Monterey Mirror: Combining Markov Models, Genetic Algorithms, and Power Laws

An Experiment in Interactive Evolutionary Music Performance

Bill Manaris
Computer Science Department
College of Charleston
Charleston, SC, USA
manaris@cs.cofc.edu

Dana Hughes
Computer Science Department
College of Charleston
Charleston, SC, USA
dana.hughes@cs.cofc.edu

Yiorgos Vassilandonakis
Music Department
College of Charleston
Charleston, SC, USA
vassilandonakisy@cofc.edu

***Abstract*—Monterey Mirror is an interactive stochastic music generator based on Markov models, genetic algorithms, and power-law metrics for music information retrieval. It combines the predictive power of Markov models with the innovative power of genetic algorithms, using power-law metrics for fitness evaluation. These metrics have been developed and refined in a decade-long project, which explores music information retrieval based on Zipf's law and related power laws. We describe the architecture of Monterey Mirror, which can generate musical responses based on aesthetic variations of user input. We also explore how such a system may be used as a musical meta-instrument / environment in avant-garde music composition and performance projects.**

Keywords – genetic algorithms, Markov models, Zipf's law, power laws, stochastic music, music performance, music composition.

I. Introduction

Music, over the thousands of years of its existence, has been influenced considerably by mathematical and technological advances in society. These advances have contributed, for instance, to the development of new instruments (e.g., brass instruments, and the violin), to new ways of formulating music (e.g., equal temperament, and stochastic models), and finally, within the last 50 years, to automated, computational techniques contributing to the analysis, generation, composition, and performance of music.

Music narrative in Western music is traditionally advanced through an expansive process, shaped by transforming a few central musical gestures (themes) into larger sections, according to pre-conceived formal designs (i.e., sonata form, or theme & variations). The listener is able to follow the narrative by staying connected to the various transformations of the thematic material, and by being familiar with the general structure of the formal designs employed by the composer. This music language is closely linked to tonality, which fades by the beginning of the 20th century, at which point composers seek out alternative ways of organizing musical material. The importance of thematic unity and pre-conceived formal design diminish considerably, giving way to novel formal plans that are more ambitious in design, scope and complexity, and are shaped by organizing musical parameters previously considered secondary, such as texture, timbre, registral placement and density.

In the 1960's, composers like Gyorgy Ligeti and Iannis Xenakis started experimenting with cloud-like textures and dense sound masses, where the number of sound particles is so high that the listener cannot possibly follow their individual trajectories, and therefore concentrates on tracing the behavior of the overall texture and its shifts over time (e.g., [1]). From the perspective of the composer, it became increasingly difficult to plan and execute such complex textures without using computers. Iannis Xenakis, coming from an engineering and mathematics background was among the first composers to turn to computer-generated algorithmic processes in order to calculate the behavior of a large number of sound particles. Another important stylistic development of the late 1960's was the use of aleatoric principles in composition, as a reaction to the serialist tendencies of the post-war era, which exercised total control of the pitch and rhythmic material. Composer John Cage introduced a stylistically important new idea, according to which the composer does not seek to control every minute detail of a composition, but instead leaves certain elements to be decided by the performer, or by pure chance [2]. This development introduces improvisational freedom within certain layers of an otherwise through-composed piece, yet maintains control of the overall texture, character and direction of the composition. Later composers like Witold Lutoslawski adapted these ideas and personalized the use of chance elements in orchestra pieces, otherwise conventional in nature. Finally, as computer science and artificial intelligence techniques matured during the 1980's and 90's, a new compositional trend emerged in contemporary music, combining elements of aleatorism and algorithmic processes to provide the contemporary composer with choices unavailable before.

Computer-assisted composition and performance involves the use of some process to generate new musical material or expand upon existing one. These processes can assist in exploring material more rapidly and with considerable more precision and detail than one would be able to do alone, and can furthermore steer the composer toward entirely new explorations. They also provide a considerable increase in the level of control of large-scale planning and management of musical gestures, events and formal structures, especially as the latter get more complex and involve a large number of elements.

This paper presents an innovative system, called *Monterey Mirror*, which exists in the intersection of techniques for the

Appears in Proceedings of 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011), New Orleans, LA, USA, June 5, 2011, pp. 33-40.

analysis, generation, composition, and performance of music. Monterey Mirror combines Markov models, genetic algorithms, and power laws to create an environment, which can be viewed both as

- a music meta-instrument, in that it can be used in real-time (through a well-defined MIDI interface) to generate music interactively, and
- a compositional space, in that it introduces new possibilities to shaping music form and interactions between human and artificial musicians.

Section 2 presents related systems and background research. Section 3 describes the architecture of Monterey Mirror. Section 4 discusses how Monterey Mirror may be used in contemporary music composition. Section 5 presents a case study, i.e., one particular musical piece performed with two human performers and two Monterey Mirrors. The paper closes with concluding remarks and future work.

II. Background

As already mentioned, Monterey Mirror combines Markov models, genetic algorithms, and power-law metrics. Since there are two target audiences for this paper, computer scientists and music composers, this section provides an outline of the essential aspects of these techniques. Additionally, we discuss relevant earlier work in algorithmic music composition and performance.

A. Markov Models

Markov models are stochastic systems, which capture probabilities of the occurrence of particular events based on earlier context [3].

Markov models can be trained using existing sequences of events (e.g., words in a book, or notes in a musical piece). The training algorithm constructs an n-dimensional matrix (either explicitly or implicitly) which records how often one subsequence of events resolves into a given event.

Once trained, a Markov model can be used to generate a new sequence of events that is statistically similar to the sequences of events used for training. It is possible for a Markov model to recreate an exact training sequence, but it is highly improbable, for all practical purposes (i.e., it depends on the training corpus, and the different ways particular subsequences of events resolve).

An important limitation of Markov models, which makes them not as interesting for generating music and art, is that the statistical similarity between the training material and the generated material is local. In other words, Markov models only capture short-term dependencies in training material (as opposed to long-term dependencies). For instance, it is quite common in music to have long-term dependencies, such as the ABA, and ABBA high-level forms. In other words, the material used in the beginning of a song (i.e., initial section A) is repeated (perhaps with some variation) at the end (i.e., final section A). This type of dependency is quite common in musical form, but it is not explicitly captured by Markov models. It possible for a Markov model trained on music with, say, ABBA form, to generate a sequence with this form, since the training material contains local transitions that may lead to such a sequence. However, the chances of arriving at such an outcome are exponentially small.

Our approach uses genetic algorithms to evolve Markov model output to discover precisely such rare outcomes that have desired long-term dependencies, in addition to the short-term dependencies that are guaranteed through the Markov process.

B. Genetic Algorithms

A *genetic algorithm* is a computational approach for building sequences of symbols (e.g., musical phrases) that have certain desirable properties [4]. These properties are evaluated through a *fitness function*, which computes measurements specific to the task at hand (e.g., measurable aspect(s) of musical aesthetics). These measurements estimate the distance of a constructed artifact from the best possible artifact(s).

A genetic algorithm progresses by constructing generations of individual artifacts. Each generation is produced by evolving individuals from the previous generation. The first generation is initialized in some random way. The best individuals from each generation (also know as *the elite*) are preserved; they are passed unchanged to the next generation. This serves as a short-term memory of the best "solutions" found so far. The remaining (non-elite) individuals are modified through well-defined operations, such as mutation and crossover. Mutation randomly changes parts of an individual. Crossover swaps parts between two randomly-selected *parent* individuals.

This process continues until the fitness function identifies an individual that is close enough to the desired outcome, or after a number of generations has passed without finding such an individual.

There are many adjustable parameters in a genetic algorithm, including the set of symbols (i.e., *alphabet*) used to construct individuals (also known as *genotypes*); how to convert these individuals to "real-world" artifacts (i.e., *phenotypes*); the number of individuals per generation; what percent of individuals to preserve as the elite; how often to mutate an individual; how often to crossover two individuals; and so on.

Clearly, a very important consideration is what aspects of the "solution" to capture in the fitness function, and how to go about measuring them. This last point is most essential, as the fitness function guides the evolution process. If the fitness function leaves certain important aspects unmeasured (i.e., has "holes"), the genetic process will exploit these "holes" to arrive at solutions that are substandard (similarly to water finding the path of least resistance).

Our approach uses fitness functions consisting of power-law metrics, which have been shown to capture essential aspects of musical aesthetics. These are introduced below.

A significant advantage of genetic algorithms is that they search a large space of possibilities quickly in search of optimal "solutions". Each individual in a genetic algorithm's population is exploring a different part of the search space. Individuals with more desirable characteristics (i.e., the elite) get to "attract" other individuals to their part of the search space (by surviving across generations and, thus, being more likely to contribute their genetic material to other individuals via crossover). Additionally, the mutation operation "moves" individuals into new areas of the search space, potentially arriving closer to new, more desirable solutions.

Genetic algorithms are very efficient for searching large spaces of possibilities. However, if this space is astronomically large (as is the case with evolving musical pieces), it is difficult to find desirable solutions in a practical amount of time. This problem becomes more pronounced with interactive, real-time music situations, as is the case with Monterey Mirror.

To deal with this problem some constraints (or approximations) have to be introduced.

C. Algorithmic Composition

The development of electronic computers has enabled a wide variety of computer programs within the last 50 years, performing aspects of analysis, generation, composition, and performance of music [5-13]. Approaches similar to Monterey Mirror are briefly described here.

GenJam uses a genetic algorithm to generate jazz improvisations [5]. The developer of this system has demonstrated its ability to provide accompaniment during performance. However, the system must be trained off-line, and requires a human to judge the quality of evolved melodies.

SWARMUSIC utilizes a swarm simulator to generate music [6]. This system can be used independently, or by responding to user input, and may be used in an interactive manner. The system generates free, rather than form-based, improvisations.

Experiments in Musical Intelligence (Emmy) is a system that generates novel musical pieces using Markov models [7]. This system works off-line. It has generated several pieces in the style of various composers, including Bach and Chopin.

Continuator is an interactive music performance system that accepts partial input from a human musician and continues in the same style as the input [8]. The system utilizes various Markov models to learn from the user input. The Markov models are organized in decreasing order of information retrained, i.e., the first one remembers everything the user has played (e.g., note pitches, durations, start times, etc.). Subsequent Markov models retain less information. The system tries to provide an answer from the most-informed Markov model, and if a match is not found with the user input, the system continues with the less-informed Markov models. Thus, the system sometimes generates accurate "reproductions" (or continuations) of the user input, and other times less accurate ones (which sound as interesting variations).

Finally, *NEvMuse* is a system combining evolutionary computation with artificial music critics to evolve variations of musical excerpts [9]. NEvMuse makes extensive use of power-law metrics, which measure statistical proportions of musical artifacts. Power-law metrics are based on Zipf's law, and have been shown to capture essential aspects of music aesthetics. In particular, using power-law metrics as features, earlier experiments with artificial neural networks have demonstrated high accuracies, e.g.., for composer identification, 93.6% - 95% accuracy; style classification, 71.5% - 96.6% accuracy; and pleasantness prediction, 90.7% accuracy. Finally, several psychological experiments (e.g., pleasantness prediction, popularity prediction, and similarity prediction) have demonstrated the aesthetic relevance of power-law metrics relative to human listeners [9, 14-16].

Our approach with Monterey Mirror combines elements of the Continuator (Markov models for interactive music performance) and NEvMuse` (evolutionary computation and power-law metrics), as discussed in Section III.

D. Power-Law Metrics

Power laws are statistical models of proportions exhibited by various natural and artificial phenomena. Zipf's law is a special type of a power law, where the probability of a certain event occurring, $P(f)$, is related to the frequency of occurrence, f, by the following equation:

$$P(f) = 1/f \tag{1}$$

This equation describes a unique property of Zipfian phenomena, where the 2nd most-frequent event appears 1/2 as many times as the 1st most frequent event; the 3rd most-frequent event appears 1/3 as many times; the 4th appears 1/4 as many times, and so on. The emergence of the harmonic series in Zipfian artifacts (i.e., 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, etc.) is believed to be of significance with respect to how nature works. This intriguing relationship (regularity) is found in many human and natural phenomena, including music, city sizes, incomes, subroutine calls, earthquake magnitudes, thickness of sediment depositions, opening chess moves, textbooks, extinctions of species, traffic jams, and visits to websites [17-19].

Music, in particular, exhibits near-Zipfian proportions across many dimensions (e.g., pitch, duration, harmonic intervals, distance of repeated notes, etc.). Moreover, power-law behavior (e.g., numeric deviations from the ideal Zipfian proportions) across a multitude of attributes has been shown to be of aesthetic significance [14-16]. Power-law proportions (i.e., vectors of power-law measurements across numerous dimensions) extracted from musical pieces have been used successfully in the past for identifying the contextual information of a given piece, such as author and genre.

III. System Architecture

Monterey Mirror is designed to generate novel responses to user-provided musical gestures and ideas (melodic, rhythmic, and harmonic) in real time. The system is divided into two main components, the *Listener* and the *Mirror*. The Listener interacts directly with the user to receive and process events generated by MIDI instruments, and to play generated musical responses. The Mirror generates the musical responses when queried by the Listener, utilizing a Markov model, a genetic algorithm, and numerous power-law metrics for fitness.

A. Listener

The Listener is an event-driven component, which processes MIDI messages and constructs higher-level note objects (consisting of pitch, rhythm, volume, and channel).

Additionally, the user may trigger the Listener to perform one of the following four actions:

- **ListenOn** - The Listener begins to record the user's melody (i.e., sequence of MIDI events) for use as basis for the Mirror's response. The pitch, rhythm and dynamic value of notes played by the user are added to the Listener's local memory. The Listener's local memory is flushed at every ListenOn trigger, allowing the user to generate a new melody for future use.
- **ListenOff** - The Listener stops recording notes played by the user. The local memory is maintained for future use.
- **Respond** - The Listener asks the Mirror to generate a response. The Listener provides the Mirror with the current local memory (i.e., the user's most recently played musical material) for use as learning information, and as a basis for generating a response.
- **Forget** - The Listener instructs the Mirror to forget any previously learned material. This allows learning new material (e.g., user style).

The Listener listens concurrently to the user while requesting and playing Mirror responses, so that no user material is lost.

The Listener interacts with the Mirror through the following API:

- **Learn** - The Listener provides the Mirror with previously unlearned portions of user musical material.
- **Forget** - The Listener instructs the Mirror to forget any previously learned musical material.
- **RequestResponse** - The Listener passes user musical material and requests a Mirror response to be generated.

B. Mirror

The Mirror component is responsible for generating responses to provided musical material. In contrast to the Continuator system (see Background), the Mirror provides "answers" to user "questions". This is more in line with what users usually expect [8].

As mentioned earlier, the Mirror consists of three components, a Markov model, a genetic algorithm and a fitness function.

The Markov model captures the transition probabilities in the provided user input (such as pitch, rhythm and timing information). This model maintains information on all the learning material provided. This provides the Mirror with an engine for rapidly generating musical phrases. The implied musical structure of the user's input is maintained after each learning phase. This allows the user to build a dynamic "relationship" with the Mirror, which can be honed, developed and "learned" over time, similar to how a chamber group of musicians trains to react to each other's musical gestures in the process of rehearsing.

The Genetic Algorithm component allows the Mirror to evolve a population of musical responses. An initial population of phrases is provided by the Markov model. Consequently, each individual in the population will resemble the user's implied musical style. Each individual contains a variable-length genotype, avoiding placing any constraint on the length of the material generated. The genotype represents a potential Mirror response.

The population is evolved through several generations, using crossover and mutation operations on each individual's genotype. The crossover operation recombines material between two individuals in a way that is consistent with the Markov model (i.e., it could have been generated by the Markov model itself, given enough generation attempts). Mutation of individuals involves regenerating a portion of the genotype using the Markov model.

At each generation, the evaluation function calculates the fitness of each individual by determining the aesthetic similarity between the individual's melody and the user input. There are other possibilities (such as comparing each individual to material by other composers/performers, e.g., J.S. Bach or Thelonious Monk). This way, the system may create a composite Mirror, i.e., a mirror combining elements from the user and other (even non-living) musicians.

The genetic algorithm seeks to maximize each individual's similarity (while avoiding identity) to the user input. A certain percentage of the population with the highest similarity (i.e., elite population) is maintained during each generation to ensure that generated phrases do not degrade.

The fitness function of the Mirror calculates hundreds of power-law metrics for each individual melody as well as for the user input. The similarity of each individual to the user input is determined by calculating the mean-square error (MSE) of the user input metrics and the individual's metrics. Individuals with a low MSE are more similar to the user input than those with a high MSE.

After evolving a fixed number of generations, the best response, i.e., the response most similar (but not identical) to the user input) is output.

IV. MUSIC DESIGN AND MONTEREY MIRROR

In musical terms, Monterey Mirror can be described as an interactive stochastic music generator, able to interact with human musical creativity and return intelligent and meaningful responses, based on variations within the user's input. As a material generator, a compositional environment designer, and a meta-instrument with interactive capabilities, Monterey Mirror is a powerful tool for contemporary music creation, targeted toward two main areas: compositional design and interactive improvisation.

The most obvious ways of tapping into Monterey Mirror's possibilities are within the context of (structured or free) improvisation, by exploring its capacity of generating material much like a human performer would, as well as an intelligent meta-instrument, that can be learned and performed upon, much like a violin. A performer will quickly realize and take musical advantage of the system's ability to "listen" to a phrase and "answer" it with a meaningful musical response that can then be interacted with to create a musical dialog. It's very interesting to experiment with altering specific elements of the input material, in order to elicit a specific response from the system. With a little practice, the performer can easily develop a vocabulary of interactive gestures that can be integrated into a wide variety of performance situations in almost any genre. In improvisational performance settings, a performer can use Monterey Mirror to enrich his individual output and create the illusion of richer textures and more polyphony.

In its role as a meta-instrument, Monterey Mirror can be integrated within a musical ensemble of mixed media (acoustic and computer-generated), or could be employed as a stand-alone system for performance. Within the context of a mixed media ensemble, new possibilities of orchestration and ensemble interaction become available, by exploring the concept of *multiplicity*, where one performer can appear to be producing several strands of independently controlled material with the help of the system. In this scenario, it is not only the instrument that gets multiplied, but also the performer, along with her personal idiosyncrasies, training and approaches to technique, phrasing, and timbre.

Within the context of composition, Monterey Mirror can function as a planning template of considerable insight in exploring material and its possibilities within the musical space. It can allow a composer to tangibly blow up and time-stretch gestures and musical cells into complex textures of vast spatial and formal scope, while maintaining close relationship to the generating material, allowing for precise control, shaping and shifting power of the total musical space (e.g., see [1], pp. 37-42).

Monterey Mirror also presents new possibilities with visualizing and realizing formal designs, where the composer can maintain tight control of certain musical parameters (registral orientation & trajectories, temporal blocking and timbral identities), and simultaneously be relaxed, almost improvisational in nature with others, (pitch patterns and rhythmic permutations), because the internal architecture of the system will preserve the integrity of the material that was fed into it. This allows for a new kind of aleatoric approach to form that is not possible otherwise.

V. Evaluation: Composing 2x2

2X2 is a composition intended to demonstrate some of Monterey Mirror's possibilities. It explores modes of interaction between two performers and two Monterey Mirror systems, within a carefully planned compositional space.

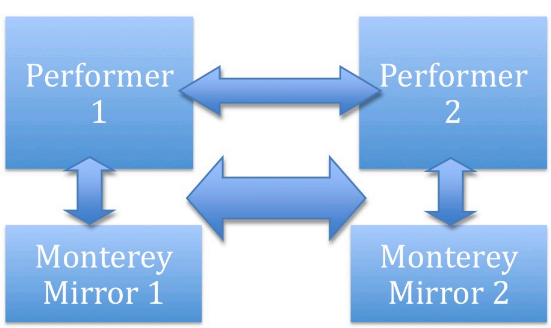


Figure 1. *2x2* setup.

The setup (see Fig. 1) calls for two performers, playing MIDI controllers of any kind (keyboard, guitar, wind or mallet based), each connected via MIDI to a separate laptop running a Monterey Mirror system. This setup allows for the following modes of interaction: performer to performer, performer to Monterey Mirror system, and duo1 (performer1 & MM1) to duo2 (performer2 & MM2).

Both performer and system MIDI controller outputs play the same sound, and all audio outputs are routed through the same speaker setup, purposefully blurring the origin of the material to the listener, who cannot locate aurally whether sounds are coming from a human or are computer generated.

The form of the piece (to be demonstrated at the conference) combines aleatoric/quasi-improvisational properties in the surface structure with precise large-scale form design.

As seen in the Appendix, the piece is globally organized in 19 sections, each successive one introducing a new combination of interactivity, with a clear overall direction to a climactic point (section 9), where all possible combinations of material are superimposed, creating the most complex texture of the piece. The piece comes to a close (section 19) with a coda that features interaction between the two performers only, with no additional output from the Monterey Mirror systems.

The musical material within each section is propagated through a "listening" process. One of the performers acts as an initiator of new material (sections 1, 3, 6, 7, 11, 13, 15-17), and the system connected to his individual MIDI controller is triggered to "listen" to his material, process it, and then later triggered to play material that is based on the section it has processed. As it plays processed material, it can be triggered to "listen" concurrently, and either combine new material with previously learned one, or "dump" older material, and start anew. The duration of each section is fluid and controlled by the performers (it is suggested to be between 15-25"). The Monterey Mirror systems will output material lasting about the same time as the one they listened to, therefore, the performers can anticipate and shape each section by interacting with each other and create smooth transitions moving forward.

A. Discussion

2X2 is concerned with Monterey Mirror's interactive applications as applied to a duet of performers. There are endless possibilities to be further explored in future compositions, for which the system can be adjusted and tailored to accommodate other kinds of input material, pre-planned or not. There can be human initiation, or input from sources of any kind, including randomly generated events, or non-musical processes providing data, which is in turn applied to generate sound. Monterey Mirror would also work great for cite-specific sound installations, or sensor-based interactive systems.

Within a more traditional electro-acoustic composition environment, the system can be adapted to have triggers programmed to fire when they encounter a specific event (e.g., MIDI note) in its input, so that a performer can just perform off a traditional score without having to additionally trigger events, as the system will be "waiting" for a specific note as a trigger.

MIDI controllers have also a lot of potential in similar scenarios, and elements like MIDI volume, panning, portamento, etc. can be "listened to" and incorporated into the compositional design.

The ability to operate a Monterey Mirror with a portable laptop setup and relatively little computing power can allow for larger ensembles incorporating several systems, multiplying the sonic possibilities, or even eliminating performers altogether and setting up multiple systems which can feed each other in a loop-like situation.

VI. Conclusion and Future Work

Monterey Mirror is an interactive stochastic music generator based on Markov models, genetic algorithms, and power-law metrics. It combines the predictive power of Markov models with the innovative power of genetic algorithms, using power-law metrics for fitness evaluation. Unlike other systems, the evolutionary process is confined within the search space defined by the Markov model. Initially, this might appear as a limitation. It would have been straightforward to include random mutation and crossover operations to generate "novel" material. However, given the real-time constraint of Monterey Mirror, it was a conscious decision to evolve material consistent with the Markov model. Since Markov models only capture local dependencies, they tend to generate output lacking the long-term structure or coherence of the input (training) material. However, once in a while the Markov model may "get lucky" and generate an output that exhibits such dependencies. It is precisely this possibility that the genetic algorithm exploits. By laboriously exploring the search space defined by the Markov model, it eventually will encounter material exhibiting longer-term dependencies. The fitness function ensures that such individuals are sought and rewarded, given that power-law metrics capture long-term dependencies in musical data [9].

The use of power-law metrics allows for evaluation of the generated musical phrases during performance. There are two distinct advantages to this. First, Monterey Mirror requires no training prior to performance, and does not require interactive evaluation of generated material. Second, because evaluation relies solely on the statistical proportions of the input material, the system is not limited to a certain style of music.

In musical terms, Monterey Mirror is able to interact with human musical creativity and return intelligent and meaningful responses, based on variations within the user's input. Both as a compositional environment, and as a meta-instrument with interactive capabilities, Monterey Mirror is a powerful tool for contemporary music creation, targeted toward two main areas: compositional design and interactive improvisation.

Acknowledgements

This work has been supported in part by NSF (grants IIS-0736480, IIS-0849499 and IIS-1049554). The first author would like to acknowledge the contribution of David Cope, Peter Elsie, Daniel Brown, and Michael O'Bannon in shaping the Monterey Mirror concept during the 2010 Workshop in Algorithmic Computer Music (WACM) at the Univ. of California, Santa Cruz. We also like to acknowledge the following for their contribution in formulating and evaluating power-law metrics: Thomas Zalonis, Brys Sepulveda, Patrick Roos, Dwight Krehbiel, Luca Pellicoro, Timothy Hirzel, Penousal Machado, and Juan Romero.

References

[1] I. Xenakis, *Formalized Music: Thought and Mathematics in Music*, Pendagon Press, 1992.

[2] J. Cage, *Silence; Lectures and Writings of John Cage*, Wesleyan University Press, 1961.

[3] L.R. Rabiner and B.H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4-16, 1986.

[4] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[5] J.A. Biles, "GenJam: A genetic algorithm for generating jazz solos," *Proceedings of the International Computer Music Conference*, Aarhus, Jutland, Denmark, pp. 131-137, 1994.

[6] T.M. Blackwell and P. Bentley, "Improvised music with swarms," *Proceedings of the Congress Evolutionary Computation (CEC02)*, vol. 2, pp. 1462-1467, 2002.

[7] D. Cope, *Virtual Music: Computer Synthesis of Musical Style*, MIT Press, 2001.

[8] F. Pachet, "Playing with virtual musicians: the Continuator in practice," IEEE Multimedia, vol. 9, no. 3, pp. 77-82, 2002.

[9] B. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro and J. Romero, "A corpus-based hybrid approach to music analysis and composition," *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*, Vancouver, BC, pp. 839-845, July 2007.

[10] G. Wiggins, G. Papaopoulos, S. Phon-Amnuaisuk, and A. Tuson, "Evolutionary methods for musical composition," *International Journal of Computer Anticipatory Systems*, vol. 1, no. 1, 1999.

[11] A.R. Brown, "Opportunities for Evolutionary Music Composition," *Proceedings Australasian Computer Music Conference*, Melbourne, Austrialia, pp. 27-34, 2002.

[12] J. McCormack, "Grammar-Based Music Composition," *Complex systems 96: from local interactions to global phenomena,* ISO Press, Amsterdam, The Netherlands, pp. 321-336, 1996.

[13] D. Burraston, E. Edmonds, D. Livingstone, and E.R. Miranda, "Cellular automata in MIDI based computer music," *Proceedings of the International Computer Music Conference (ICMC2004),* pp. 71-78, 2004.

[14] B. Manaris, J. Romero, P. Machado, D. Krehbiel, T. Hirzel, W. Pharr and R.B. Davis, "Zipf's law, music classification and aesthetics," *Computer Music Journal*, vol. 29, no. 1, pp. 55-69, MIT Press, Spring 2005.

[15] B. Manaris, J.R. Armstrong, T. Zalonis, and D. Krehbiel, "Armonique: a framework for web audio archiving, searching, and metadata extraction," *International Association of Sound and Audiovisual Archives (IASA) Journal*, vol. 35, pp. 57-68, June 2010.

[16] B. Manaris, P. Machado, C. McCauley, J. Romero, and D. Krehbiel, "Developing fitness functions for pleasant music: Zipf's law and interactive evolution systems," *EvoMUSART2005 – 3rd European Workshop on Evolutionary Music and Art*, Lausanne, Switzerland, *Lecture Notes in Computer Science, Applications of Evolutionary Computing, LNCS 3449*, Springer-Verlag, pp. 498-507, March 2005.

[17] G.K. Zipf, *Human Behavior and the Principle of Least Effort*, Hafner Publishing Company, 1949.

[18] R.F. Voss, and J. Clarke, "'1/f noise' in music and speech," *Nature*, vol. 258, no. 5533, pp. 317-318, Nov. 27, 1975.

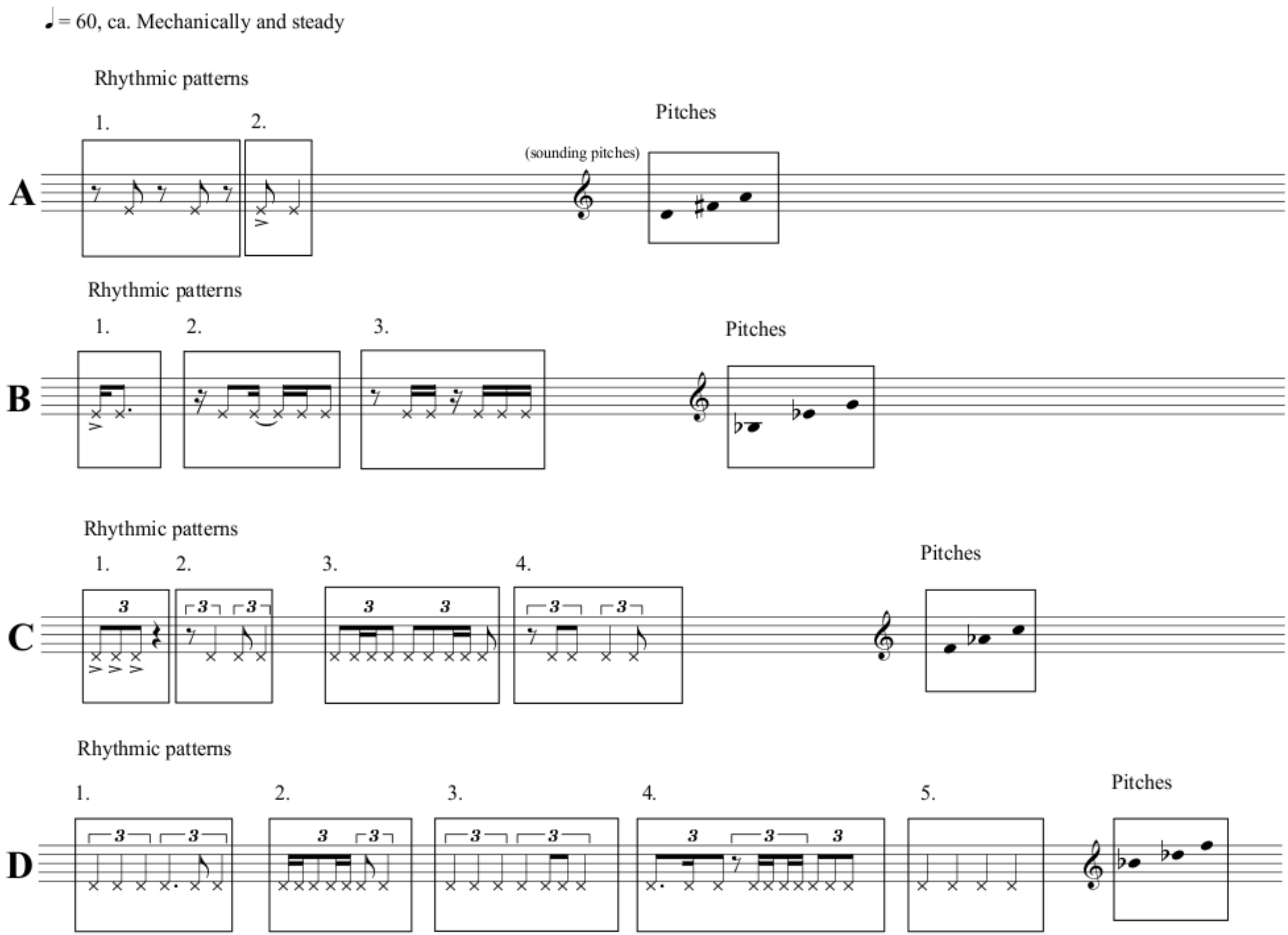
[19] M. Schroeder, *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*, W.H. Freeman, 1991.

APPENDIX: 2X2 FORMAL SCHEME

Formal Scheme

Section	1	2	3	4	5	6	7	8	9
Performer 1	**Play A**		**Play B**				**Play D**	**Play D**	**Play A**
System 1			Listen B	**Play B**			Listen D	Listen D	**Play D** Listen A
Performer 2		**Play A**	**Play A**		**Play B**	**Play C**		**Play C**	**Play C**
System 2	Preload long B						**Play B**	**Play B**	**Play B** Listen C

Section	10	11	12	13	14	15	16	17	18	19
Performer 1		**Play C**		**Play D**		**Play C**	**Play B**	**Play A**	**Play A**	**Play A**
System 1	**Play D+A**	Listen C	**Play C**	Listen D		**Play D** Listen C	**Play C** Listen B	**Play B** Listen A	**Play A**	
Performer 2		**Play D**		**Play D**		**Play C**	**Play B**	**Play A**	**Play A**	**Play A**
System 2	**Play B+C**			Listen D	**Play D**	Listen C	**Play C** Listen B	**Play B** Listen A	**Play A**	


♩ = 60, ca. Mechanically and steady
Rhythmic patterns
1.
2.
Pitches
(sounding pitches)
A
Rhythmic patterns
1.
2.
3.
Pitches
B
Rhythmic patterns
1.
2.
3.
4.
Pitches
C
Rhythmic patterns
1.
2.
3.
4.
5.
Pitches
D