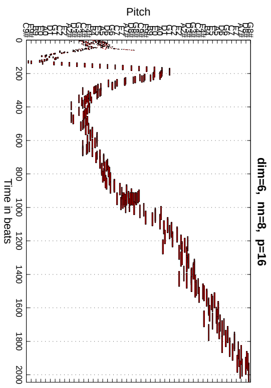

dim=6, nn=5, p=2
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
20
40
60
80
100
120

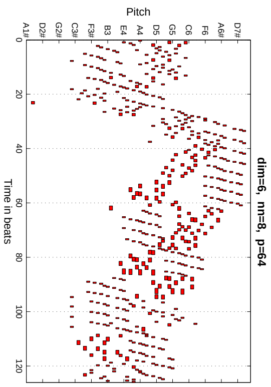

dim=6, nn=5, p=16
Pitch
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
20
40
60
80
100
120

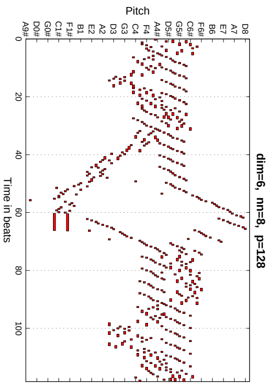

dim=6, nn=5, p=64
Pitch
G7#
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
20
40
60
80
100
120

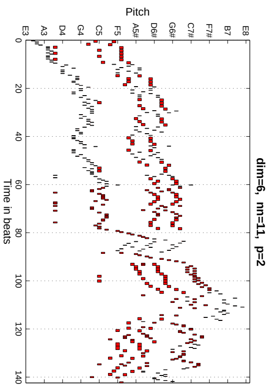

dim=6, nn=5, p=128
Pitch
E8
B7
F7#
C7#
G6#
D6#
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
D1#
A0#
Time in beats
0
20
40
60
80
100

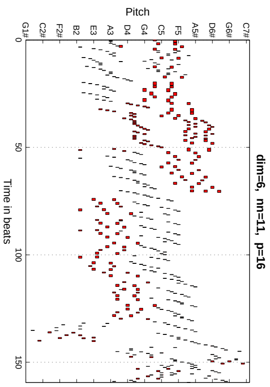

dim=6, nn=8, p=2
Pitch
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
C7
F7
Time in beats
0
20
40
60
80
100
120

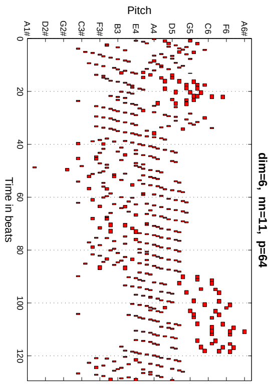

dim=6, nn=8, p=16
Pitch
Time in beats
0
200
400
600
800
1000
1200
1400
1600
1800
2000

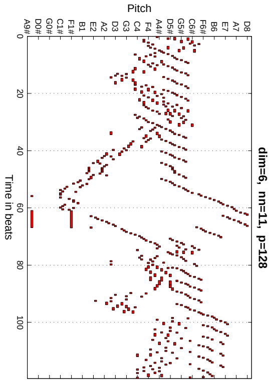

dim=6, nn=8, p=64
Pitch
A1#
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
F6
A6#
D7#
Time in beats
0
20
40
60
80
100
120

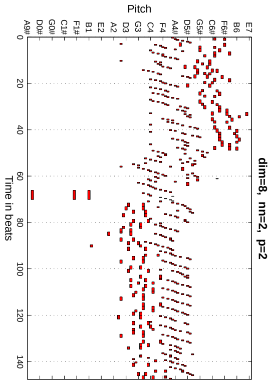

dim=6, nn=8, p=128
Pitch
A9#
D0#
G0#
C1#
F1#
B1
E2
A2
D3
G3
C4
F4
A4#
D5#
G5#
C6#
F6#
B6
E7
A7
D8
Time in beats
0
20
40
60
80
100

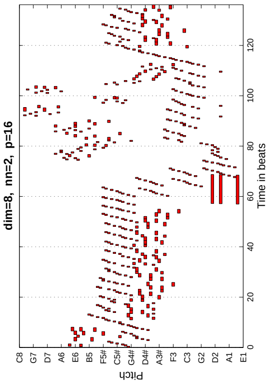

dim=6, nn=11, p=2
Pitch
E3
A3
D4
G4
C5
F5
A5#
D6#
G6#
C7#
F7#
B7
E8
Time in beats
0
20
40
60
80
100
120
140

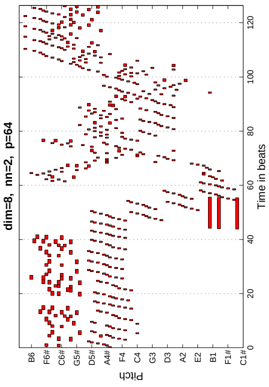

dim=6, nn=11, p=16
Pitch
G1#
C2#
F2#
B2
E3
A3
D4
G4
C5
F5
A5#
D6#
G6#
C7#
Time in beats
0
50
100
150

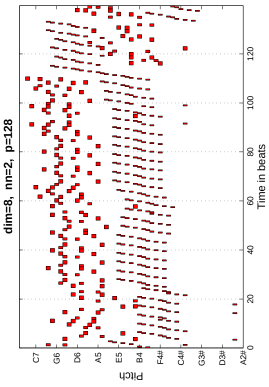

dim=6, nn=11, p=64
Pitch
A1#
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
F6
A6#
Time in beats
0
20
40
60
80
100
120

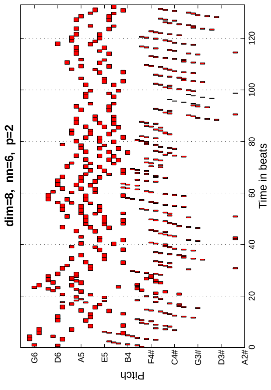

dim=6, nn=11, p=128
Pitch
A9#
D0#
G0#
C1#
F1#
B1
E2
A2
D3
G3
C4
F4
A4#
D5#
G5#
C6#
F6#
B6
E7
A7
D8
Time in beats
0
20
40
60
80
100

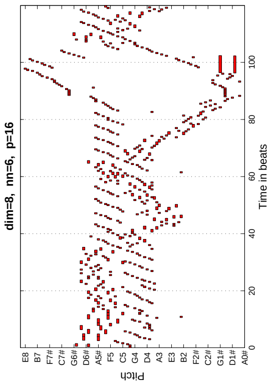

dim=8, nn=2, p=2
Pitch
A9#
D0#
G0#
C1#
F1#
B1
E1
A2
D3
G3
C4
F4
A4#
D5#
G5#
C6#
F6#
B6
E7
Time in beats
0
20
40
60
80
100
120
140

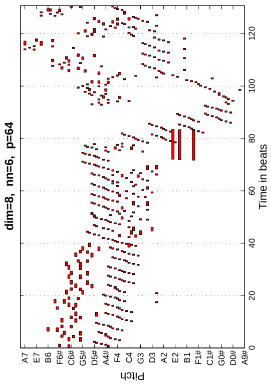

dim=8, nn=2, p=16

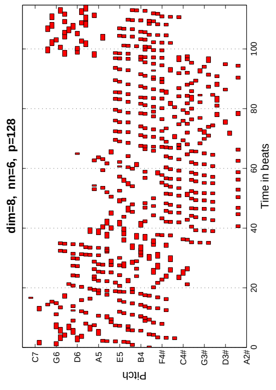

dim=8, nn=2, p=64

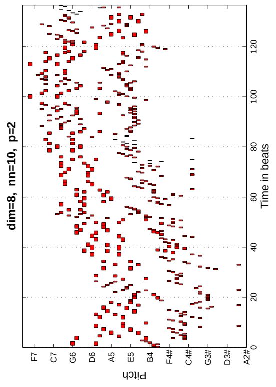

dim=8, nn=2, p=128

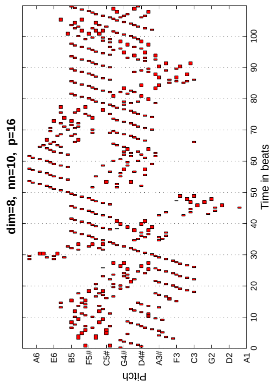

dim=8, nn=6, p=2

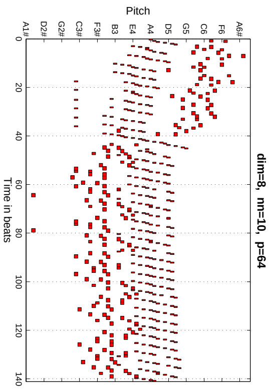

dim=8, nn=6, p=16

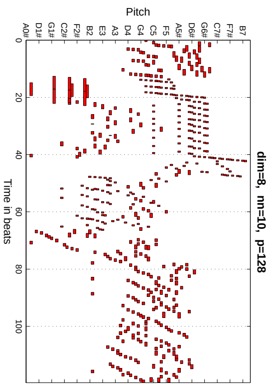

dim=8, nn=6, p=64

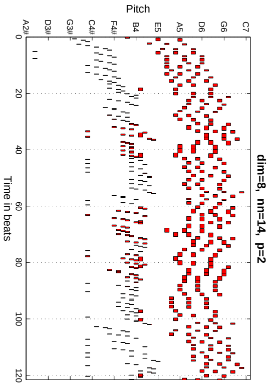

dim=8, nn=6, p=128

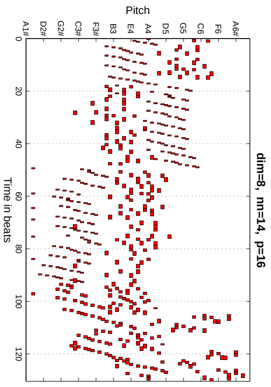

dim=8, nn=10, p=2

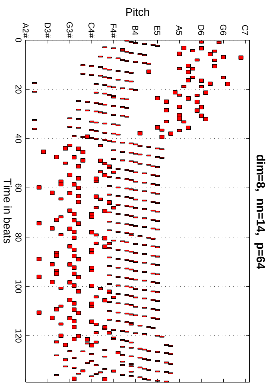

dim=8, nn=10, p=16

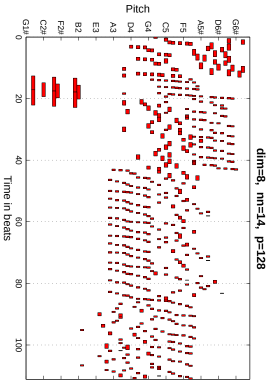

dim=8, nn=10, p=64
Pitch
A1# D2# G2# C3# F3# B3 E4 A4 D5 G5 C6 F6 A6#
Time in beats
0 20 40 60 80 100 120 140


dim=8, nn=10, p=128
Pitch
A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5# D6# G6# C7# F7# B7
Time in beats
0 20 40 60 80 100


dim=8, nn=14, p=2
Pitch
A2# D3# G3# C4# F4# B4 E5 A5 D6 G6 C7
Time in beats
0 20 40 60 80 100 120


dim=8, nn=14, p=16
Pitch
A1# D2# G2# C3# F3# B3 E4 A4 D5 G5 C6 F6 A6#
Time in beats
0 20 40 60 80 100 120


dim=8, nn=14, p=64
Pitch
A2# D3# G3# C4# F4# B4 E5 A5 D6 G6 C7
Time in beats
0 20 40 60 80 100 120


dim=8, nn=14, p=128
Pitch
G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5# D6# G6#
Time in beats
0 20 40 60 80 100

B.4 Combination of Etudes 4 and 6, Method 1

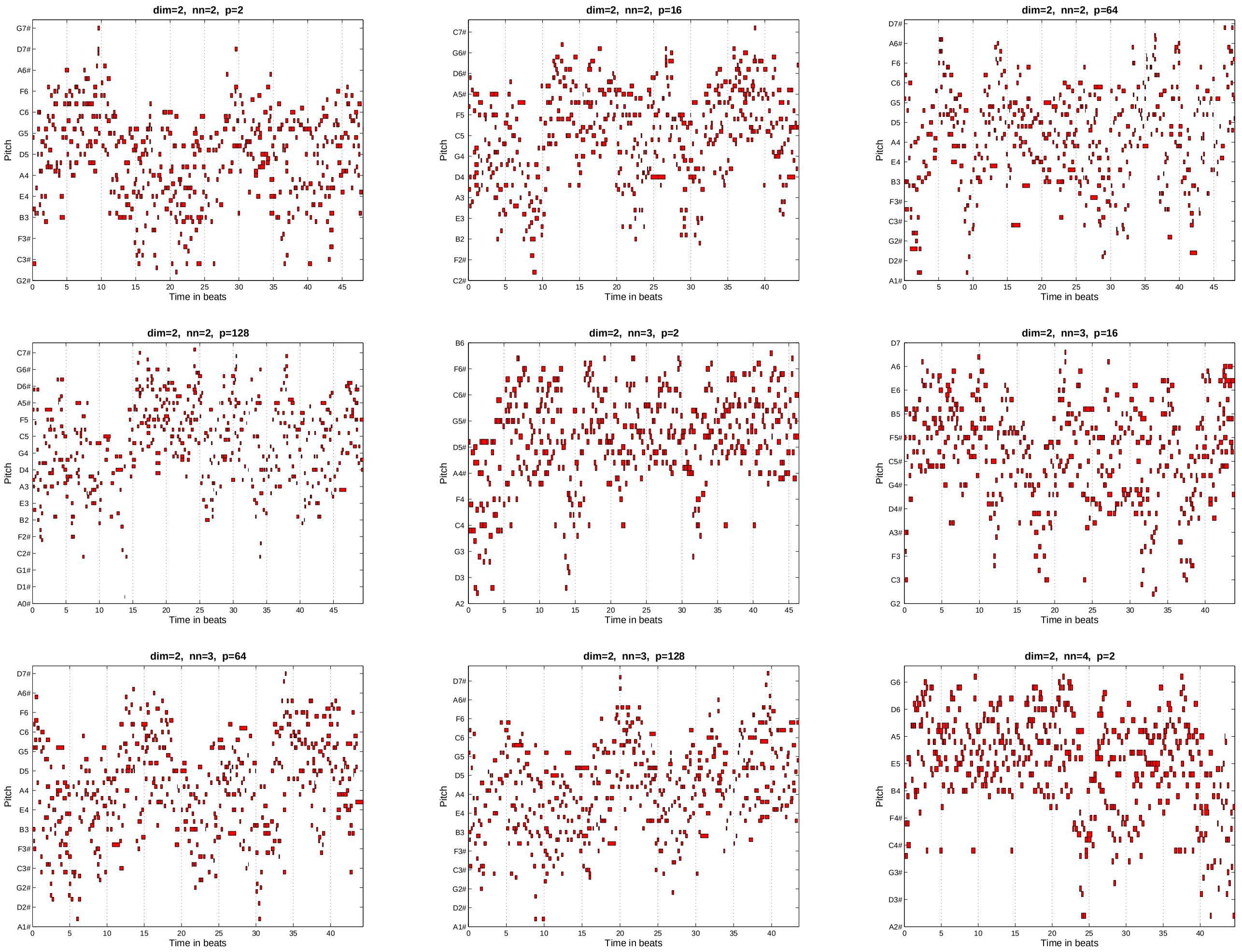
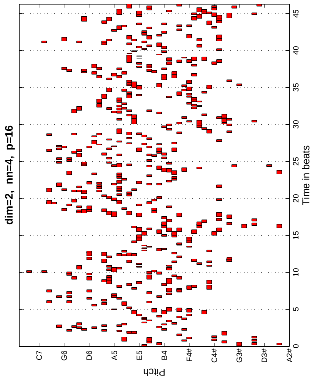
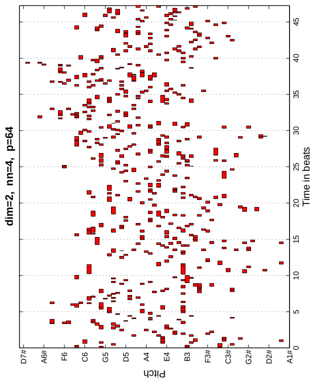
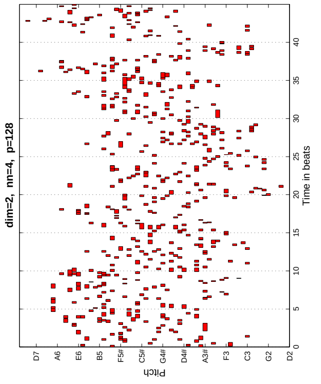
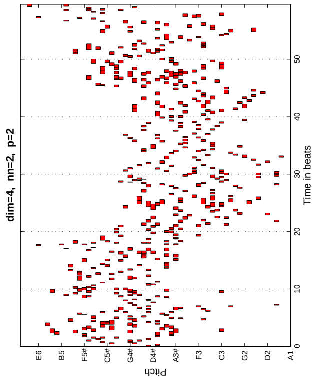


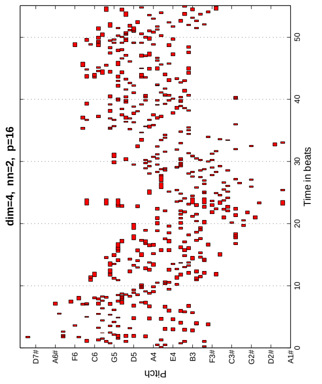
Figure B-4: **500 note fragments from combinations of Ligeti's Etudes 4 and 6 Book 1 using *method 1*, with 20% for Etude 4 and 80% for Etude 6. Each panel is a combination of the Etudes using different parameter values.**

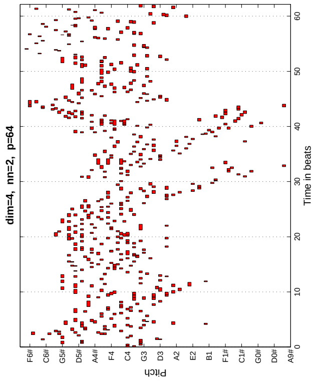

dim=2, nn=4, p=16

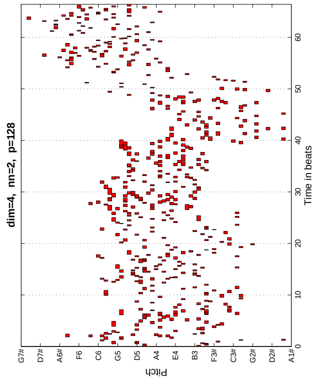

dim=2, nn=4, p=64

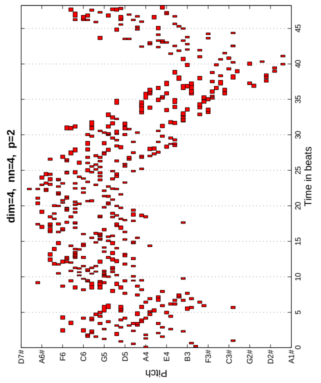

dim=2, nn=4, p=128

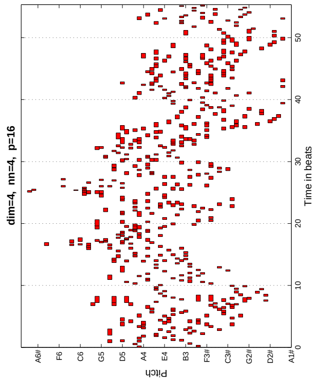

dim=4, nn=2, p=2

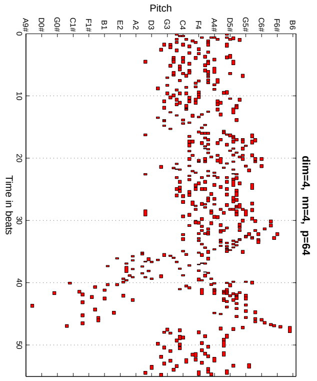

dim=4, nn=2, p=16

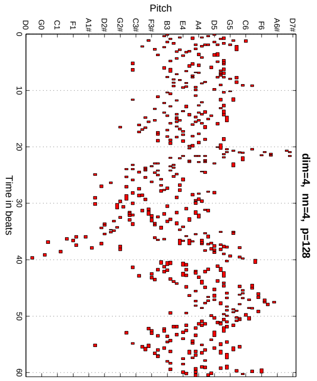

dim=4, nn=2, p=64

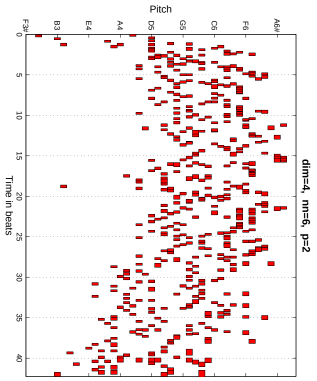

dim=4, nn=2, p=128

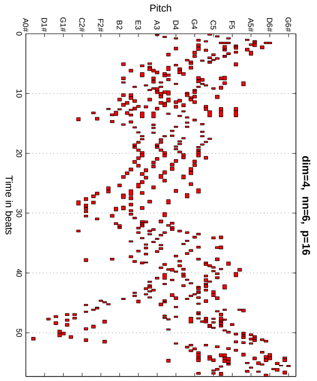

dim=4, nn=4, p=2

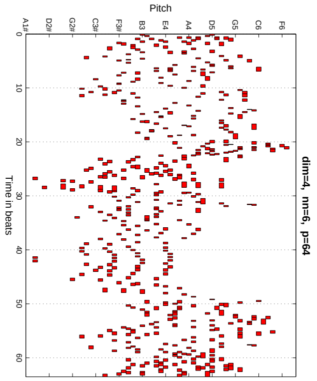

dim=4, nn=4, p=16

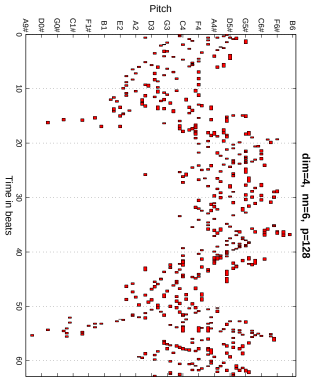

dim=4, nn=4, p=64
Pitch
Time in beats

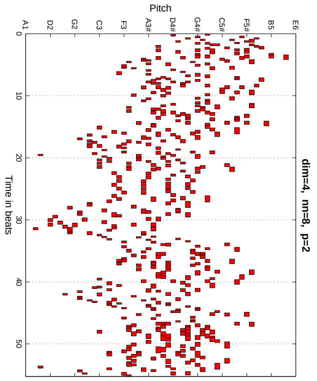

dim=4, nn=4, p=128
Pitch
Time in beats

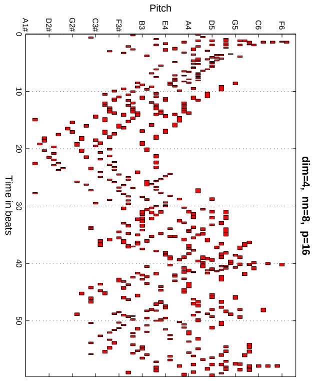

dim=4, nn=6, p=2
Pitch
Time in beats

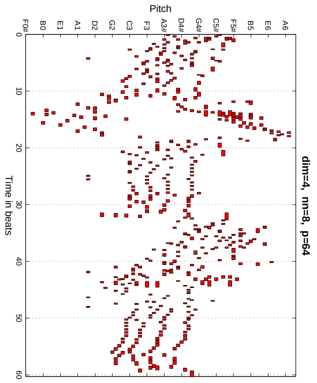

dim=4, nn=6, p=16
Pitch
Time in beats

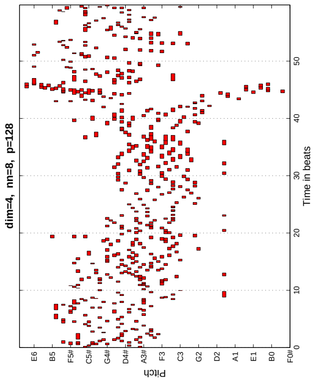

dim=4, nn=6, p=64
Pitch
Time in beats

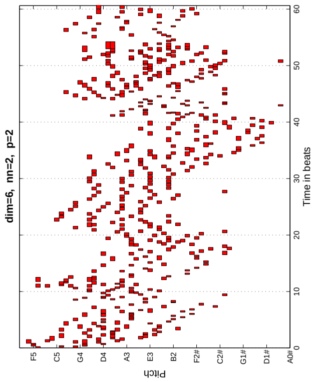

dim=4, nn=6, p=128
Pitch
Time in beats

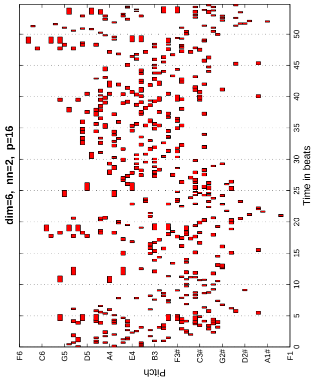

dim=4, nn=8, p=2
Pitch
Time in beats

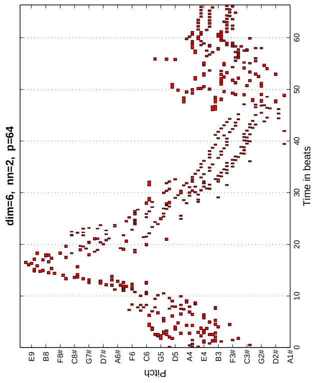

dim=4, nn=8, p=16
Pitch
Time in beats

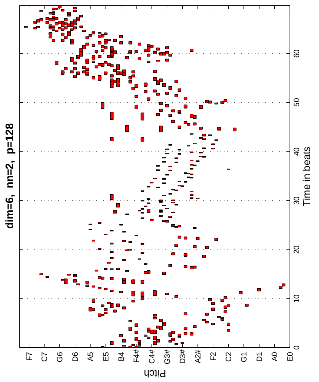

dim=4, nn=8, p=64
Pitch
Time in beats

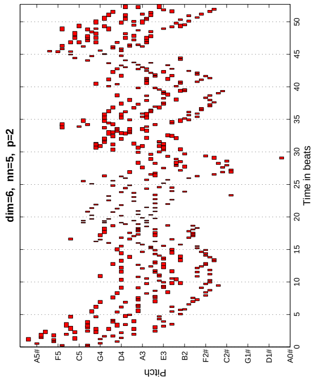

dim=4, nn=8, p=128
Pitch
E6
B5
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
A1
E1
B0
F0#
Time in beats
0
10
20
30
40
50

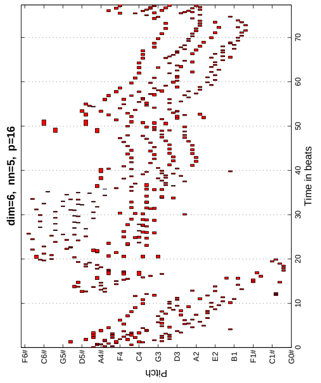

dim=6, nn=2, p=2
Pitch
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
D1#
A0#
Time in beats
0
10
20
30
40
50
60

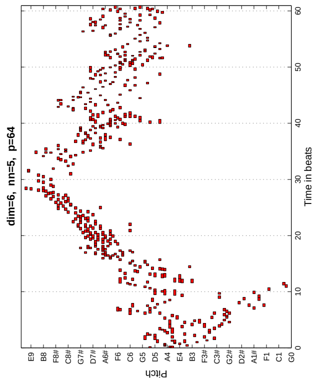

dim=6, nn=2, p=16
Pitch
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
F1
Time in beats
0
5
10
15
20
25
30
35
40
45
50

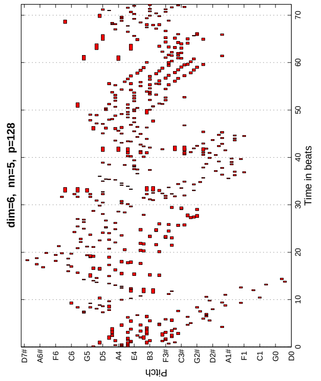

dim=6, nn=2, p=64
Pitch
E9
B8
F8#
C8#
G7#
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
10
20
30
40
50
60

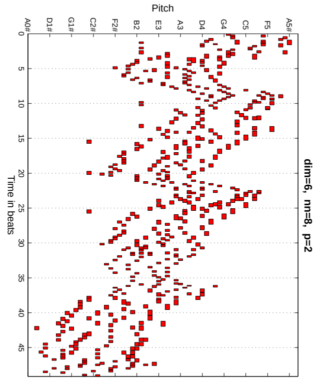

dim=6, nn=2, p=128
Pitch
F7
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
F2
C2
G1
D1
A0
E0
Time in beats
10
20
30
40
50
60

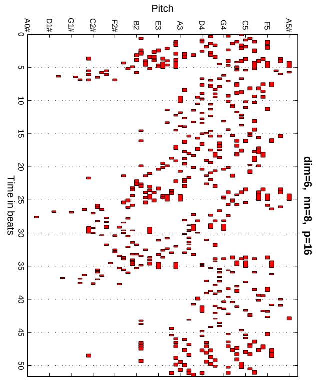

dim=6, nn=5, p=2
Pitch
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
D1#
A0#
Time in beats
0
5
10
15
20
25
30
35
40
45
50

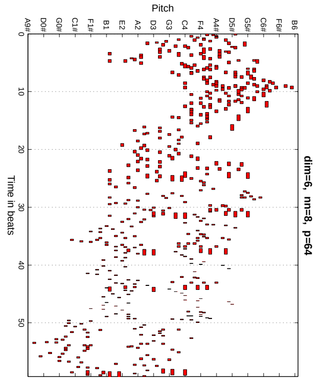

dim=6, nn=5, p=16
Pitch
F6#
C6#
G5#
D5#
A4#
F4
C4
G3
D3
A2
E2
B1
F1#
C1#
G0#
Time in beats
0
10
20
30
40
50
60
70

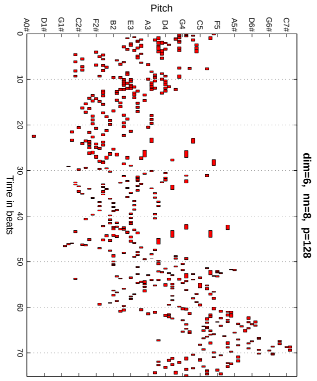

dim=6, nn=5, p=64
Pitch
E9
B8
F8#
C8#
G7#
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
F1
C1
G0
Time in beats
0
10
20
30
40
50
60

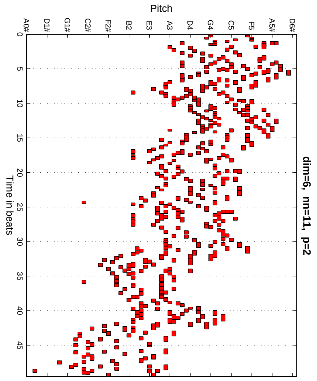

dim=6, nn=5, p=128
Pitch
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
F1
C1
G0
D0
Time in beats
0
10
20
30
40
50
60
70

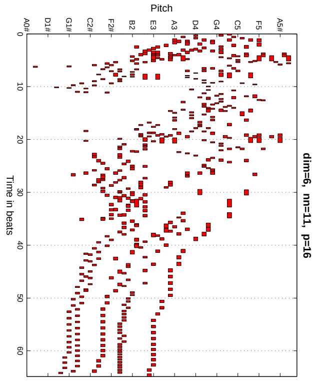

dim=6, nn=8, p=2
Pitch
A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5#
Time in beats
0 5 10 15 20 25 30 35 40 45

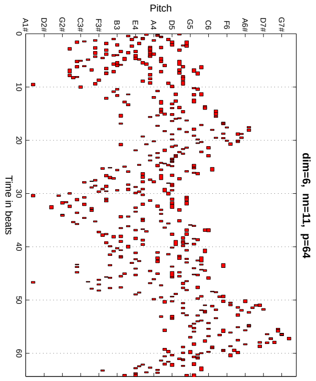

dim=6, nn=8, p=16
Pitch
A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5#
Time in beats
0 5 10 15 20 25 30 35 40 45 50

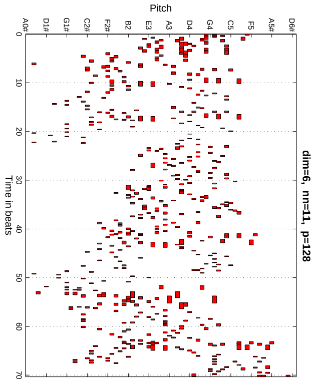

dim=6, nn=8, p=64
Pitch
A9# D0# G0# C1# F1# B1 E2 A2 D3 G3 C4 F4 A4# D5# G5# C6# F6# B6
Time in beats
0 10 20 30 40 50

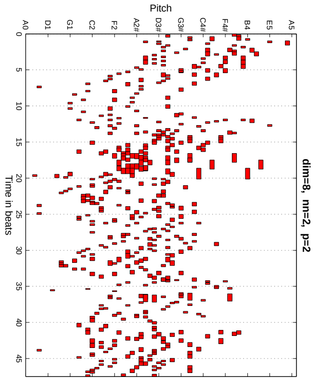

dim=6, nn=8, p=128
Pitch
A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5# D6# G6# C7#
Time in beats
0 10 20 30 40 50 60 70

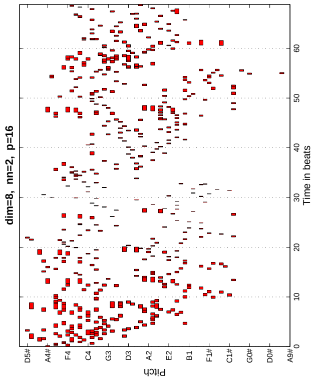

dim=6, nn=11, p=2
Pitch
A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5# D6#
Time in beats
0 5 10 15 20 25 30 35 40 45

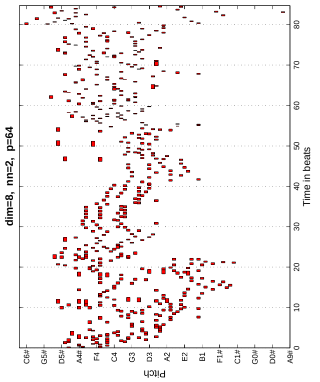

dim=6, nn=11, p=16
Pitch
A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5#
Time in beats
0 10 20 30 40 50 60

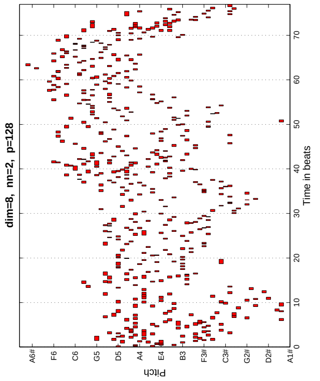

dim=6, nn=11, p=64
Pitch
A1# D2# G2# C3# F3# B3 E4 A4 D5 G5 C6 F6 A6# D7# G7#
Time in beats
0 10 20 30 40 50 60

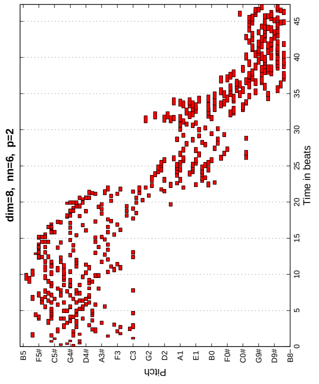

dim=6, nn=11, p=128
Pitch
A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5# D6#
Time in beats
0 10 20 30 40 50 60 70

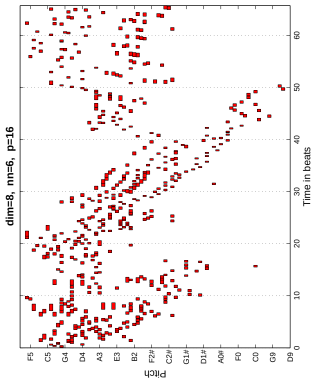

dim=8, nn=2, p=2
Pitch
A0 D1 G1 C2 F2 A2# D3# G3# C4# F4# B4 E5 A5
Time in beats
0 5 10 15 20 25 30 35 40 45

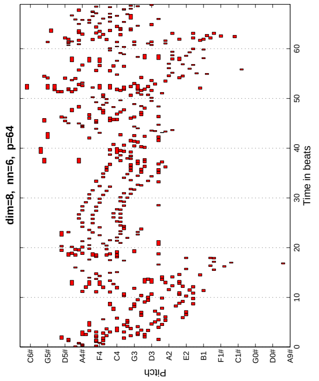

dim=8, nn=2, p=16
Pitch
D5# A4# F4 C4 G3 D3 A2 E2 B1 F1# C1# G0# D0# A9#
Time in beats
0 10 20 30 40 50 60

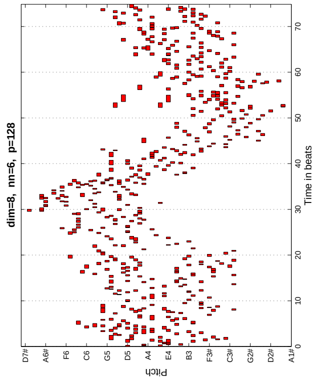

dim=8, nn=2, p=64
Pitch
C6# G5# D5# A4# F4 C4 G3 D3 A2 E2 B1 F1# C1# G0# D0# A9#
Time in beats
0 10 20 30 40 50 60 70 80

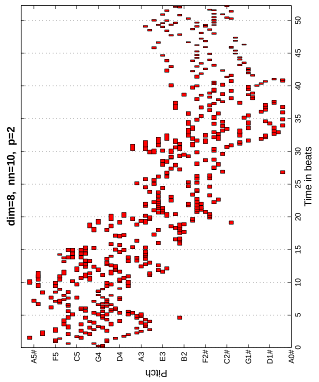

dim=8, nn=2, p=128
Pitch
A6# F6 C6 G5 D5 A4 E4 B3 F3# C3# G2# D2# A1#
Time in beats
0 10 20 30 40 50 60 70

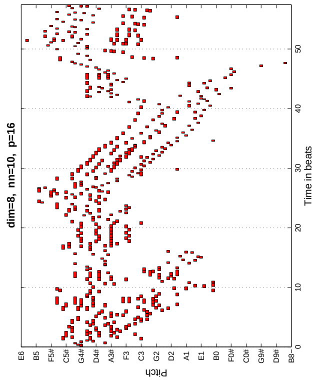

dim=8, nn=6, p=2
Pitch
B5 F5# C5# G4# D4# A3# F3 C3 G2 D2 A1 E1 B0 F0# C0# G9# D9# B8-
Time in beats
0 5 10 15 20 25 30 35 40 45

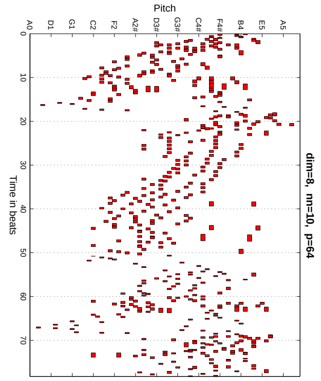

dim=8, nn=6, p=16
Pitch
F5 C5 G4 D4 A3 E3 B2 F2# C2# G1# D1# A0# F0 C0 G9 D9
Time in beats
0 10 20 30 40 50 60

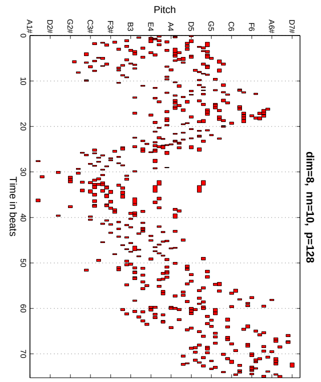

dim=8, nn=6, p=64
Pitch
C6# G5# D5# A4# F4 C4 G3 D3 A2 E2 B1 F1# C1# G0# D0# A9#
Time in beats
0 10 20 30 40 50 60

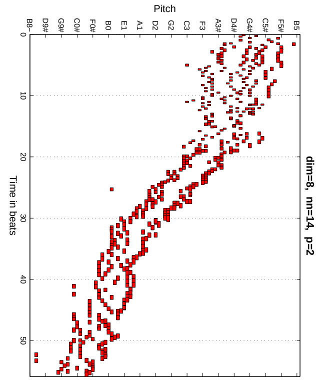

dim=8, nn=6, p=128
Pitch
D7# A6# F6 C6 G5 D5 A4 E4 B3 F3# C3# G2# D2# A1#
Time in beats
0 10 20 30 40 50 60 70

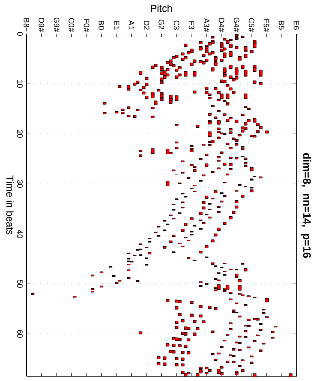

dim=8, nn=10, p=2
Pitch
A5# F5 C5 G4 D4 A3 E3 B2 F2# C2# G1# D1# A0#
Time in beats
0 5 10 15 20 25 30 35 40 45 50

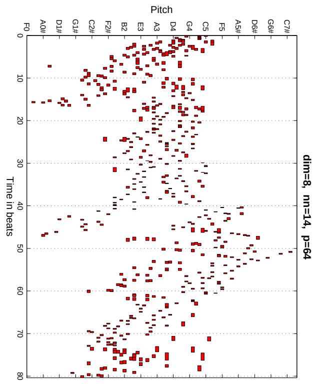

dim=8, nn=10, p=16
Pitch
E6 B5 F5# C5# G4# D4# A3# F3 C3 G2 D2 A1 E1 B0 F0# C0# G9# D9# B8-
Time in beats
0 10 20 30 40 50

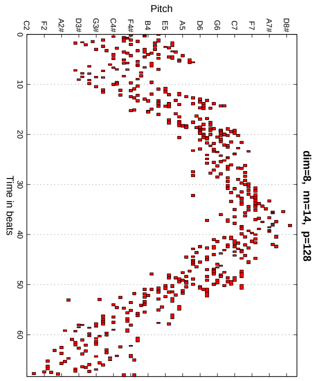

dim=8, nn=10, p=64
Pitch
A0
D1
G1
C2
F2
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
Time in beats
0
10
20
30
40
50
60
70

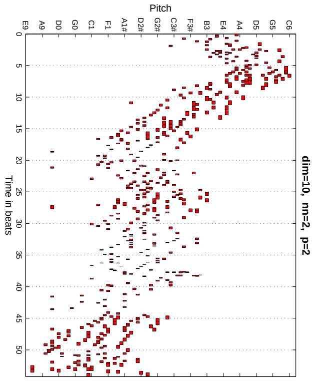

dim=8, nn=10, p=128
Pitch
A1#
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
F6
A6#
D7#
Time in beats
0
10
20
30
40
50
60
70

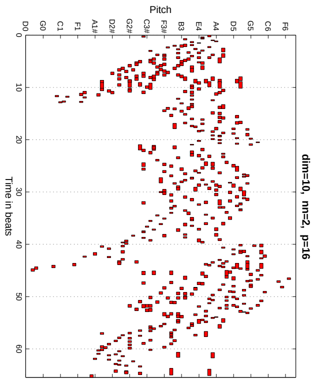

dim=8, nn=14, p=2
Pitch
B8-
D9#
G9#
C0#
F0#
B0
E1
A1
D2
G2
C3
F3
A3#
D4#
G4#
C5#
F5#
B5
Time in beats
0
10
20
30
40
50

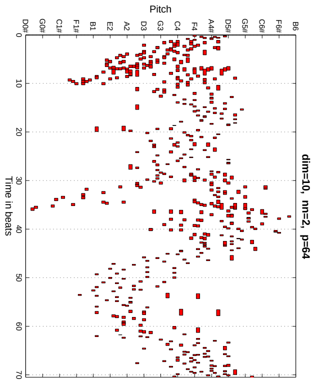

dim=8, nn=14, p=16
Pitch
B8-
D9#
G9#
C0#
F0#
B0
E1
A1
D2
G2
C3
F3
A3#
D4#
G4#
C5#
F5#
B5
E6
Time in beats
0
10
20
30
40
50
60

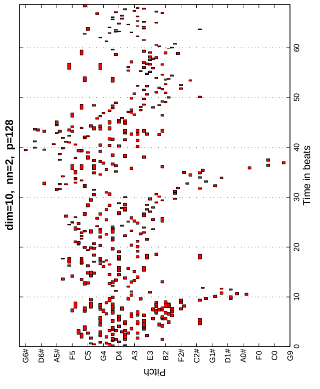

dim=8, nn=14, p=64
Pitch
F0
A0#
D1#
G1#
C2#
F2#
B2
E3
A3
D4
G4
C5
F5
A5#
D6#
G6#
C7#
Time in beats
0
10
20
30
40
50
60
70
80

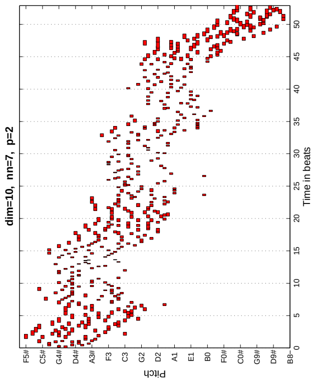

dim=8, nn=14, p=128
Pitch
C2
F2
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
C7
F7
A7#
D8#
Time in beats
0
10
20
30
40
50
60

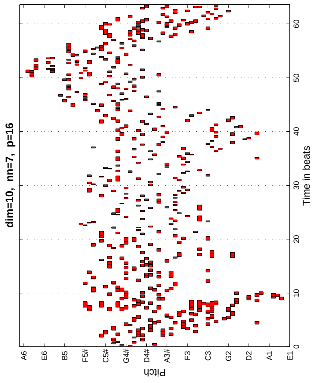

dim=10, nn=2, p=2
Pitch
E8
A9
D0
G0
C1
F1
A1#
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
Time in beats
0
5
10
15
20
25
30
35
40
45
50

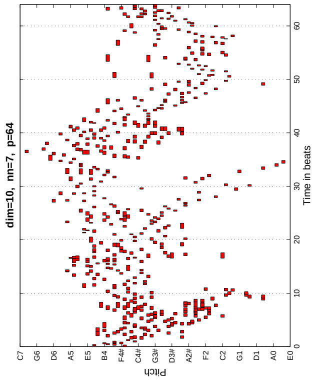

dim=10, nn=2, p=16
Pitch
D0
G0
C1
F1
A1#
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
F6
Time in beats
0
10
20
30
40
50
60

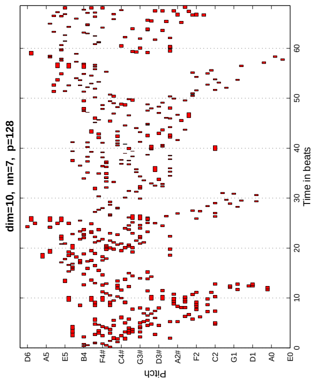

dim=10, nn=2, p=64
Pitch
D0#
G0#
C1#
F1#
B1
E2
A2
D3
G3
C4
F4
A4#
D5#
G5#
C6#
F6#
B6
Time in beats
0
10
20
30
40
50
60
70

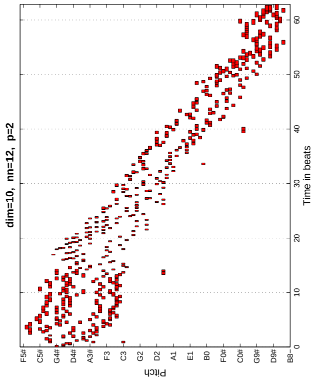

dim=10, nn=2, p=128
Pitch
G6#
D6#
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
D1#
A0#
F0
C0
G9
Time in beats
0
10
20
30
40
50
60

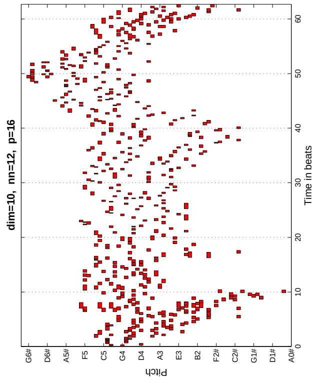

dim=10, nn=7, p=2
Pitch
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
A1
E1
B0
F0#
C0#
G9#
D9#
B8
Time in beats
0
5
10
15
20
25
30
35
40
45
50

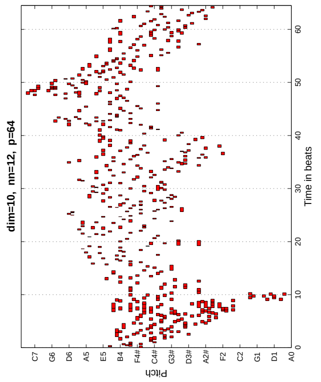

dim=10, nn=7, p=16
Pitch
A6
E6
B5
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
A1
E1
Time in beats
0
10
20
30
40
50
60

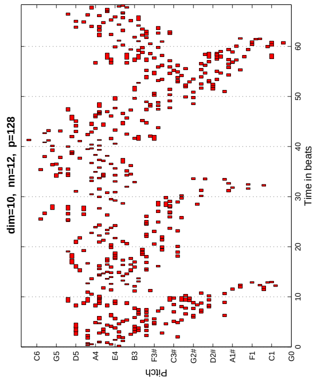

dim=10, nn=7, p=64
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
F2
C2
G1
D1
A0
E0
Time in beats
0
10
20
30
40
50
60

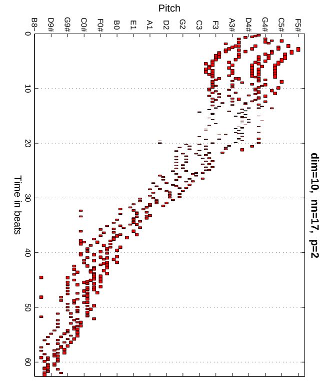

dim=10, nn=7, p=128
Pitch
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
F2
C2
G1
D1
A0
E0
Time in beats
0
10
20
30
40
50
60

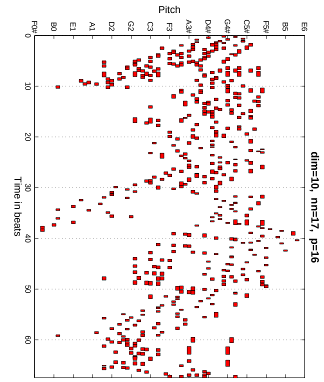

dim=10, nn=12, p=2
Pitch
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
A1
E1
B0
F0#
C0#
G9#
D9#
B8
Time in beats
0
10
20
30
40
50
60

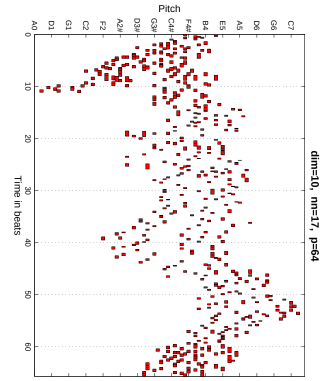

dim=10, nn=12, p=16
Pitch
G6#
D6#
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
D1#
A0#
Time in beats
0
10
20
30
40
50
60

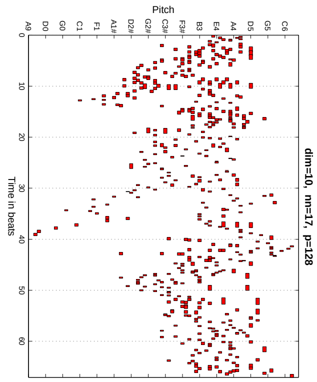

dim=10, nn=12, p=64
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
F2
C2
G1
D1
A0
Time in beats
0
10
20
30
40
50
60


dim=10, nn=12, p=128
Pitch
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
F1
C1
G0
Time in beats
0
10
20
30
40
50
60


dim=10, nn=17, p=2
Pitch
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
A1
E1
B0
F0#
C0#
G9#
D9#
B8
Time in beats
0
10
20
30
40
50
60


dim=10, nn=17, p=16
Pitch
E6
B5
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
A1
E1
B0
F0#
Time in beats
0
10
20
30
40
50
60


dim=10, nn=17, p=64
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
F2
C2
G1
D1
A0
Time in beats
0
10
20
30
40
50
60


dim=10, nn=17, p=128
Pitch
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
F1
C1
G0
D0
A9
Time in beats
0
10
20
30
40
50
60

B.5 Combination of Etudes 4 and 6, Method 2

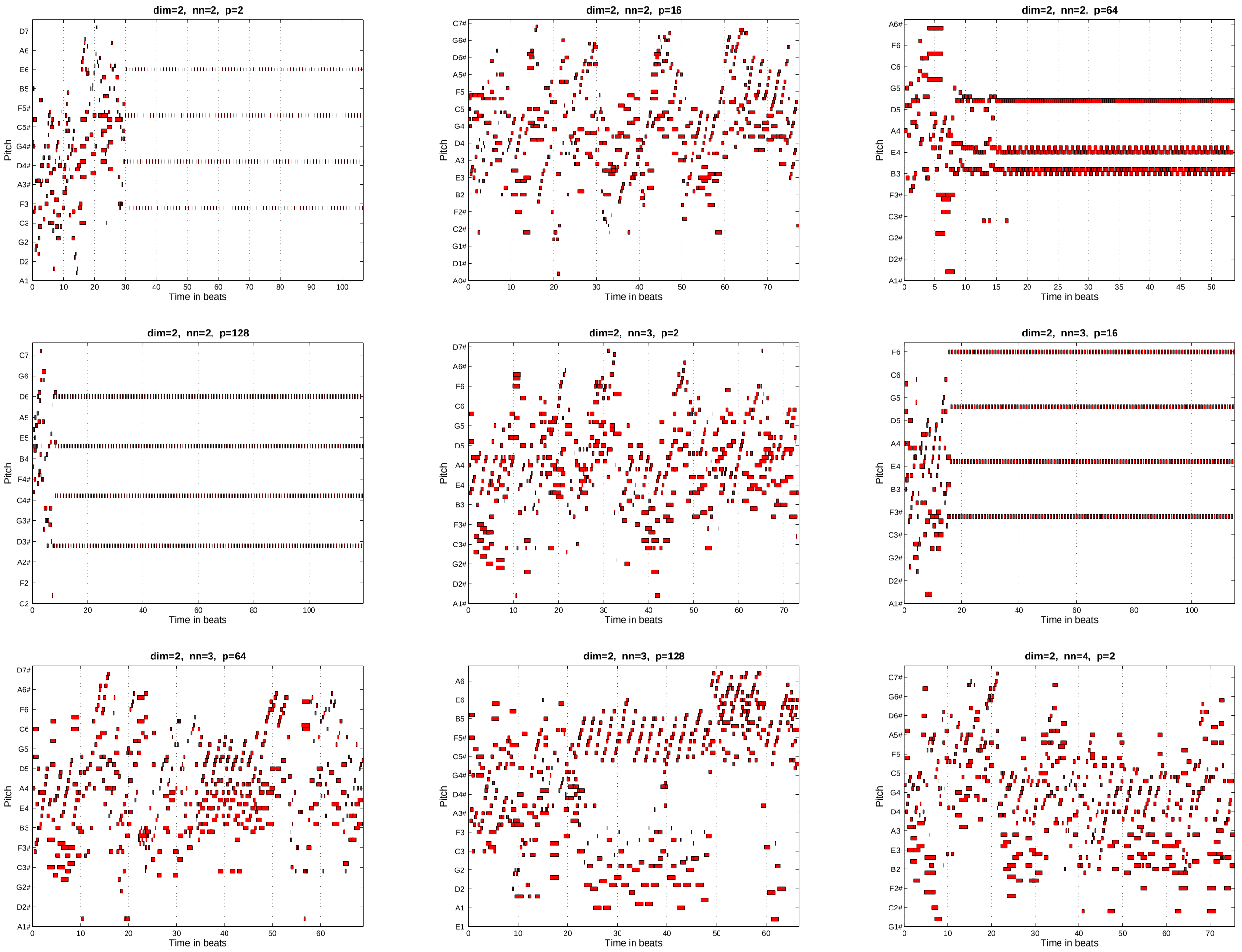
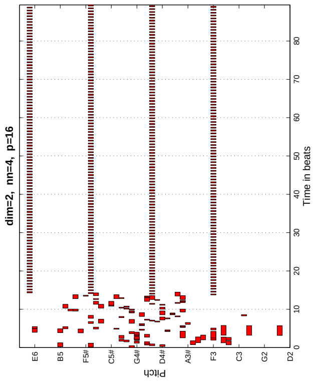
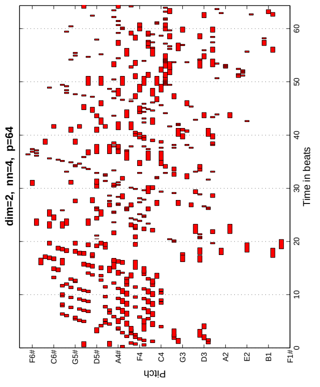
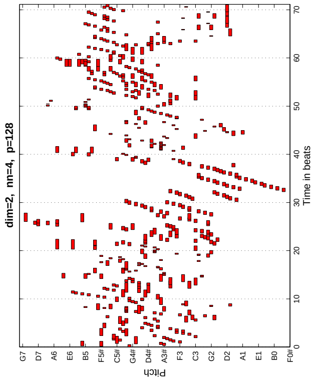
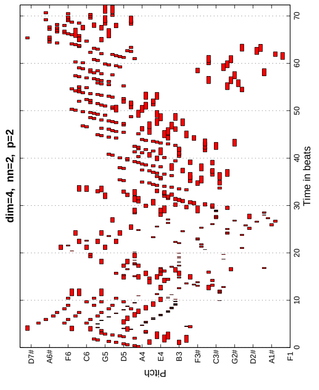


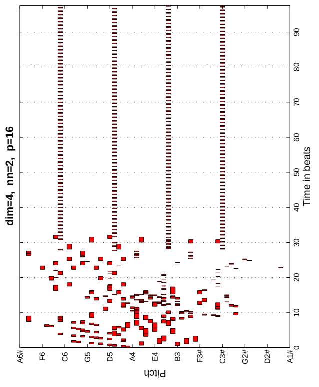
Figure B-5: 500 note fragments from combinations of Ligeti's Etudes 4 and 6 Book 1 using *method 2*. Each panel is a combination of the Etudes using different parameter values.

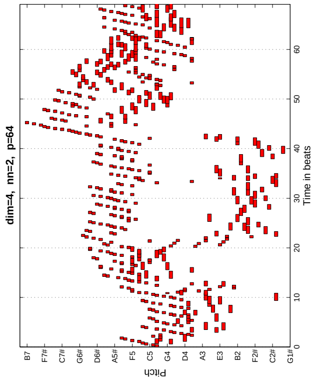

dim=2, nn=4, p=16
Pitch
E6
B5
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
Time in beats
0
10
20
30
40
50
60
70
80

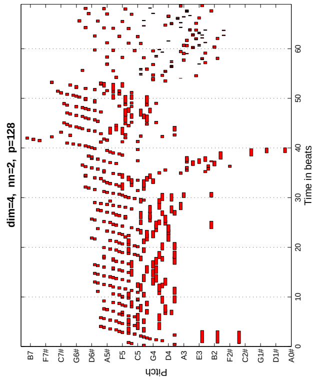

dim=2, nn=4, p=64
Pitch
F6#
C6#
G5#
D5#
A4#
F4
C4
G3
D3
A2
E2
B1
F1#
Time in beats
0
10
20
30
40
50
60

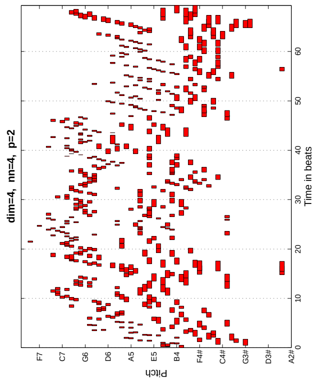

dim=2, nn=4, p=128
Pitch
G7
D7
A6
E6
B5
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
A1
E1
B0
F0#
Time in beats
0
10
20
30
40
50
60
70

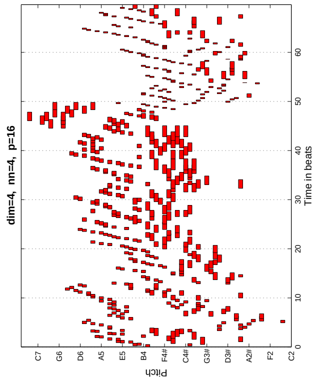

dim=4, nn=2, p=2
Pitch
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
F1
Time in beats
0
10
20
30
40
50
60
70

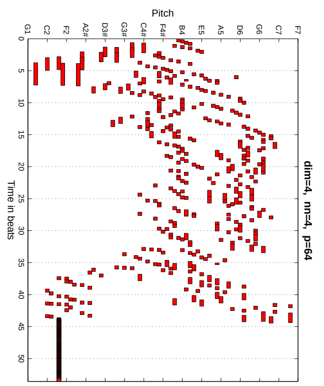

dim=4, nn=2, p=16
Pitch
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
10
20
30
40
50
60
70
80
90

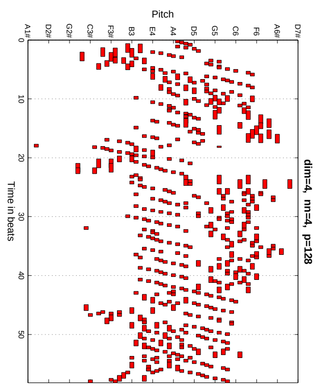

dim=4, nn=2, p=64
Pitch
B7
F7#
C7#
G6#
D6#
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
Time in beats
0
10
20
30
40
50
60

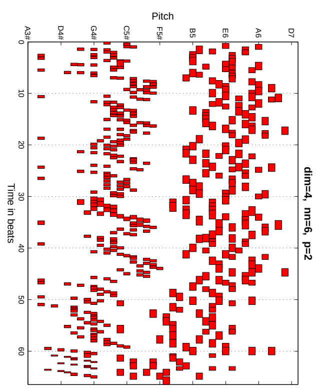

dim=4, nn=2, p=128
Pitch
B7
F7#
C7#
G6#
D6#
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
D1#
A0#
Time in beats
0
10
20
30
40
50
60

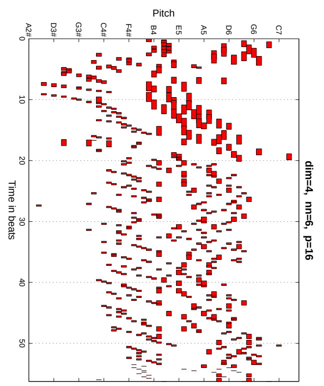

dim=4, nn=4, p=2
Pitch
F7
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
10
20
30
40
50
60

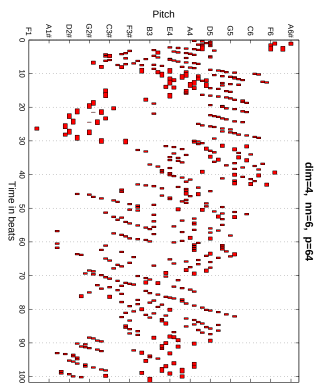

dim=4, nn=4, p=16
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
F2
C2
Time in beats
0
10
20
30
40
50
60

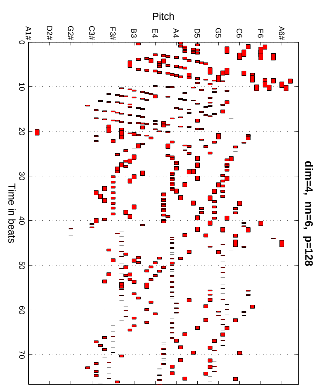

dim=4, nn=4, p=64
Pitch
G1
C2
F2
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
C7
F7
Time in beats
0
5
10
15
20
25
30
35
40
45
50

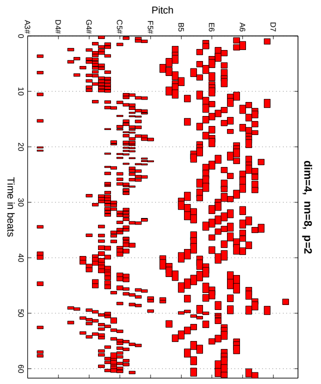

dim=4, nn=4, p=128
Pitch
A1#
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
F6
A6#
D7#
Time in beats
0
10
20
30
40
50

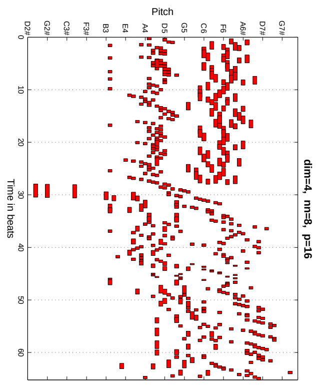

dim=4, nn=6, p=2
Pitch
A3#
D4#
G4#
C5#
F5#
B5
E6
A6
D7
Time in beats
0
10
20
30
40
50
60

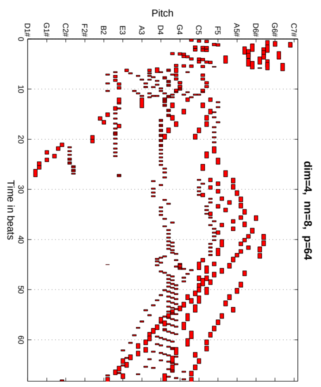

dim=4, nn=6, p=16
Pitch
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
C7
Time in beats
0
10
20
30
40
50

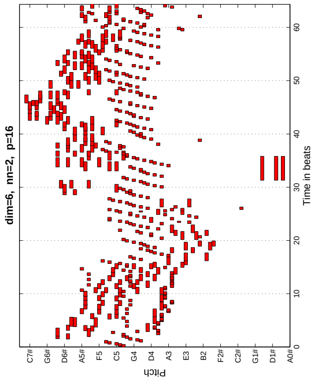

dim=4, nn=6, p=64
Pitch
F1
A1#
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
F6
A6#
Time in beats
0
10
20
30
40
50
60
70
80
90
100

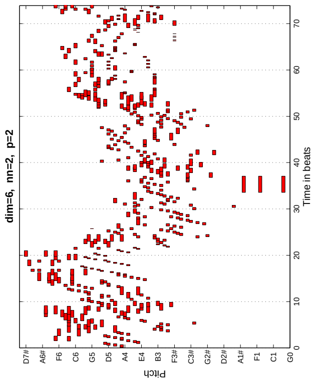

dim=4, nn=6, p=128
Pitch
A1#
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
F6
A6#
Time in beats
0
10
20
30
40
50
60
70

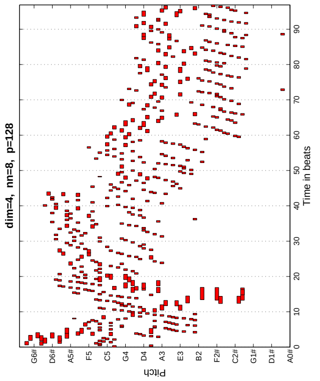

dim=4, nn=8, p=2
Pitch
A3#
D4#
G4#
C5#
F5#
B5
E6
A6
D7
Time in beats
0
10
20
30
40
50
60

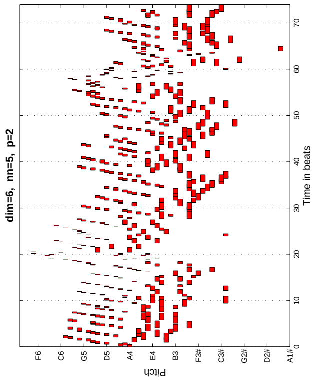

dim=4, nn=8, p=16
Pitch
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
F6
A6#
D7#
G7#
Time in beats
0
10
20
30
40
50
60

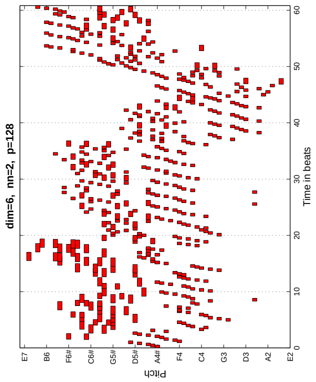

dim=4, nn=8, p=64
Pitch
D1#
G1#
C2#
F2#
B2
E3
A3
D4
G4
C5
F5
A5#
D6#
G6#
C7#
Time in beats
0
10
20
30
40
50
60

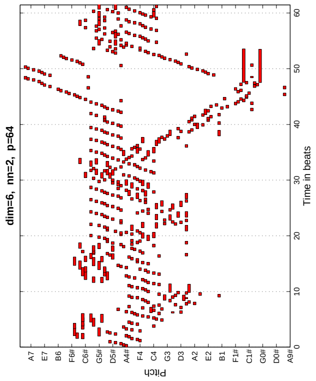

dim=6, nn=2, p=16

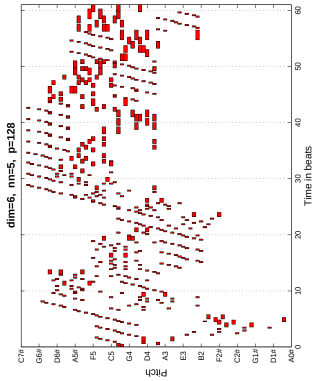

dim=6, nn=2, p=2

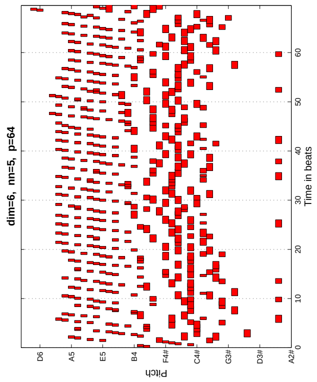

dim=4, nn=8, p=128

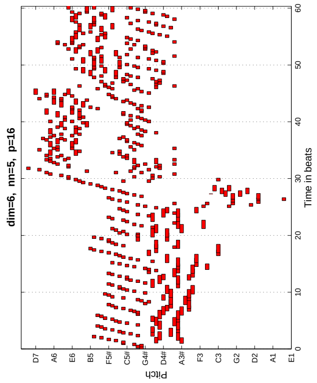

dim=6, nn=5, p=2

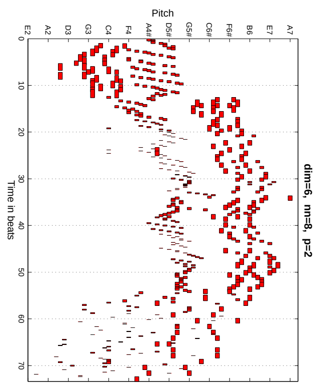

dim=6, nn=2, p=128

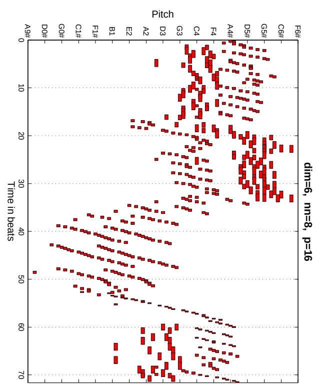

dim=6, nn=2, p=64

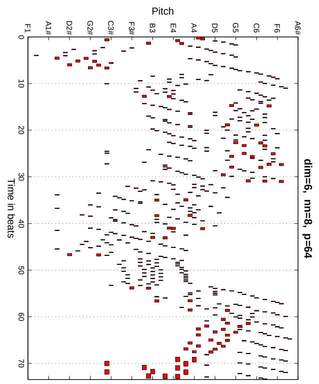

dim=6, nn=5, p=128

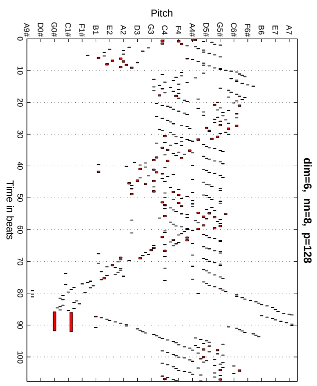

dim=6, nn=5, p=64

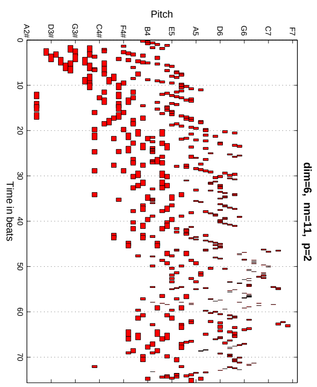

dim=6, nn=5, p=16

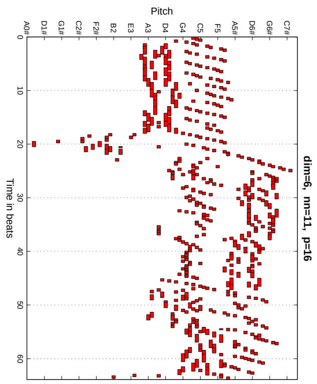

dim=6, nn=8, p=2
Pitch
E2 A2 D3 G3 C4 F4 A4# D5# G5# C6# F6# B6 E7 A7
Time in beats
0 10 20 30 40 50 60 70

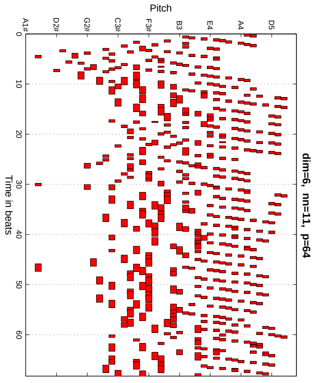

dim=6, nn=8, p=16
Pitch
A9# D0# G0# C1# F1# B1 E2 A2 D3 G3 C4 F4 A4# D5# G5# C6# F6#
Time in beats
0 10 20 30 40 50 60 70

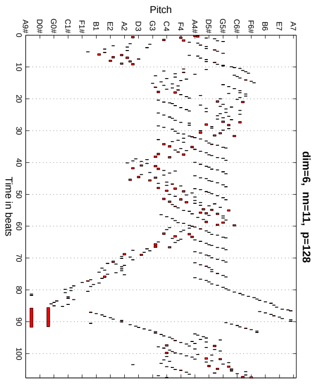

dim=6, nn=8, p=64
Pitch
F1 A1# D2# G2# C3# F3# B3 E4 A4 D5 G5 C6 F6 A6#
Time in beats
0 10 20 30 40 50 60 70

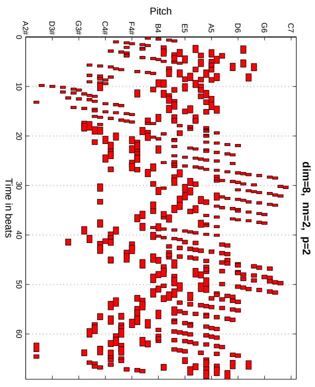

dim=6, nn=8, p=128
Pitch
A9# D0# G0# C1# F1# B1 E2 A2 D3 G3 C4 F4 A4# D5# G5# C6# F6# B6 E7 A7
Time in beats
0 10 20 30 40 50 60 70 80 90 100

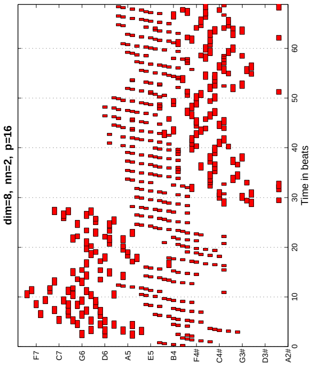

dim=6, nn=11, p=2
Pitch
A2# D3# G3# C4# F4# B4 E5 A5 D6 G6 C7 F7
Time in beats
0 10 20 30 40 50 60 70

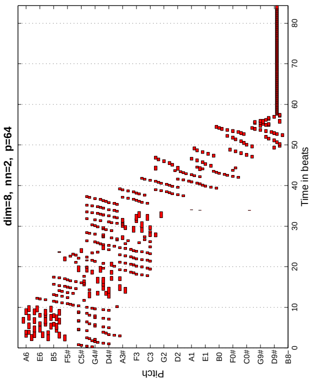

dim=6, nn=11, p=16
Pitch
A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5# D6# G6# C7#
Time in beats
0 10 20 30 40 50 60

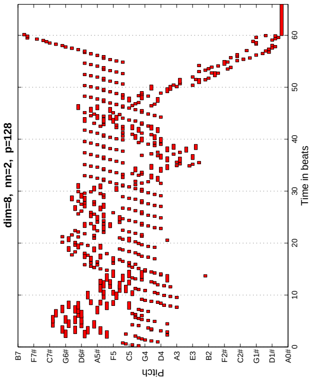

dim=6, nn=11, p=64
Pitch
A1# D2# G2# C3# F3# B3 E4 A4 D5
Time in beats
0 10 20 30 40 50 60

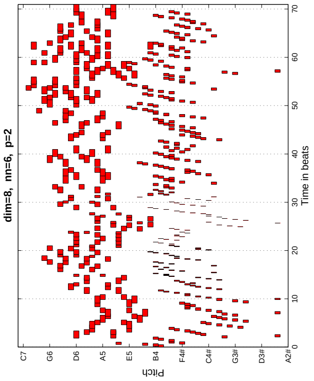

dim=6, nn=11, p=128
Pitch
A9# D0# G0# C1# F1# B1 E2 A2 D3 G3 C4 F4 A4# D5# G5# C6# F6# B6 E7 A7
Time in beats
0 10 20 30 40 50 60 70 80 90 100

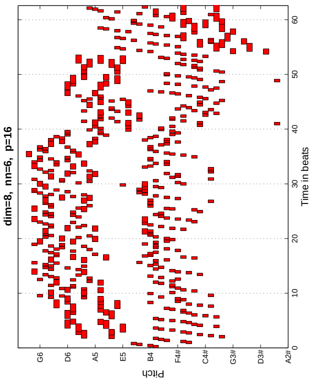

dim=8, nn=2, p=2
Pitch
A2# D3# G3# C4# F4# B4 E5 A5 D6 G6 C7
Time in beats
0 10 20 30 40 50 60

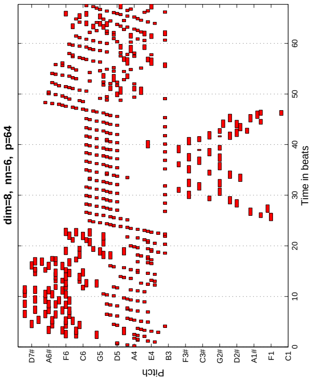

dim=8, nn=2, p=16
Time in beats
Pitch
F7
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
0
10
20
30
40
50
60

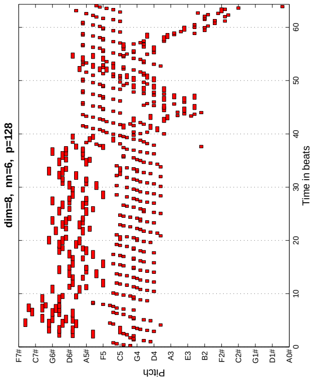

dim=8, nn=2, p=64
Time in beats
Pitch
A6
E6
B5
F5#
C5#
G4#
D4#
A3#
F3
C3
G2
D2
A1
E1
B0
F0#
C0#
G9#
D9#
B8-
0
10
20
30
40
50
60
70
80

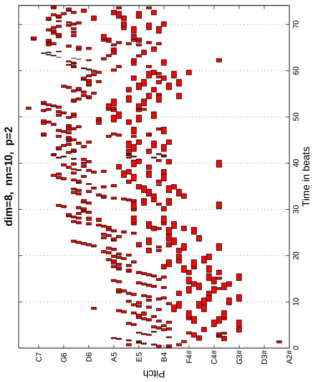

dim=8, nn=2, p=128
Time in beats
Pitch
B7
F7#
C7#
G6#
D6#
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
D1#
A0#
0
10
20
30
40
50
60

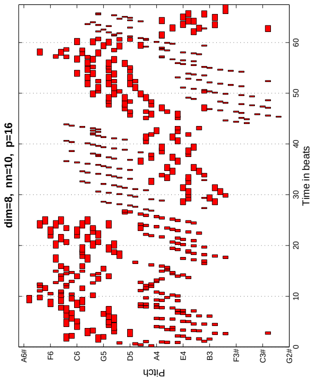

dim=8, nn=6, p=2
Time in beats
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
0
10
20
30
40
50
60
70

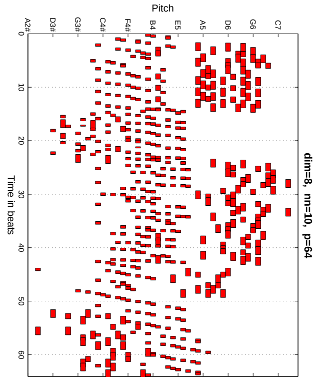

dim=8, nn=6, p=16
Time in beats
Pitch
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
0
10
20
30
40
50
60

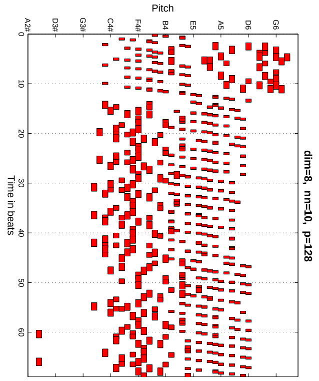

dim=8, nn=6, p=64
Time in beats
Pitch
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
F1
C1
0
10
20
30
40
50
60

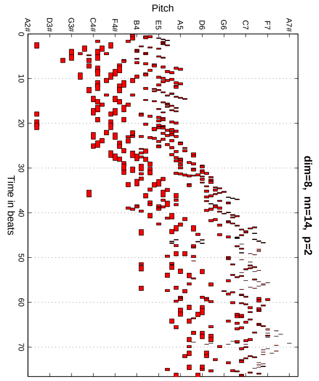

dim=8, nn=6, p=128
Time in beats
Pitch
F7#
C7#
G6#
D6#
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
D1#
A0#
0
10
20
30
40
50
60

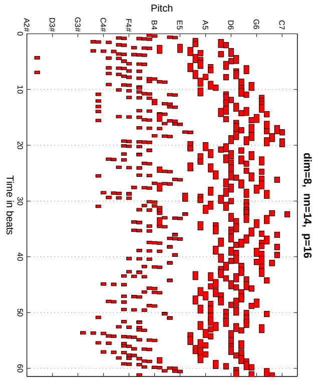

dim=8, nn=10, p=2
Time in beats
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
0
10
20
30
40
50
60
70

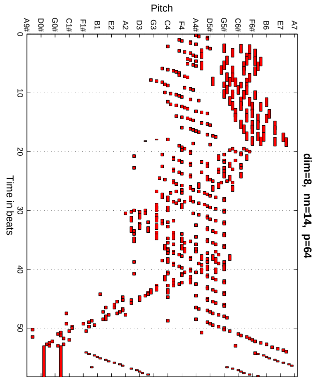

dim=8, nn=10, p=16
Time in beats
Pitch
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
0
10
20
30
40
50
60

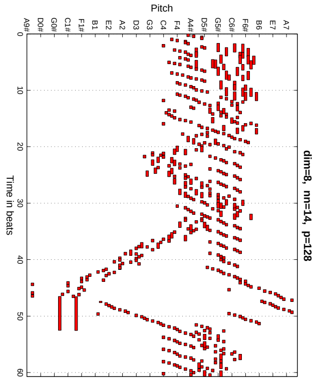

dim=8, nn=10, p=64
Pitch
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
C7
Time in beats
0
10
20
30
40
50
60

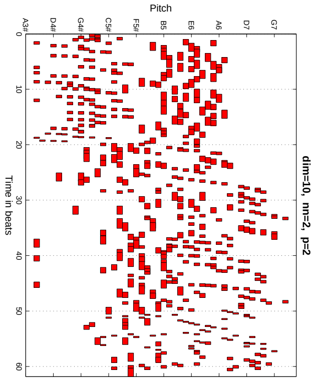

dim=8, nn=10, p=128
Pitch
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
Time in beats
0
10
20
30
40
50
60

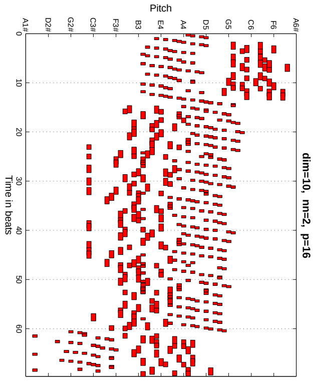

dim=8, nn=14, p=2
Pitch
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
C7
F7
A7#
Time in beats
0
10
20
30
40
50
60
70

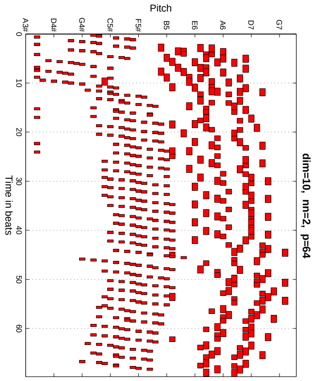

dim=8, nn=14, p=16
Pitch
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
C7
Time in beats
0
10
20
30
40
50
60

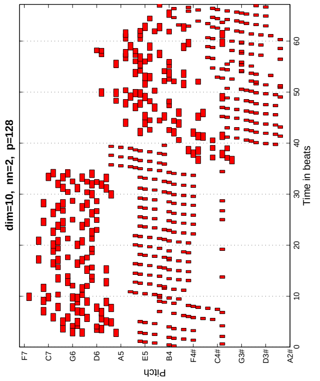

dim=8, nn=14, p=64
Pitch
A0#
D0#
G0#
C1#
F1#
B1
E2
A2
D3
G3
C4
F4
A4#
D5#
G5#
C6#
F6#
B6
E7
A7
Time in beats
0
10
20
30
40
50

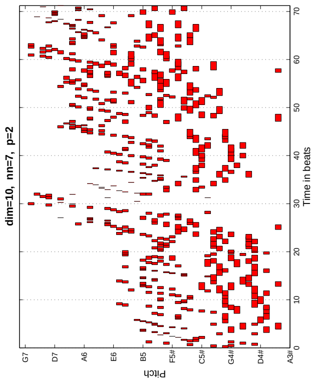

dim=8, nn=14, p=128
Pitch
A0#
D0#
G0#
C1#
F1#
B1
E2
A2
D3
G3
C4
F4
A4#
D5#
G5#
C6#
F6#
B6
E7
A7
Time in beats
0
10
20
30
40
50
60

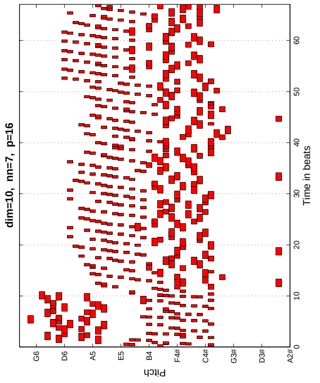

dim=10, nn=2, p=2
Pitch
A3#
D4#
G4#
C5#
F5#
B5
E6
A6
D7
G7
Time in beats
0
10
20
30
40
50
60

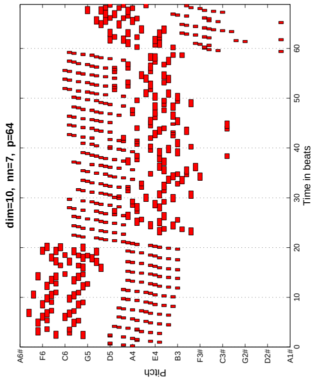

dim=10, nn=2, p=16
Pitch
A1#
D2#
G2#
C3#
F3#
B3
E4
A4
D5
G5
C6
F6
A6#
Time in beats
0
10
20
30
40
50
60

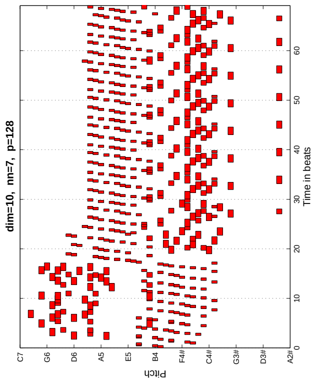

dim=10, nn=2, p=64
Pitch
A3#
D4#
G4#
C5#
F5#
B5
E6
A6
D7
G7
Time in beats
0
10
20
30
40
50
60

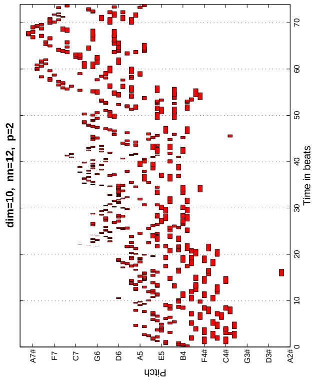

dim=10, nn=2, p=128
Pitch
F7
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
10
20
30
40
50
60

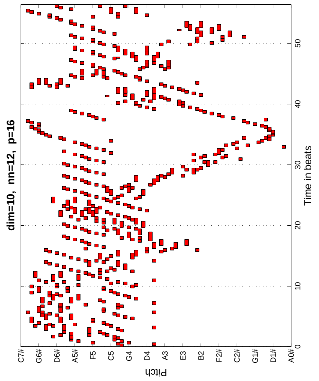

dim=10, nn=7, p=2
Pitch
G7
D7
A6
E6
B5
F5#
C5#
G4#
D4#
A3#
Time in beats
10
20
30
40
50
60
70

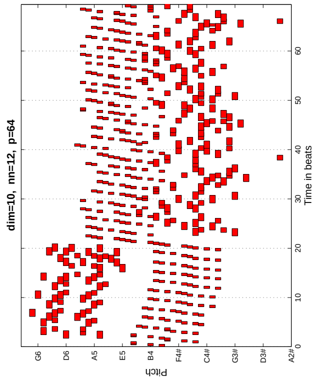

dim=10, nn=7, p=16
Pitch
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
10
20
30
40
50
60

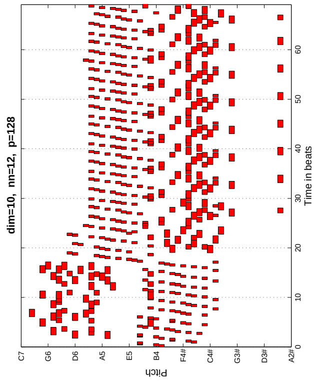

dim=10, nn=7, p=64
Pitch
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
10
20
30
40
50
60

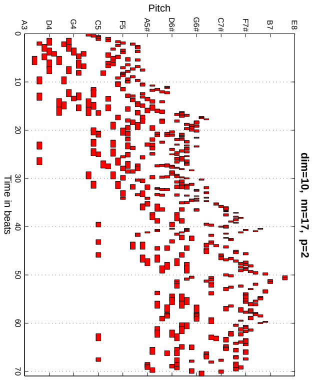

dim=10, nn=7, p=128
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
10
20
30
40
50
60

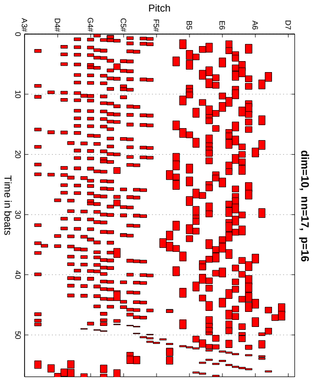

dim=10, nn=12, p=2
Pitch
A7#
F7
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
10
20
30
40
50
60
70

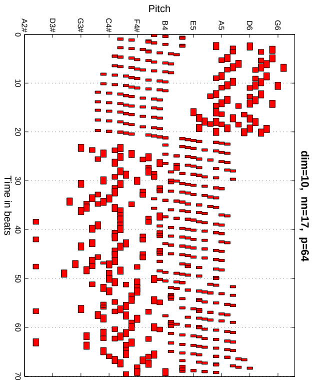

dim=10, nn=12, p=16
Pitch
C7#
G6#
D6#
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
G1#
D1#
A0#
Time in beats
0
10
20
30
40
50

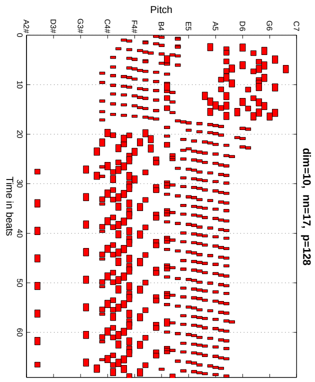

dim=10, nn=12, p=64
Pitch
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
10
20
30
40
50
60


dim=10, nn=12, p=128
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
10
20
30
40
50
60


dim=10, nn=17, p=2
Pitch
A3
D4
G4
C5
F5
A5#
D6#
G6#
C7#
F7#
B7
E8
Time in beats
0
10
20
30
40
50
60
70


dim=10, nn=17, p=16
Pitch
A3#
D4#
G4#
C5#
F5#
B5
E6
A6
D7
Time in beats
0
10
20
30
40
50


dim=10, nn=17, p=64
Pitch
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
Time in beats
0
10
20
30
40
50
60
70


dim=10, nn=17, p=128
Pitch
A2#
D3#
G3#
C4#
F4#
B4
E5
A5
D6
G6
C7
Time in beats
0
10
20
30
40
50
60

B.6 Combination of Etudes 4 and 6, Method 3 with Components Modeled Jointly

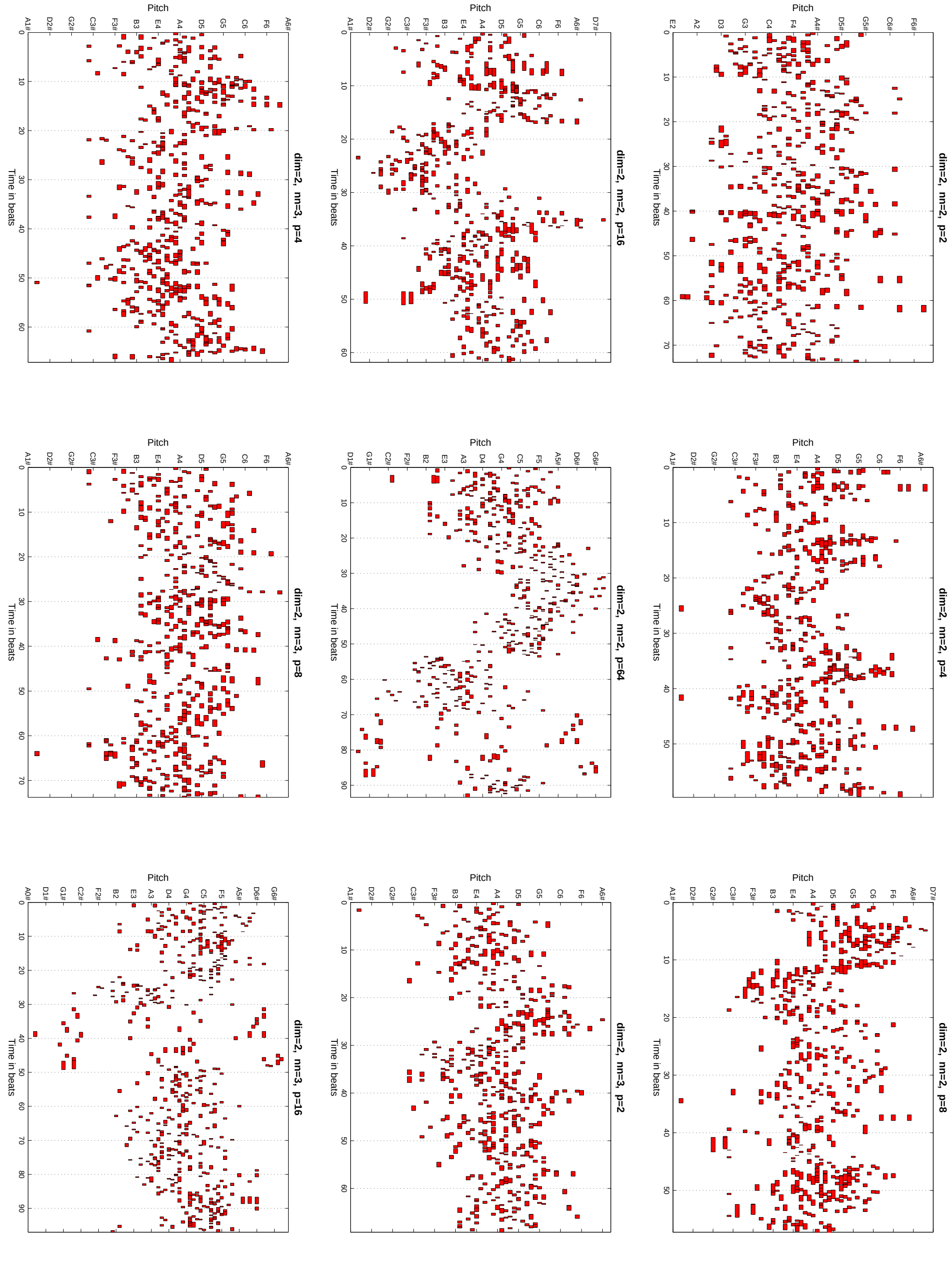
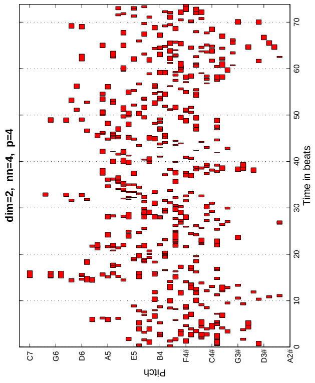
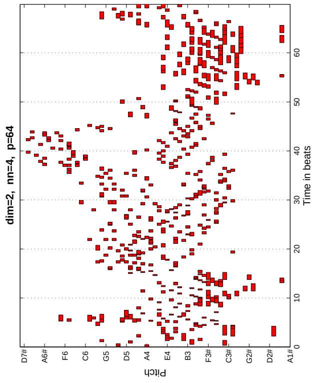
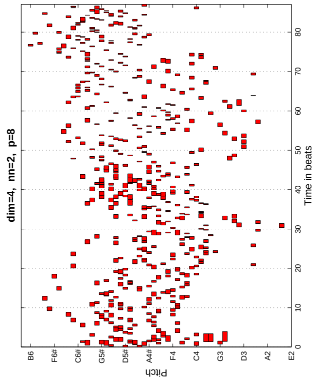
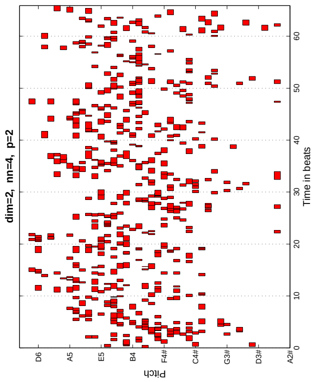


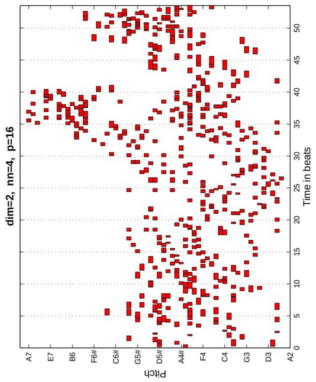
Figure B-6: 500 note fragments from combinations of Ligeti's Etudes 4 and 6 Book 1 using *method 3* with parameters modeled jointly in a single state-space. Each panel is a combination of the Etudes using different parameter values.

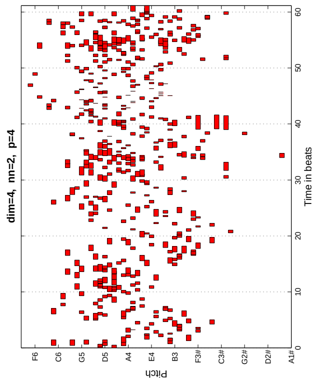

dim=2, nn=4, p=4
Pitch
C7
G6
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
10
20
30
40
50
60
70

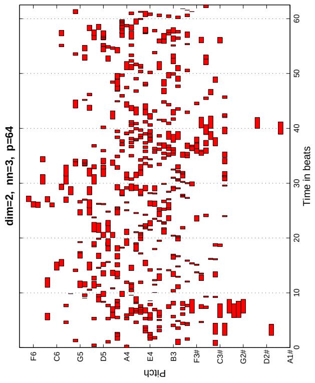

dim=2, nn=4, p=64
Pitch
D7#
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
10
20
30
40
50
60

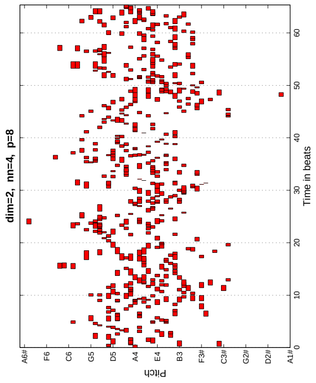

dim=4, nn=2, p=8
Pitch
B6
F6#
C6#
G5#
D5#
A4#
F4
C4
G3
D3
A2
E2
Time in beats
0
10
20
30
40
50
60
70
80

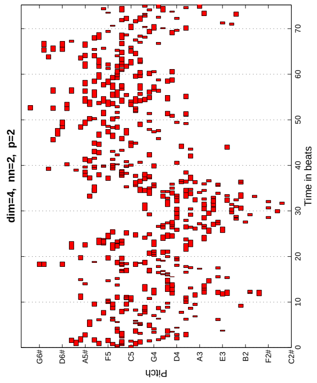

dim=2, nn=4, p=2
Pitch
D6
A5
E5
B4
F4#
C4#
G3#
D3#
A2#
Time in beats
0
10
20
30
40
50
60

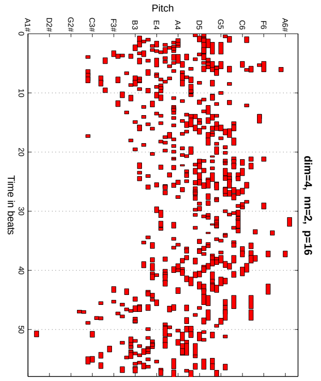

dim=2, nn=4, p=16
Pitch
A7
E7
B6
F6#
C6#
G5#
D5#
A4#
F4
C4
G3
D3
A2
Time in beats
0
5
10
15
20
25
30
35
40
45
50

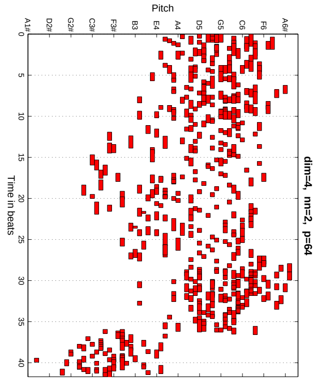

dim=4, nn=2, p=4
Pitch
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
10
20
30
40
50
60

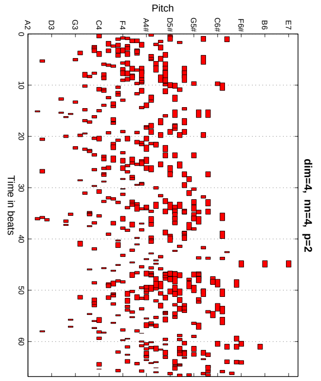

dim=2, nn=3, p=64
Pitch
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
10
20
30
40
50
60

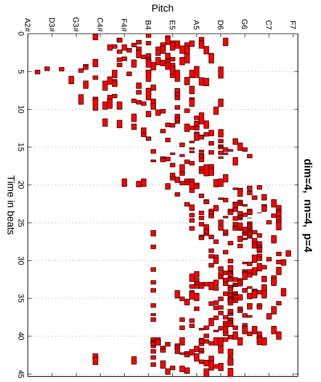

dim=2, nn=4, p=8
Pitch
A6#
F6
C6
G5
D5
A4
E4
B3
F3#
C3#
G2#
D2#
A1#
Time in beats
0
10
20
30
40
50
60

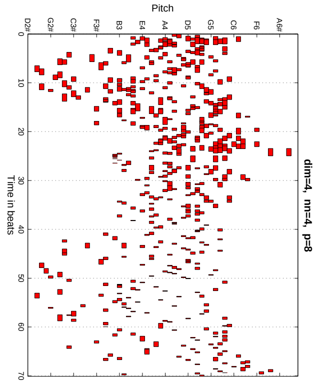

dim=4, nn=2, p=2
Pitch
G6#
D6#
A5#
F5
C5
G4
D4
A3
E3
B2
F2#
C2#
Time in beats
0
10
20
30
40
50
60
70

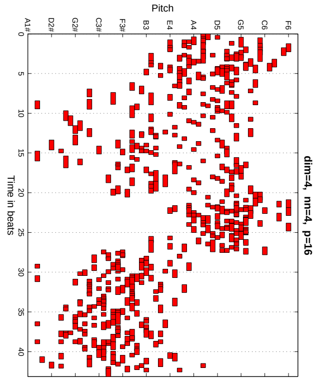

dim=4, nn=2, p=16
Pitch
A1# D2# G2# C3# F3# B3 E4 A4 D5 G5 C6 F6 A6#
Time in beats
0 10 20 30 40 50

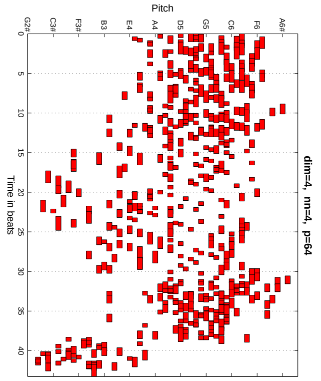

dim=4, nn=2, p=64
Pitch
A1# D2# G2# C3# F3# B3 E4 A4 D5 G5 C6 F6 A6#
Time in beats
0 5 10 15 20 25 30 35 40

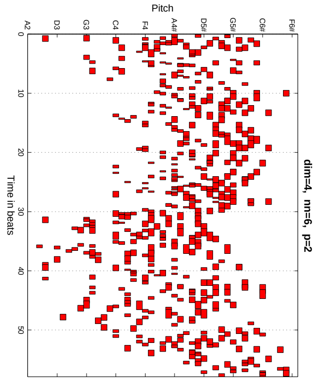

dim=4, nn=4, p=2
Pitch
A2 D3 G3 C4 F4 A4# D5# G5# C6# F6# B6 E7
Time in beats
0 10 20 30 40 50 60

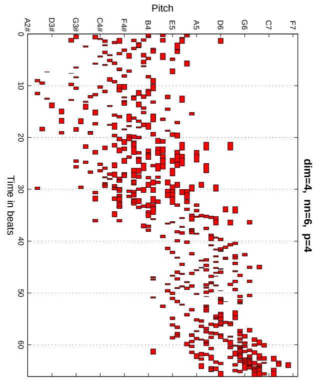

dim=4, nn=4, p=4
Pitch
A2# D3# G3# C4# F4# B4 E5 A5 D6 G6 C7 F7
Time in beats
0 5 10 15 20 25 30 35 40 45

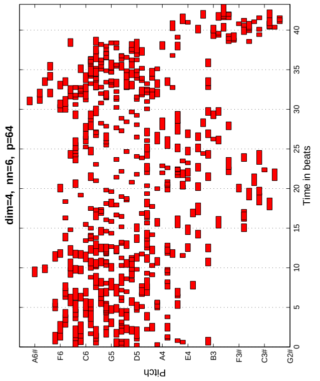

dim=4, nn=4, p=8
Pitch
D2# G2# C3# F3# B3 E4 A4 D5 G5 C6 F6 A6#
Time in beats
0 10 20 30 40 50 60 70

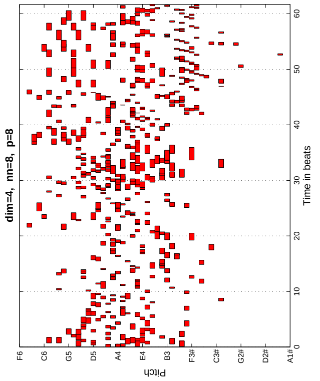

dim=4, nn=4, p=16
Pitch
A1# D2# G2# C3# F3# B3 E4 A4 D5 G5 C6 F6
Time in beats
0 5 10 15 20 25 30 35 40

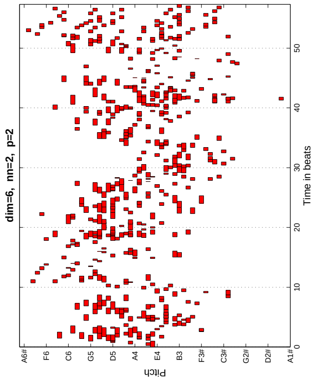

dim=4, nn=4, p=64
Pitch
G2# C3# F3# B3 E4 A4 D5 G5 C6 F6 A6#
Time in beats
0 5 10 15 20 25 30 35 40

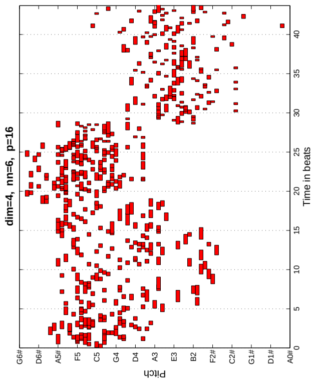

dim=4, nn=6, p=2
Pitch
A2 D3 G3 C4 F4 A4# D5# G5# C6# F6#
Time in beats
0 10 20 30 40 50

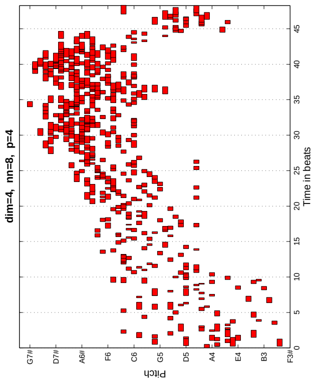

dim=4, nn=6, p=4
Pitch
A2# D3# G3# C4# F4# B4 E5 A5 D6 G6 C7 F7
Time in beats
0 10 20 30 40 50 60

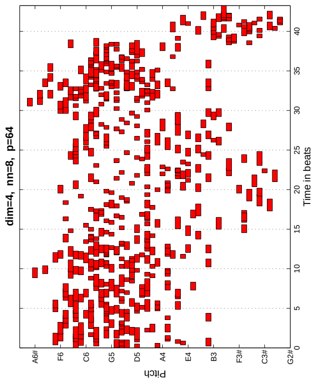

dim=4, nn=6, p=64
Pitch
Time in beats

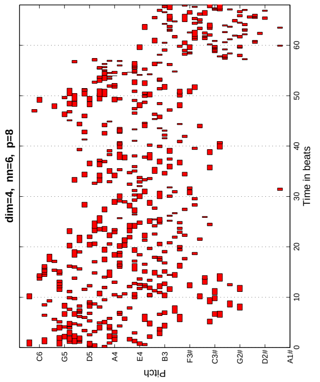

dim=4, nn=8, p=8
Pitch
Time in beats

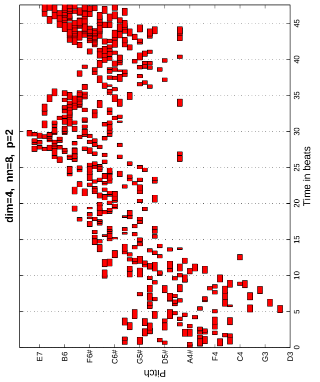

dim=6, nn=2, p=2
Pitch
Time in beats

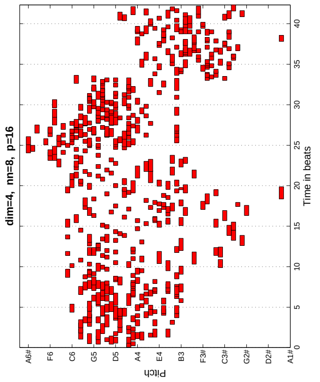

dim=4, nn=6, p=16
Pitch
Time in beats

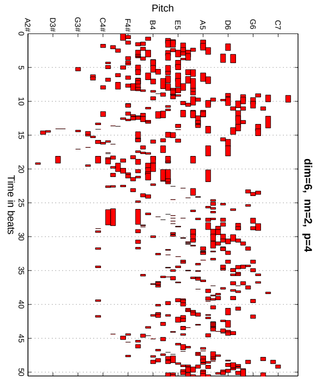

dim=4, nn=8, p=4
Pitch
Time in beats

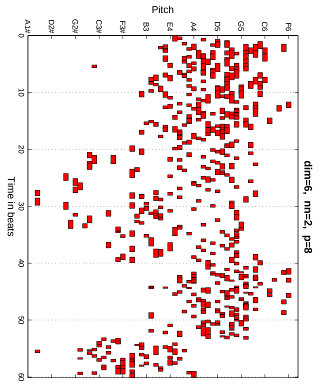

dim=4, nn=8, p=64
Pitch
Time in beats

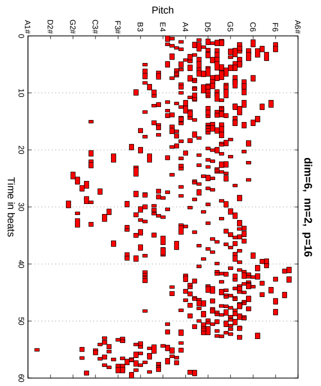

dim=4, nn=6, p=8
Pitch
Time in beats

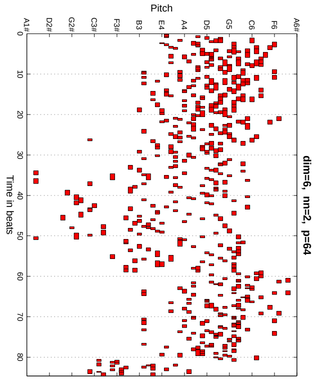

dim=4, nn=8, p=2
Pitch
Time in beats

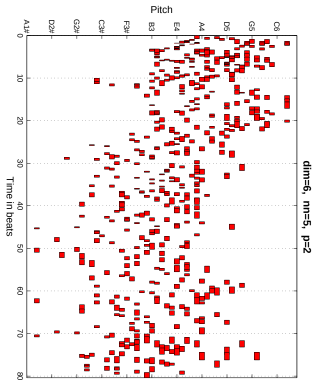

dim=4, nn=8, p=16
Pitch
Time in beats

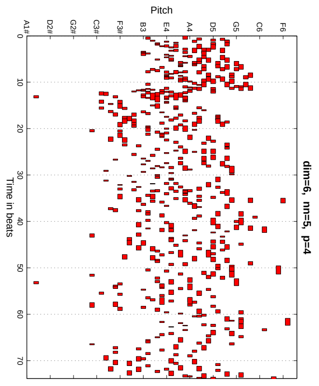

dim=6, nn=2, p=4

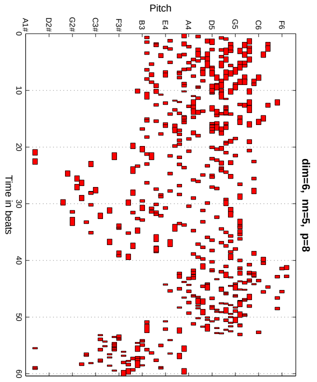

dim=6, nn=2, p=8

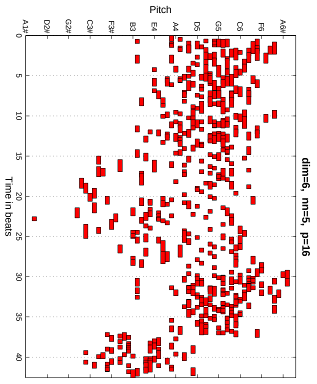

dim=6, nn=2, p=16

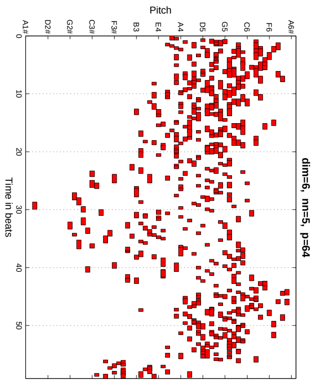

dim=6, nn=2, p=64

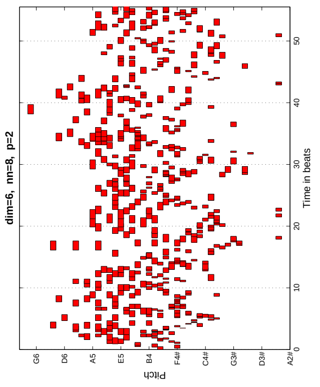

dim=6, nn=5, p=2

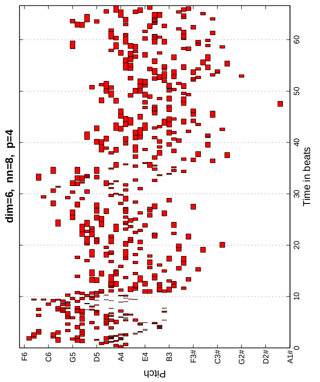

dim=6, nn=5, p=4

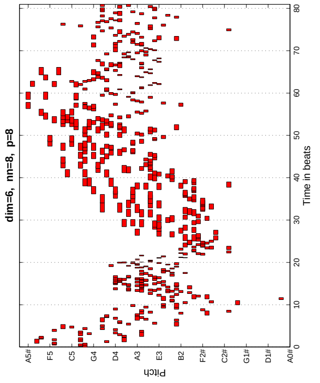

dim=6, nn=5, p=8

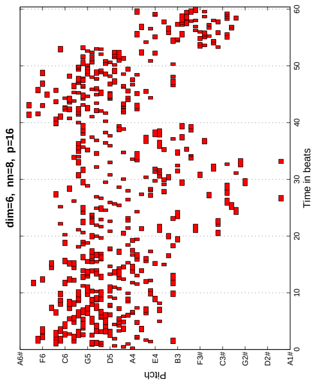

dim=6, nn=5, p=16

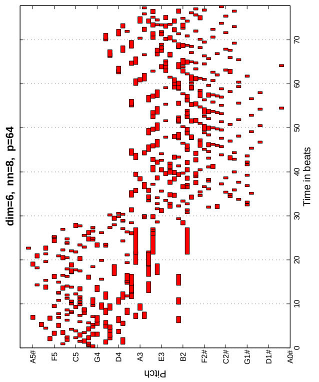

dim=6, nn=5, p=64

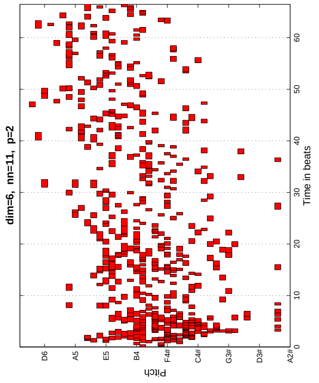

dim=6, nn=8, p=2

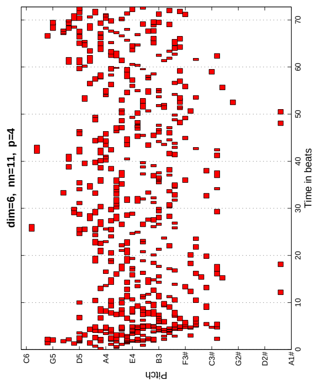

dim=6, nn=8, p=4

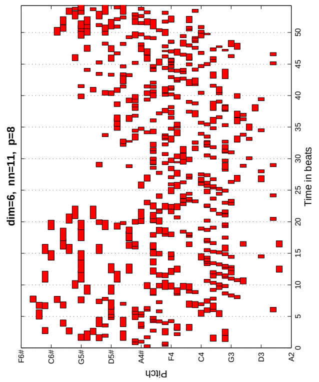

dim=6, nn=8, p=8

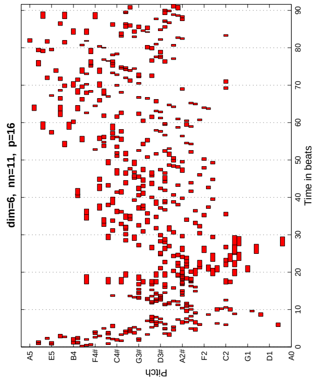

dim=6, nn=8, p=16

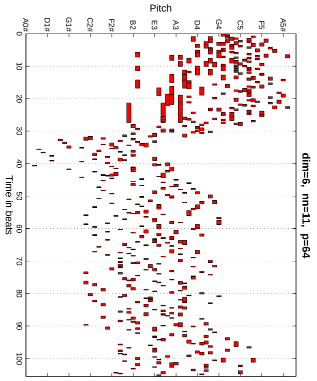

dim=6, nn=8, p=64

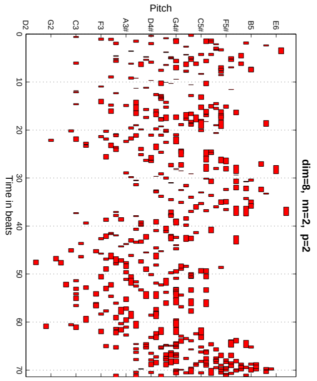

dim=6, nn=11, p=2

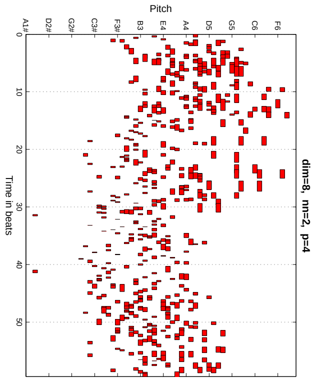

dim=6, nn=11, p=4

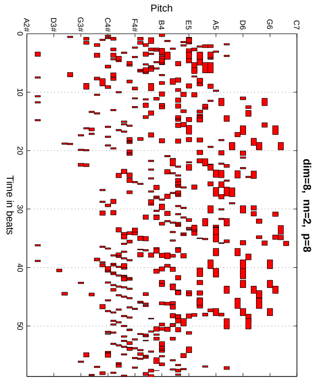

dim=6, nn=11, p=8

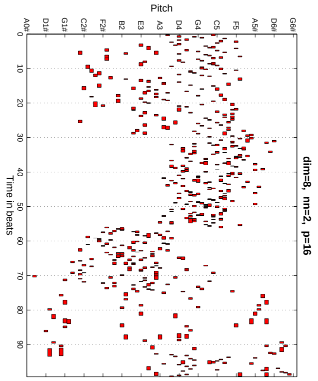

dim=6, nn=11, p=16

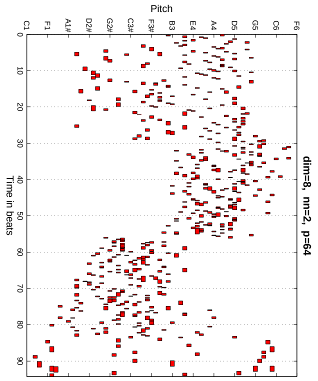

dim=6, nn=11, p=64

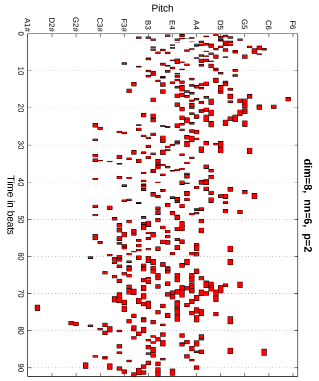

dim=8, nn=2, p=2

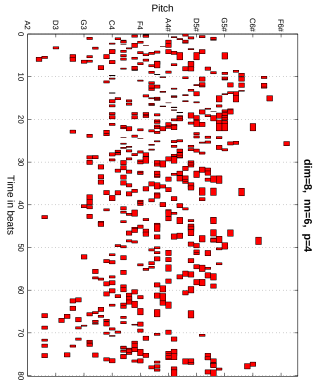

dim=8, nn=2, p=4

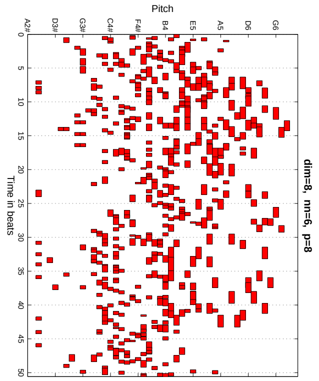

dim=8, nn=2, p=8

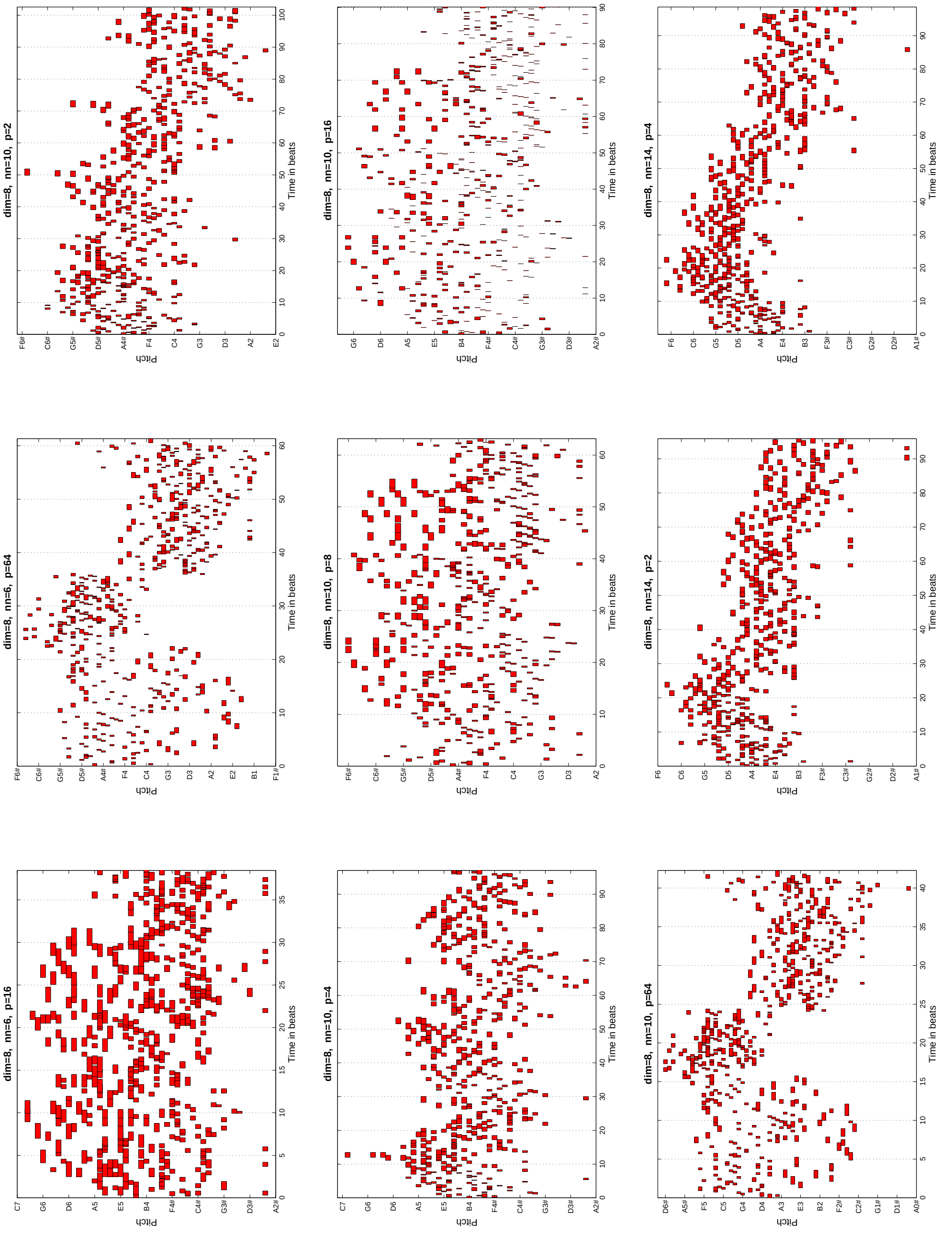

dim=8, nn=2, p=16

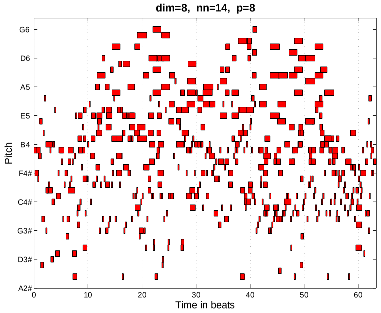

dim=8, nn=2, p=64

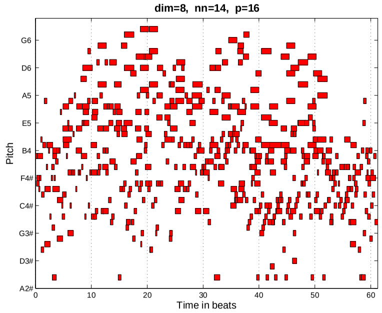

dim=8, nn=6, p=2

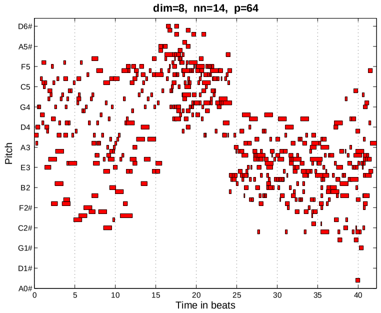

dim=8, nn=6, p=4

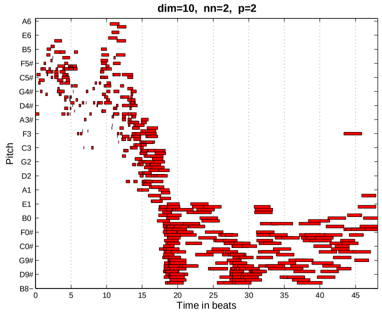

dim=8, nn=6, p=8


dim=8, nn=6, p=16
dim=8, nn=6, p=64
dim=8, nn=10, p=2
dim=8, nn=10, p=4
dim=8, nn=10, p=8
dim=8, nn=10, p=16
dim=8, nn=10, p=64
dim=8, nn=14, p=2
dim=8, nn=14, p=4
Pitch
Time in beats


dim=8, nn=14, p=8


dim=8, nn=14, p=16


dim=8, nn=14, p=64


dim=10, nn=2, p=2

B.7 Combination of Etudes 4 and 6, Method 3 with Components Modeled Independently

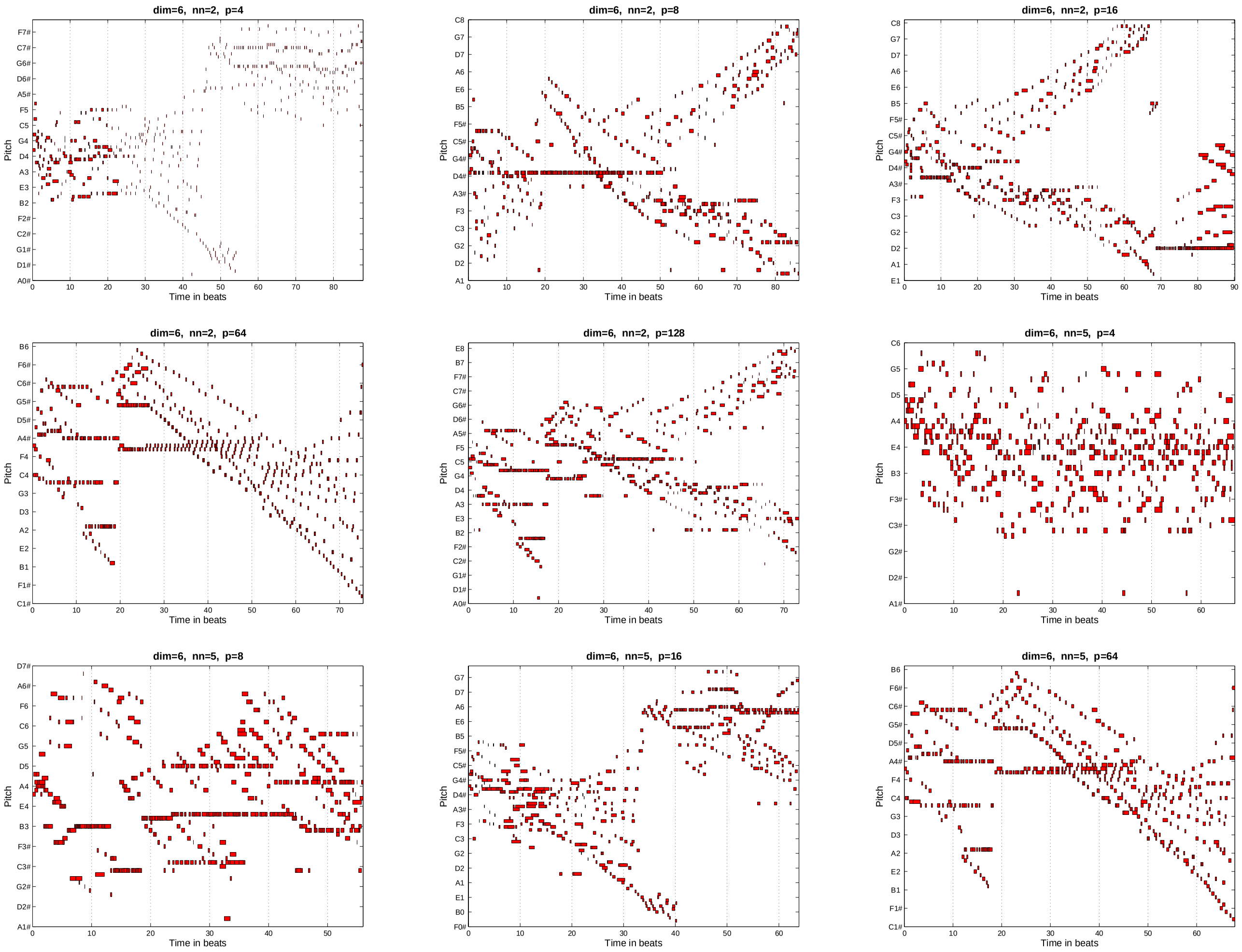
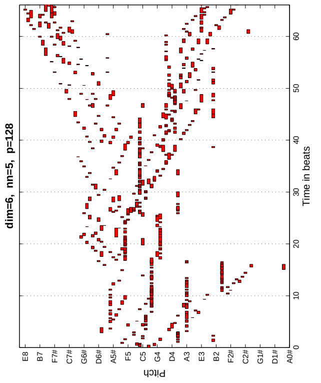
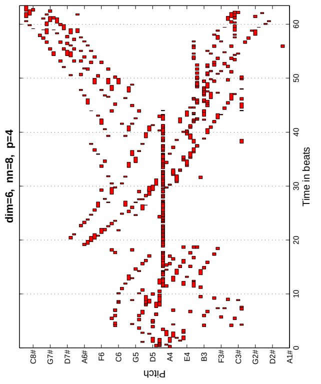


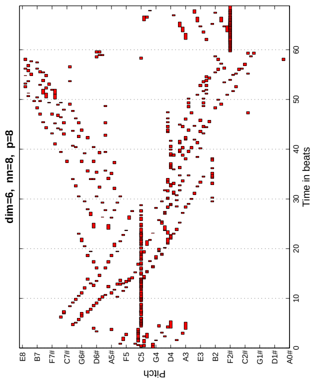
Figure B-7: **500 note fragments from combinations of Ligeti's Etudes 4 and 6 Book 1 using *method 3* with components modeled independently. Each panel is a combination of the Etudes using different parameter values.**



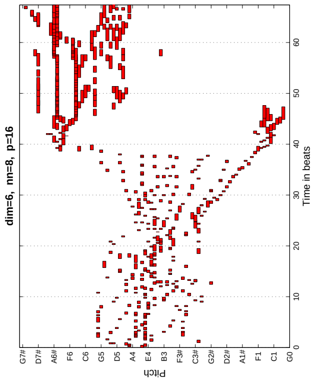
dim=6, nn=5, p=128



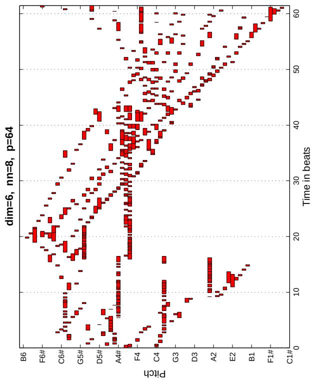
dim=6, nn=8, p=4



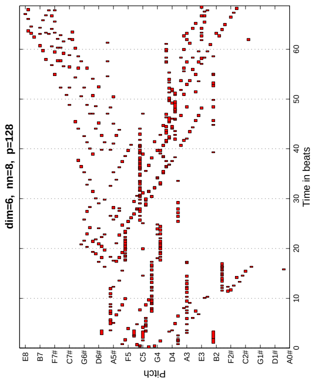
dim=6, nn=8, p=8



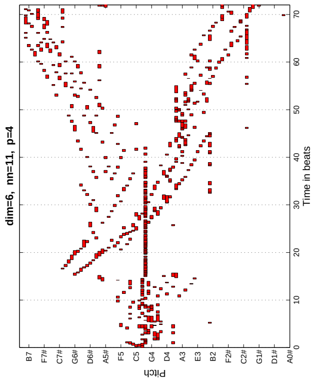
dim=6, nn=8, p=16



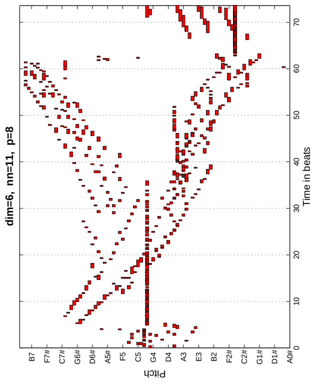
dim=6, nn=8, p=64



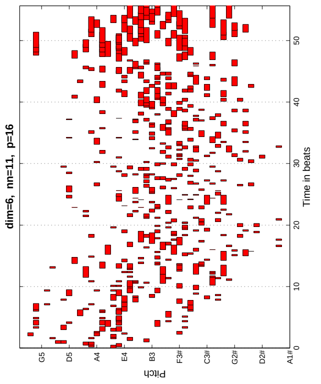
dim=6, nn=8, p=128



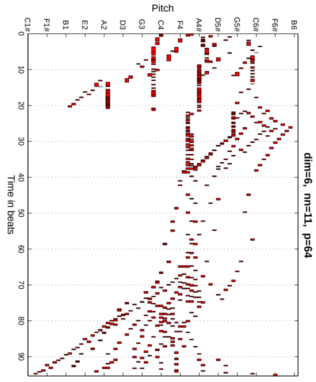
dim=6, nn=11, p=4

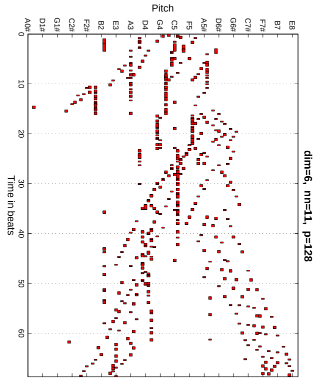


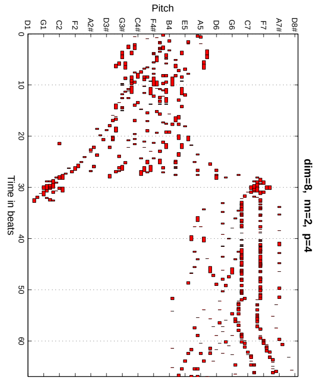
dim=6, nn=11, p=8

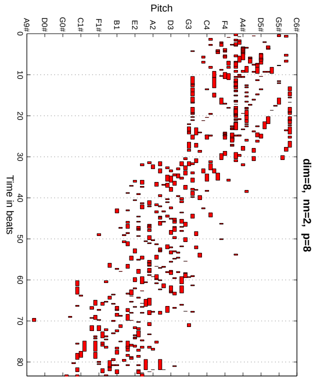


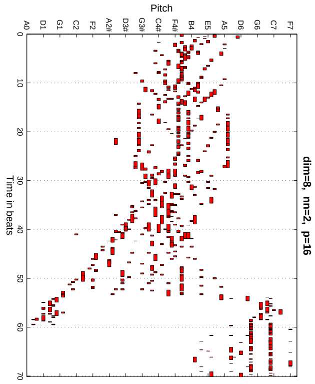
dim=6, nn=11, p=16

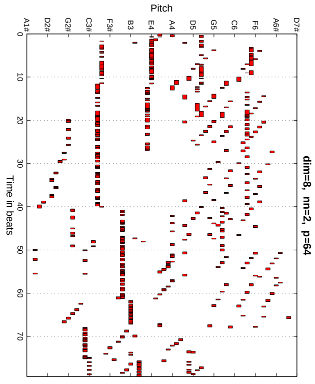

dim=6, nn=11, p=64
Pitch
Time in beats

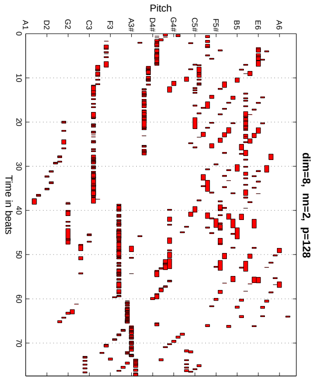

dim=6, nn=11, p=128
Pitch
Time in beats

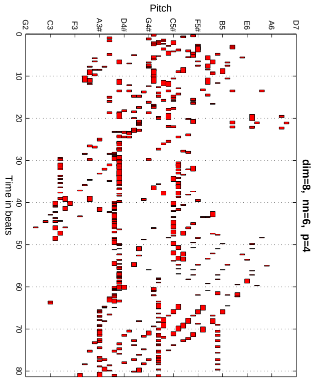

dim=8, nn=2, p=4
Pitch
Time in beats

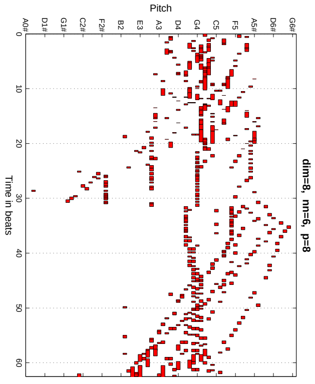

dim=8, nn=2, p=8
Pitch
Time in beats

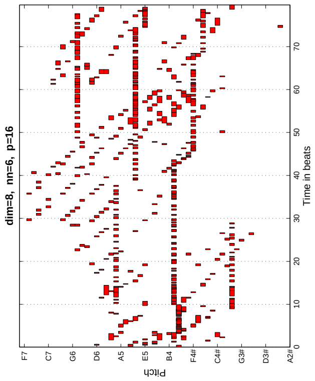

dim=8, nn=2, p=16
Pitch
Time in beats

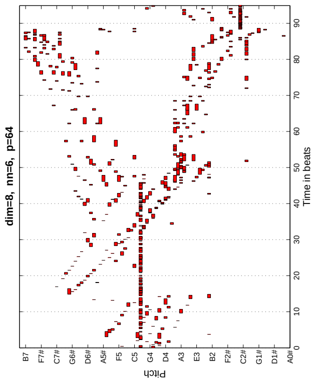

dim=8, nn=2, p=64
Pitch
Time in beats

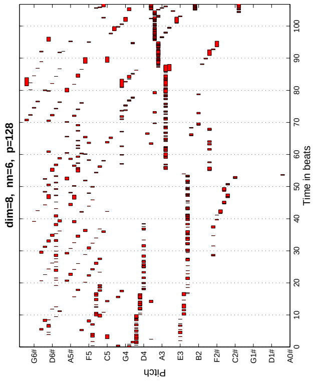

dim=8, nn=2, p=128
Pitch
Time in beats

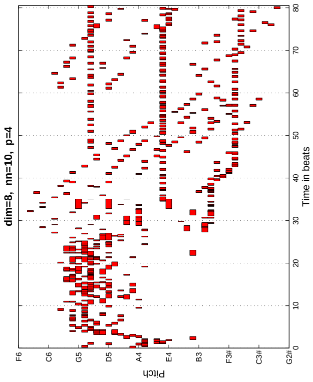

dim=8, nn=6, p=4
Pitch
Time in beats

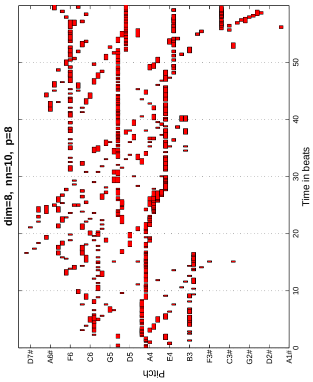

dim=8, nn=6, p=8
Pitch
Time in beats

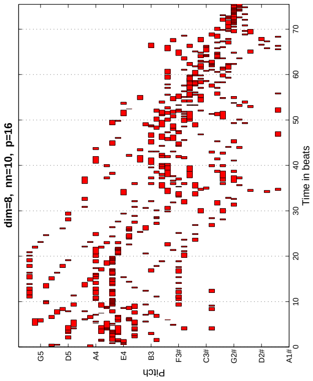

dim=8, nn=6, p=16
Pitch
Time in beats

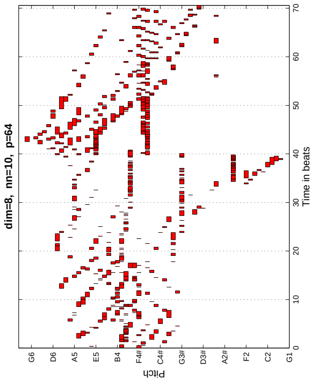

dim=8, nn=6, p=64
Pitch
Time in beats

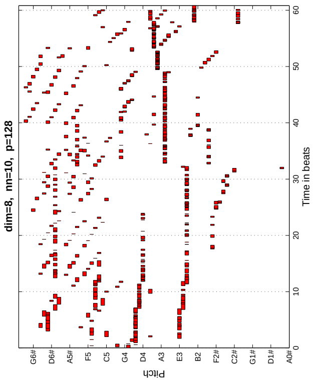

dim=8, nn=6, p=128
Pitch
Time in beats

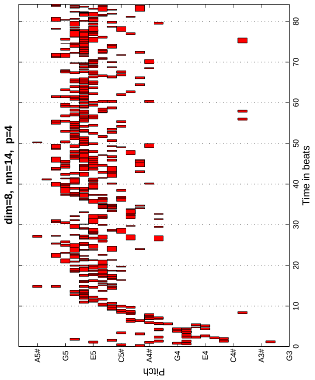

dim=8, nn=10, p=4
Pitch
Time in beats

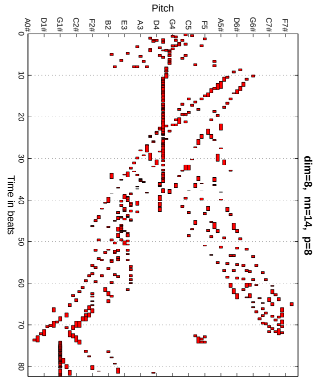

dim=8, nn=10, p=8
Pitch
Time in beats

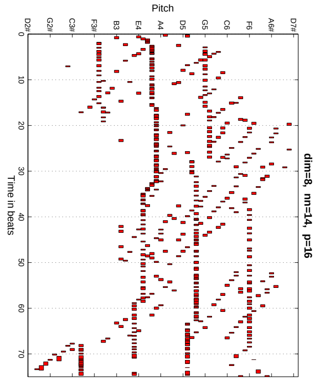

dim=8, nn=10, p=16
Pitch
Time in beats

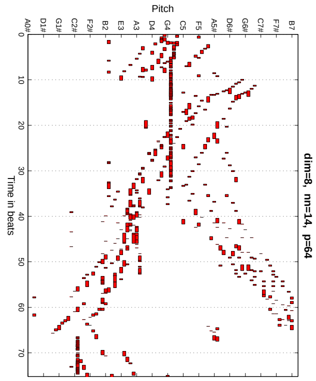

dim=8, nn=10, p=64
Pitch
Time in beats

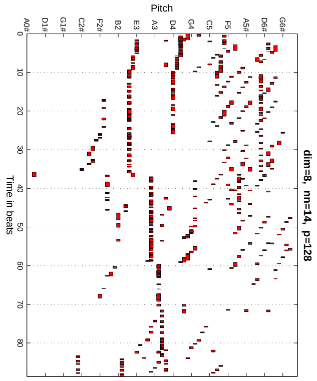

dim=8, nn=10, p=128
Pitch
Time in beats

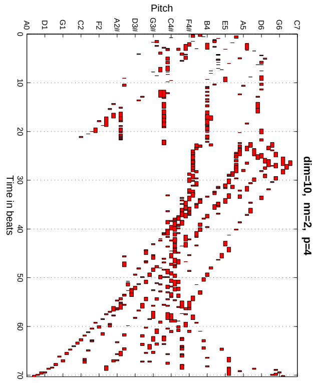

dim=8, nn=14, p=4
Pitch
Time in beats

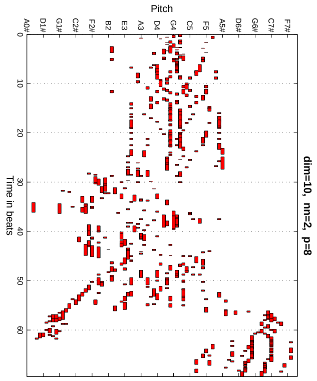

dim=8, nn=14, p=8

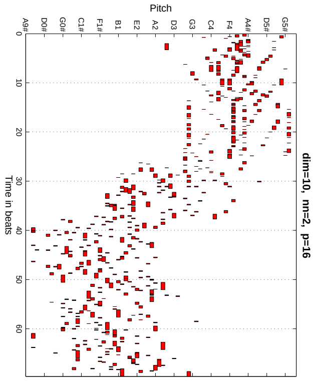

dim=8, nn=14, p=16

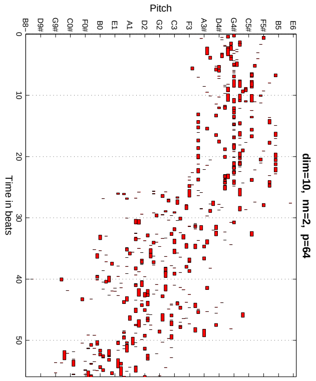

dim=8, nn=14, p=64

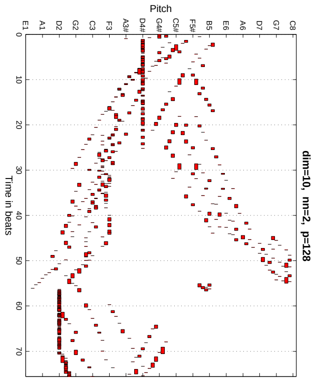

dim=8, nn=14, p=128

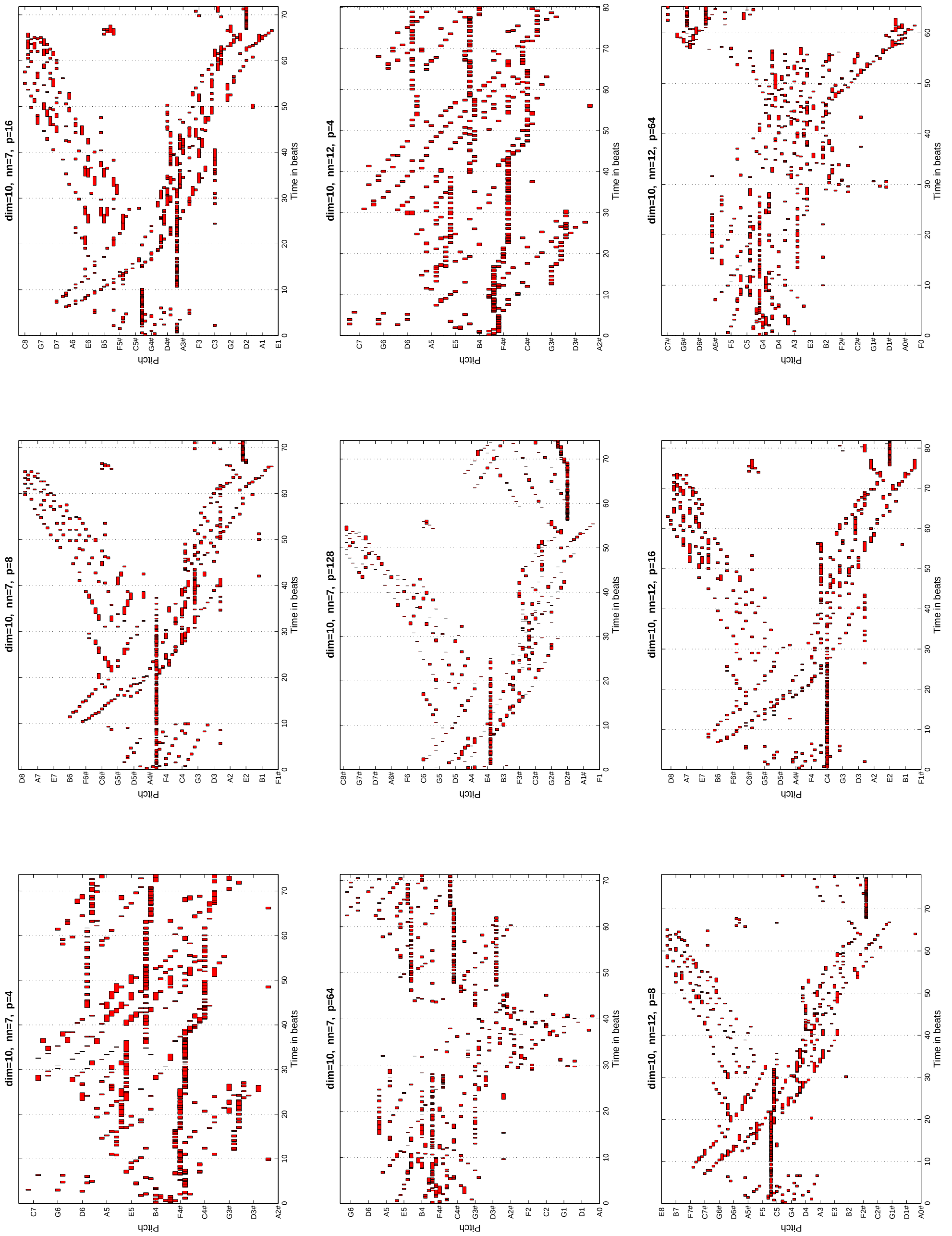

dim=10, nn=2, p=4

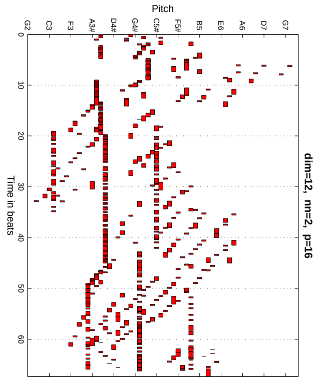

dim=10, nn=2, p=8

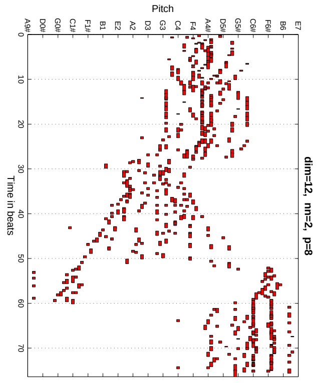

dim=10, nn=2, p=16

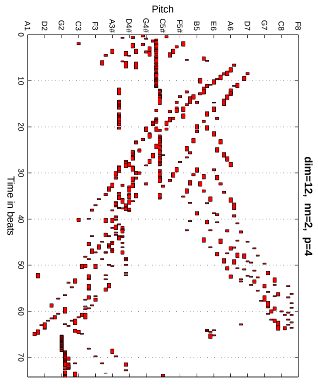

dim=10, nn=2, p=64

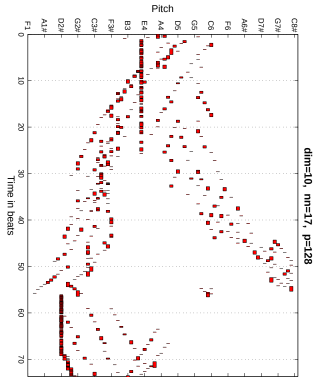

dim=10, nn=2, p=128

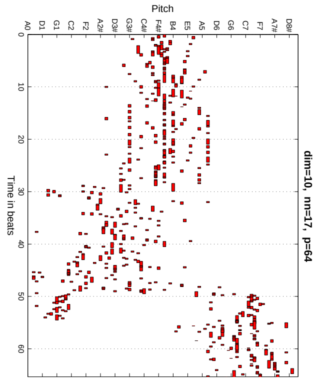

dim=10, nn=7, p=16
dim=10, nn=12, p=4
dim=10, nn=12, p=64
dim=10, nn=7, p=8
dim=10, nn=7, p=128
dim=10, nn=12, p=16
dim=10, nn=7, p=4
dim=10, nn=7, p=64
dim=10, nn=12, p=8
Pitch
Time in beats

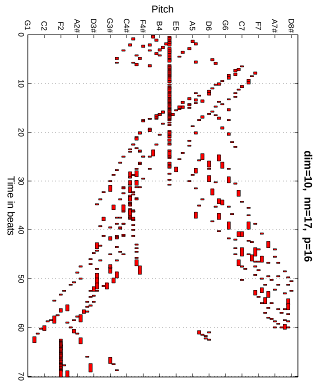

dim=12, nn=2, p=16
Pitch
Time in beats

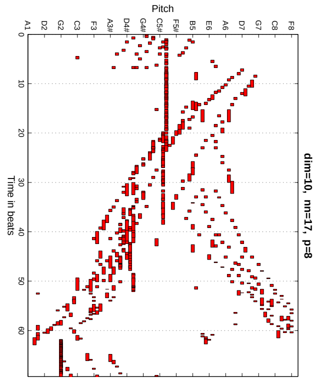

dim=12, nn=2, p=8
Pitch
Time in beats

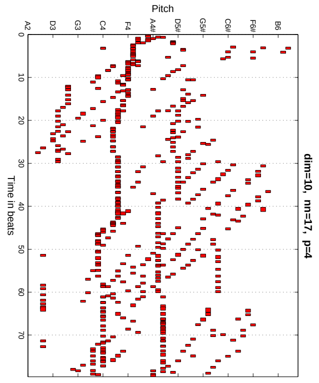

dim=12, nn=2, p=4
Pitch
Time in beats

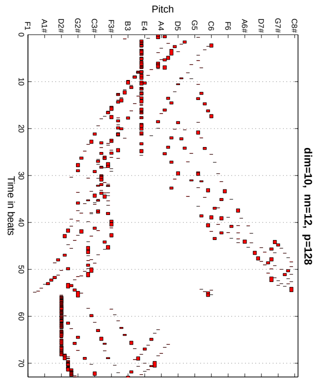

dim=10, nn=17, p=128
Pitch
Time in beats

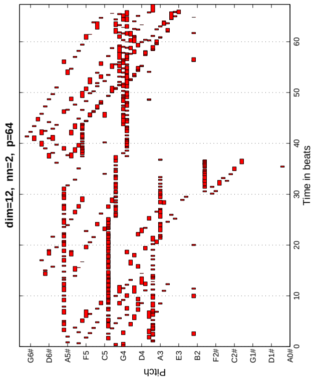

dim=10, nn=17, p=64
Pitch
Time in beats

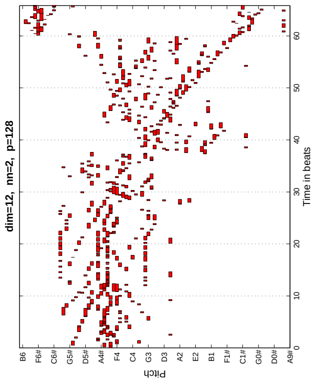

dim=10, nn=17, p=16
Pitch
Time in beats

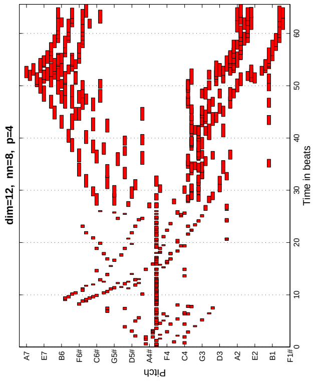

dim=10, nn=17, p=8
Pitch
Time in beats

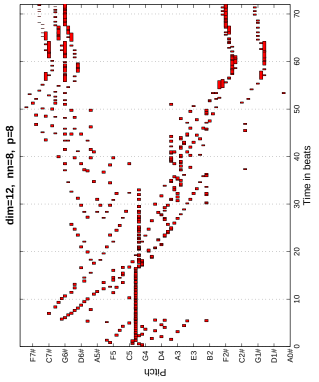

dim=10, nn=17, p=4
Pitch
Time in beats

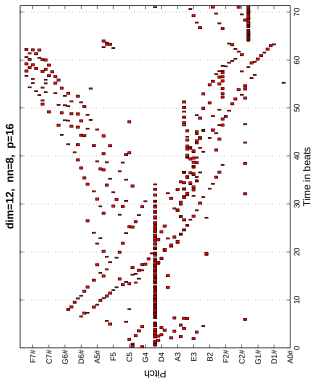

dim=10, nn=12, p=128
Pitch
Time in beats

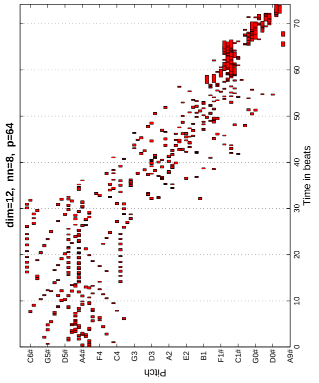

dim=12, nn=2, p=64
Pitch
Time in beats

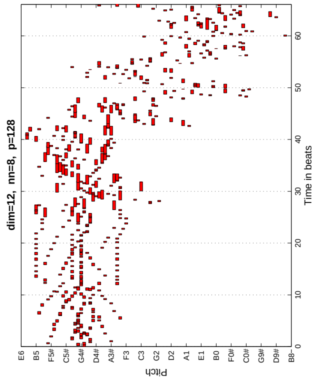

dim=12, nn=2, p=128
Pitch
Time in beats

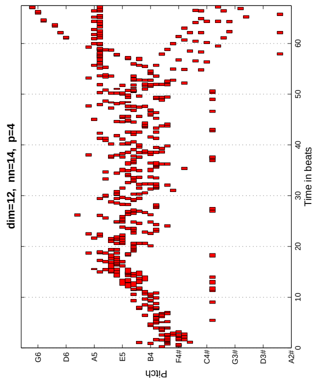

dim=12, nn=8, p=4
Pitch
Time in beats

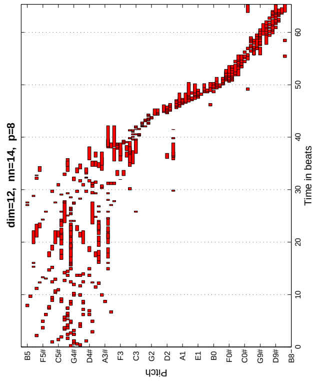

dim=12, nn=8, p=8
Pitch
Time in beats

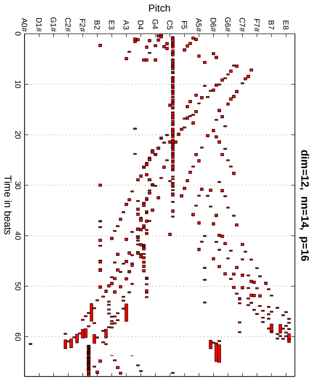

dim=12, nn=8, p=16
Pitch
Time in beats

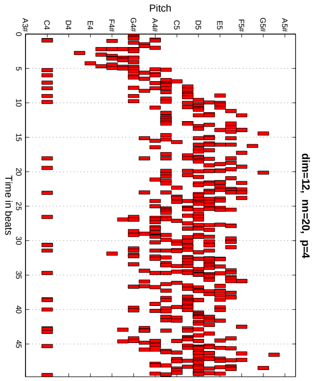

dim=12, nn=8, p=64
Pitch
Time in beats

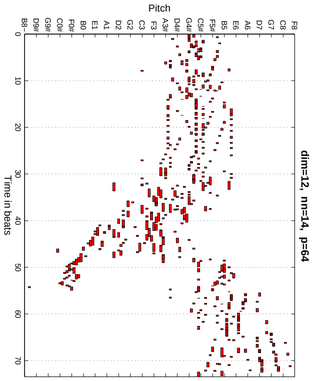

dim=12, nn=8, p=128
Pitch
Time in beats

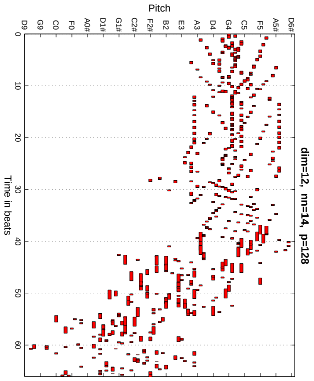

dim=12, nn=14, p=4
Pitch
Time in beats


dim=12, nn=14, p=8
Pitch
Time in beats

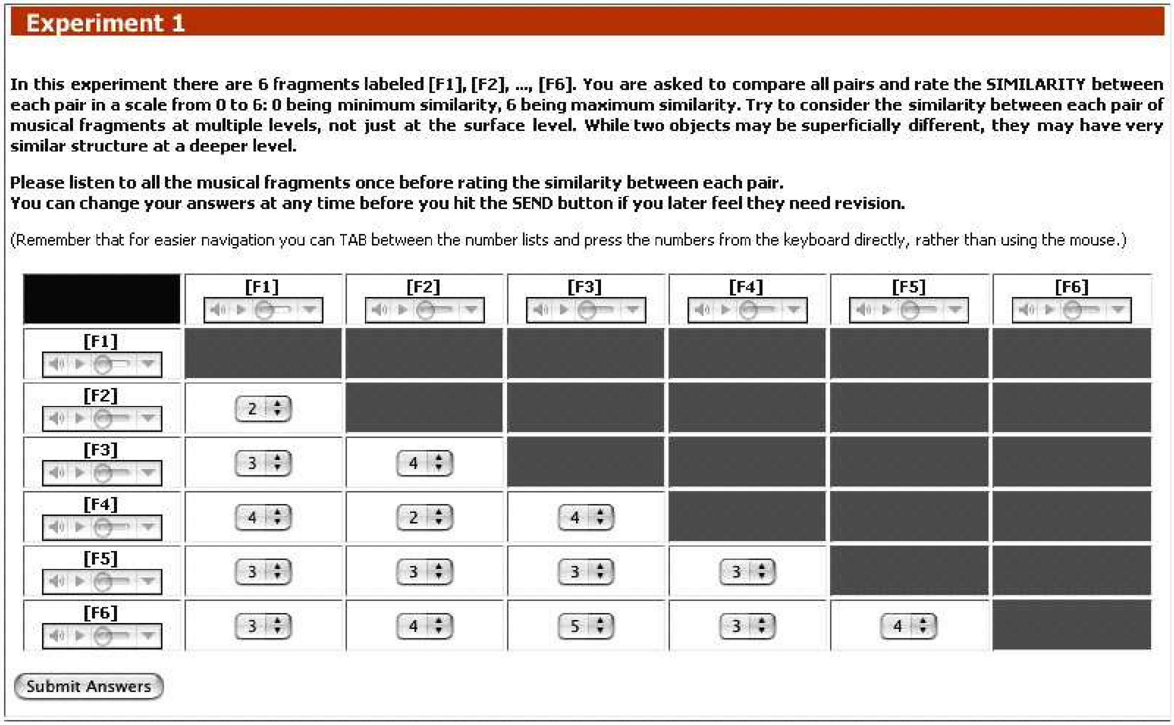

dim=12, nn=14, p=16
Pitch
Time in beats
A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5# D6# G6# C7# F7# B7 E8
0 10 20 30 40 50 60


dim=12, nn=20, p=4
Pitch
Time in beats
A3# C4 D4 E4 F4# G4# A4# C5 D5 E5 F5# G5# A5#
0 5 10 15 20 25 30 35 40 45


dim=12, nn=14, p=64
Pitch
Time in beats
B8- D9# G9# C0# F0# B0 E1 A1 D2 G2 C3 F3 A3# D4# G4# C5# F5# B5 E6 A6 D7 G7 C8 F8
0 10 20 30 40 50 60 70


dim=12, nn=14, p=128
Pitch
Time in beats
D9 G9 C0 F0 A0# D1# G1# C2# F2# B2 E3 A3 D4 G4 C5 F5 A5# D6#
0 10 20 30 40 50 60

Appendix THREE
Experiment Forms



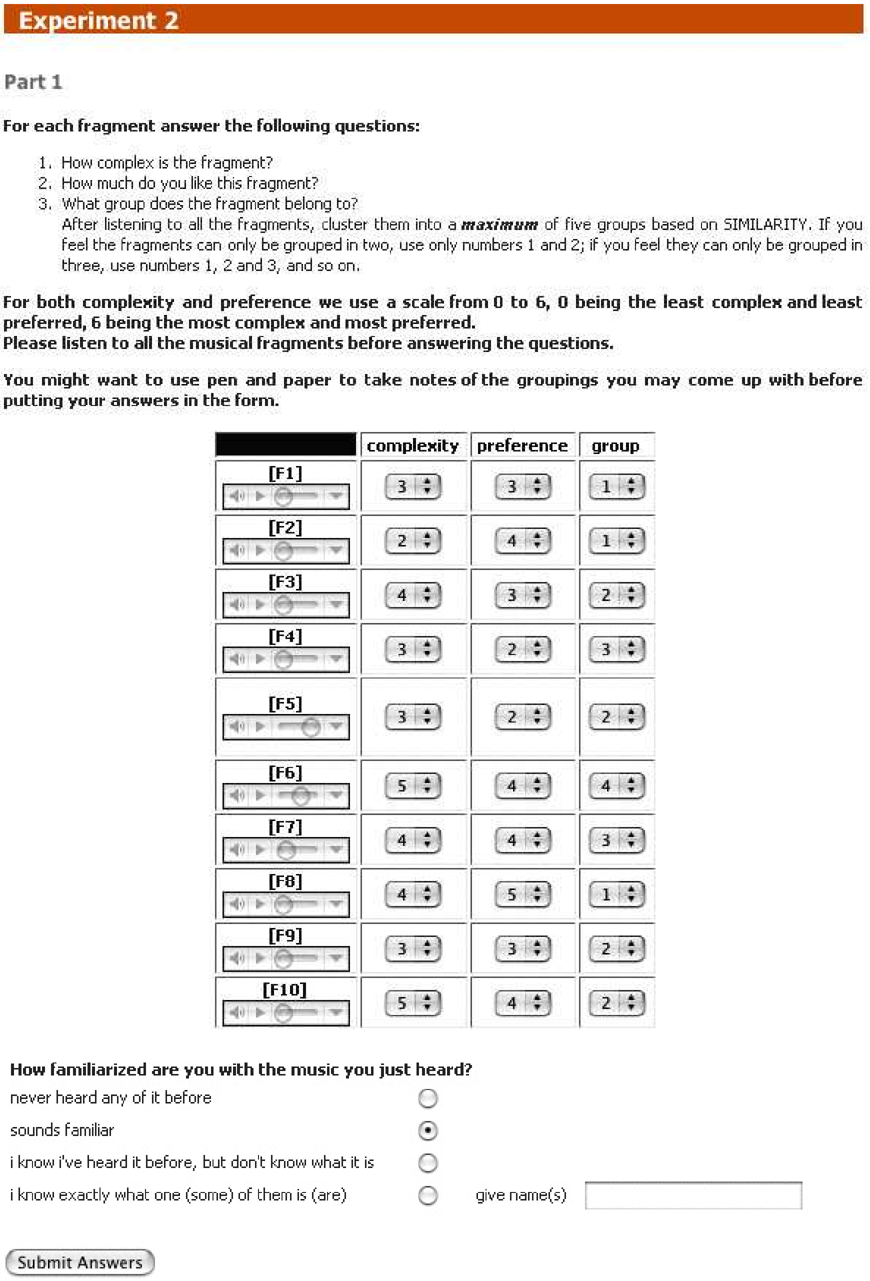
Experiment 1

In this experiment there are 6 fragments labeled [F1], [F2], ..., [F6]. You are asked to compare all pairs and rate the SIMILARITY between each pair in a scale from 0 to 6: 0 being minimum similarity, 6 being maximum similarity. Try to consider the similarity between each pair of musical fragments at multiple levels, not just at the surface level. While two objects may be superficially different, they may have very similar structure at a deeper level.

Please listen to all the musical fragments once before rating the similarity between each pair.
You can change your answers at any time before you hit the SEND button if you later feel they need revision.

(Remember that for easier navigation you can TAB between the number lists and press the numbers from the keyboard directly, rather than using the mouse.)

	[F1]	[F2]	[F3]	[F4]	[F5]	[F6]
[F1]						
[F2]	2					
[F3]	3	4				
[F4]	4	2	4			
[F5]	3	3	3	3		
[F6]	3	4	5	3	4	



Experiment 2

Part 1

For each fragment answer the following questions:

1. How complex is the fragment?
2. How much do you like this fragment?
3. What group does the fragment belong to?
   After listening to all the fragments, cluster them into a ***maximum*** of five groups based on SIMILARITY. If you feel the fragments can only be grouped in two, use only numbers 1 and 2; if you feel they can only be grouped in three, use numbers 1, 2 and 3, and so on.

For both complexity and preference we use a scale from 0 to 6, 0 being the least complex and least preferred, 6 being the most complex and most preferred.
Please listen to all the musical fragments before answering the questions.

You might want to use pen and paper to take notes of the groupings you may come up with before putting your answers in the form.

	complexity	preference	group
[F1]	3	3	1
[F2]	2	4	1
[F3]	4	3	2
[F4]	3	2	3
[F5]	3	2	2
[F6]	5	4	4
[F7]	4	4	3
[F8]	4	5	1
[F9]	3	3	2
[F10]	5	4	2

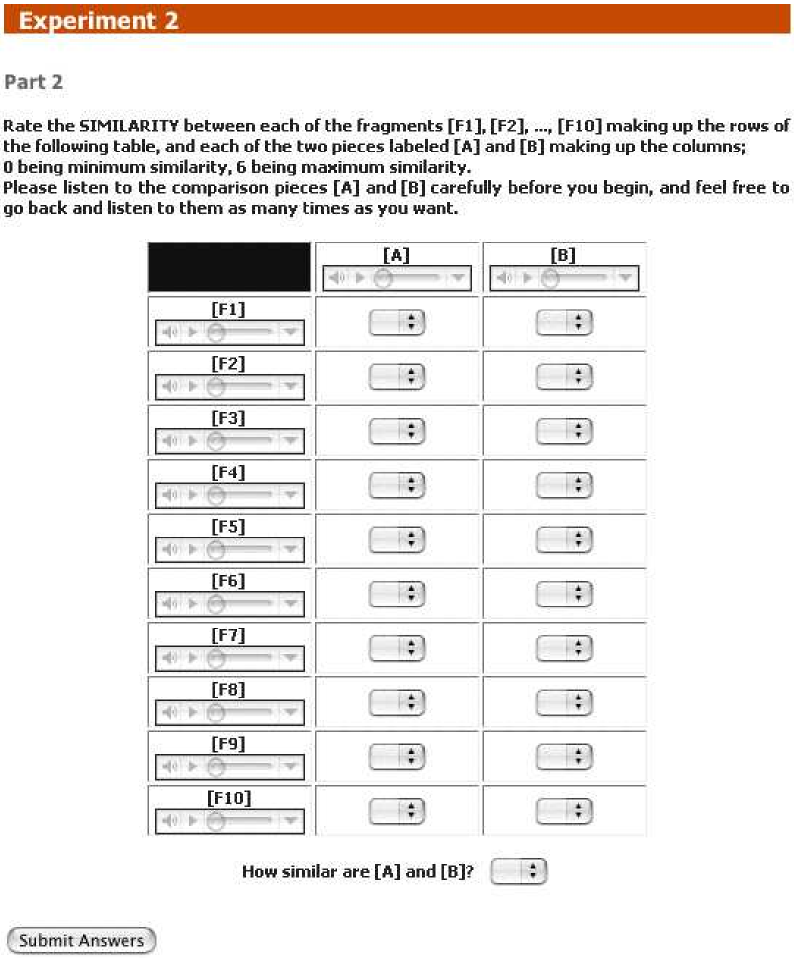
How familiarized are you with the music you just heard?

never heard any of it before ○

sounds familiar ●

i know i've heard it before, but don't know what it is ○

i know exactly what one (some) of them is (are) ○ give name(s)



Experiment 2

Part 2

Rate the SIMILARITY between each of the fragments [F1], [F2], ..., [F10] making up the rows of the following table, and each of the two pieces labeled [A] and [B] making up the columns;
0 being minimum similarity, 6 being maximum similarity.
Please listen to the comparison pieces [A] and [B] carefully before you begin, and feel free to go back and listen to them as many times as you want.

	[A]	[B]
[F1]		
[F2]		
[F3]		
[F4]		
[F5]		
[F6]		
[F7]		
[F8]		
[F9]		
[F10]		

How similar are [A] and [B]?

Bibliography

[1] V. Adán. Affine and stochastic transformations in ”rotaciones”. 2003.

[2] A. Aguilera and R. Pérez-Aguila. General n-dimensional rotations. In R. Scopigno and V. Skala, editors, *Journal of WSCG*, pages 1–8. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Science Press, February 2004.

[3] G. Assayag and S. Dubnov. Using factor oracles for machine improvisation. *Soft Computing*, pages 1–7, 2004.

[4] F. P. Brooks, A. L. Hopkins, P. G. Neumann, and W. V. Wright. An experiment in musical composition. In S. Schwanauer and D. Levitt, editors, *Machine Models of Music.* MIT Press, 1993.

[5] L. Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D*, 1997.

[6] C. Chatfield. *The Analysis of Time Series: An Introduction.* Text in Statistical Science Series. Chapman and Hall/CRC Press, 2004.

[7] D. Conklin and I. Witten. Multiple viewpoint systems for music prediction. In *Journal of New Music Research.* Swets and Zeitlinger, 1995.

[8] D. Cope. On algorithmic representation of musical style. In M. Balaban, K. Ebcioglu, and O. Laske, editors, *Understanding Music with AI: Perspectives on music cognition.* The AAAI Press/MIT Press, 1992.

[9] D. Cope. A computer model of music composition. In S. Schwanauer and D. Levitt, editors, *Machine Models of Music.* MIT Press, 1993.

[10] J. De Bonet and P. Viola. A non-parametric multi-scale statistical model for natural images. *Advances in Neural Information Processing*, 10, 1997.

[11] M. Dirst and A. Weigend. Baroque forecasting: On completing j. s. bach's last fugue. In A. Weigend and N. Gershenfeld, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past*, volume XV of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 151–172. Addison-Wesley, 1993.

[12] K. Ebcioglu. An expert system for harmonizing chorales in the style of j.s.bach. In O. Laske and M. Balaban, editors, *Understanding Music with AI: Perspectives on music cognition.* The AAAI Press/MIT Press, 1992.

[13] T. Eerola and P. Toiviainen. *MIDI Toolbox: MATLAB Tools for Music Research.* Department of Music, University of Jyväskylä, Finland, 2004.

[14] J. Estrada. *Théorie de la composition : discontinuum continuum.* PhD thesis, l'Université de Strasbourg II, Sciences Humaines, France, 1994.

[15] J. Estrada. El imaginario profundo frente a la música como lenguaje. *XXI Coloquio Internacional de Historia del Arte: La abolición del Arte*, 2000.

[16] J. Estrada. Focusing on freedom and movement in music: Methods of transcription inside a continuum of rhythm and sound. In J. R. Benjamin Boretz, Robert Morris, editor, *Perspectives of New Music*, volume 40, pages 70–91. Perspectives of New Music, 2002.

[17] A. Forte. *The Structure of Atonal Music.* Yale University press, 1973.

[18] N. Gershenfeld. *The Nature of Mathematical Modeling.* Cambridge University Press, 1999.

[19] S. Gill. A technique for composition of music in a computer. In S. Schwanauer and D. Levitt, editors, *Machine Models of Music.* MIT Press, 1993.

[20] A. Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2), 1995.

[21] L. Hiller and L. Isaacson. Musical composition with a high-speed digital computer. In S. Schwanauer and D. Levitt, editors, *Machine Models of Music.* MIT Press, 1993.

[22] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis.* Cambridge University Press, 2004.

[23] M. B. Kennel, R. Brown, and H. D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physics Review*, 1992.

[24] D. Kugiumtzis, B. Lillekjendlie, and N. Christophersen. Chaotic time series part i: Estimation of some invariant properties in state space. *Modeling, Identification and Control*, 15(4):205–224, 1994.

[25] B. Lillekjendlie, D. Kugiumtzis, and N. Christophersen. Chaotic time series part ii: System identification and predition. *Modeling, Identification and Control*, 15(4):225–243, 1994.

[26] D. Loy. Composing with computers: A survey of some compositional formalisms and music programming languages. In M. Mathews and J. Pierce, editors, *Current Directions in Computer Music Research.* MIT Press, 1989.

[27] S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1998.

[28] E. R. Miranda. *Composing music with computers.* Music Technology Series. Focal Press, 2001.

[29] M. Mozer. Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing. In N. Griffith and P. Todd, editors, *Musical Networks: parallel distributed perception and performance.* MIT Press, 1999.

[30] F. Pachet. The continuator: Musical interaction with style. In ICMA, editor, *Proceedings of ICMC*, pages 211–218. ICMA, 2002.

[31] T. Pankhurst. Schenker guide.

[32] G. Rader. A method of composing simple traditional music by computer. In S. Schwanauer and D. Levitt, editors, *Machine Models of Music.* MIT Press, 1993.

[33] J. Rissanen. Stochastic complexity and modelling., 1986.

[34] R. Rowe. *Machine Musicianship.* MIT Press, 2001.

[35] R. Shepard. Pitch perception and measurement. In P. Cook, editor, *Music, Cognition and Computerized Sound.* MIT Press, 1999.

[36] J. Shlens. A tutorial on principal component analysis. March 2003.

[37] B. Shoner. State reconstruction for determining predictability in driven nonlinear acoustical systems. Master's thesis, MIT, May 1996.

[38] H. Simon and R. Sumner. Pattern in music. In S. M. Schwanauer and D. A. Levitt, editors, *Machine Models of Music*. MIT Press, 1993.

[39] L. Smith. The maintainance of uncertainty.

[40] F. Takens. Detecting strange attractors in turbulence. In *Lecture Notes in Mathematica*. Springer, 1981.

[41] J. Trevino-Rodriguez and R. Morales-Bueno. Using multiattribute preduction suffix graphs to predict and generate music. In *Computer Music Journal*, number 25:3, pages 62–79. MIT Press, 2001.

[42] B. Vercoe. Understanding csound's spectral data types. In R. Boulanger, editor, *The Csound Book*. MIT Press, 2000.

[43] I. Xenakis. *Formalized Music*. Pendragon Press, 1992.