

THE AIMS PROJECT: CREATIVE EXPERIMENTS IN MUSICAL SONIFICATION

Reginald Bain

University of South Carolina
Experimental Music Studio (xMUSE)
rbain@mozart.sc.edu

Abstract

The Applications in Musical Sonification (AIMS) project is dedicated to creative experiments in musical sonification. It studies sonification in the context of algorithmic composition, focusing on the design and implementation of these experiments in software. An initial distribution of cross-platform, open-source software is now freely available that implements some basic mappings, including historically significant compositional formalisms and mathematical models that are widely discussed throughout the literature. These applications can be used for instructional purposes or for experimentation with the use of sonifications as compositional determinants.

1 Introduction

In the field of auditory display, sonification is defined as “the use of non-speech audio to convey information.” (Kramer et al. 1997) In the context of algorithmic composition, however, the term sonification is sometimes used to refer to “the process of turning non-aural information into sound” for essentially artistic purposes. (Burke et al. 2005) The Applications in Musical Sonification (AIMS) project studies the latter, focusing on the design and implementation of creative experiments in musical sonification in software. The AIMS project also attempts to study the so-called mapping problem and to identify issues, concerns and strategies common to the fields of auditory display and algorithmic composition. Whereas previous studies such as Bargar (1994) and Childs (2003) have attempted to link compositional techniques and thought processes with the design of improved, more intuitive methods of auditory display, the AIMS project attempts to do just the opposite.

We use the term *musical sonification* here to refer to the composer’s creative engagement and exploration of the multiplicity of musical ideas that are the potential resultants of a given sonification. Using Cycling 74’s MaxMSP and other tools, we create custom software applications to implement these experiments. Each application explores a different mapping of an extra-musical idea, mathematical model, data set, and so forth, into the audio domain,

allowing us to study different aspects of the mapping problem.

A model for our work is Ben-Tal and Berger (2004), whose study of the creative aspects of sonification focuses on the listener’s ability to track changes occurring in variables associated with complex multidimensional data. As the model for their work is creative music listening, they also use the term musical sonification to describe their approach to the study of nonverbal categorical perception in the context of auditory scene analysis. (Bregman 1990) They propose that such abstractions and characterizations are “simplified models of creative engagement with sound.” What is more, Ben-Tal and Berger are composers who have used sonifications as compositional determinants in their music. Though we do share some of Ben-Tal and Berger’s research goals, it should again be noted that we are not concerned with the design of improved methods of auditory display. Rather, the primary goal of our research is to use the wealth of knowledge emerging from the field of auditory display to suggest new approaches to sonification in the context of algorithmic composition.

2 The Mapping Problem

Polansky (2002) succinctly describes the *mapping problem* in the following way: “an idea in one domain is manifested in [another].” Exactly how should one go about mapping extra-musical data to musical parameters such as pitch, amplitude and timbre? Given the artistic nature of the problem, it is perhaps not surprising that there is no easy answer to this question. In addition to describing the problem, Polansky also discusses the types of distinctions that should be made when determining what should rightly be called sonification and what should more appropriately be called algorithmic composition. Furthermore, he makes a distinction between *scientific sonification* and *artistic sonification*, proposing the term *manifestation* to refer to the latter when in his words, “the intent is clearly to use a formal or mathematical/formal process to create a new musical idea as a form of sonification.” (Polansky 2002) We hope to shed light on the mapping problem by methodically

studying it in a wide variety of creative experiments in musical sonification.

The following historical example is provided to help clarify what we mean by mapping problem, and to help demonstrate the difference between musical sonification and algorithmic composition in the context of the AIMS project. Charles Dodge's *Earth's Magnetic Field* (Dodge 1970) has frequently been cited as an example of sonification in the context of algorithmic composition. (e.g., Childs 2003) Dodge describes his composition as "a musical setting of the succession of values that are produced by an index of the effect of the sun's radiation on the magnetic field that surrounds the earth." (Dodge and Jerse 1997) Obviously, there are many legitimate ways to approach the mapping process, that is, many different ways to *sonify* the magnetic field indices in the musical domain. The problem of how to *interpret* the indices and ultimately map them to specific musical parameters is what we mean by the mapping problem. The experimental process of creating a variety of different musical realizations based on the same succession of indices is what we mean by musical sonification. The sonification experiment ends when the composer chooses a specific mapping for use in his composition. In *Earth's Magnetic Field*, when Dodge decides to map the 28 possible index values into a four-octave diatonic scale on C using a Meantone temperament. (Dodge and Jerse 1997)

3 Compositional Formalisms

Rather than begin our work with the sonification of natural functions produced by scientific data, we have chosen to begin our work with the study of compositional formalisms—classic implementations of algorithmic composition systems (Roads 1996) widely discussed throughout the literature. Loy (1989) describes compositional formalisms as ways of thinking. He says that procedures, algorithms, methods, and games are words we associate with the expression of musical formalisms. A formalism may be viewed as a systematic way of analyzing, ordering, or creating a compositional system. Of course, formalisms may or may not be algorithmic. Nonetheless, it seems that many traditional compositional formalisms do have an inherently algorithmic basis and are thus easy to implement on a general-purpose computer using a powerful real-time interactive composition environment like MaxMSP.

Loy (1989) discusses a wide variety of examples of compositional formalisms that are considered by many to be classic implementations of algorithmic composition systems including: Guido D'Arezzo's text-setting method from *Micrologus*, Medieval isorhythmic motets, English rounds, Renaissance musical acrostic techniques such as *soggetto cavato*, the art of canon, dice music attributed to Mozart, and more modern examples such as combinatorics in the music of Schoenberg, and chance techniques associated with the music of John Cage. He also discusses the role

formalisms have played in the work of Hiller and Isaacson, Koenig, Schillinger, and Xenakis, among others. We study such systems intensively, and then use them to provide appropriate deterministic and stochastic contexts for our initial experiments.

4 Application Design Issues

Designing intuitive, responsive, and easy to use human-computer interfaces for our musical sonification applications is quite a challenge. We have used Kramer (1997) and Winkler (1998) as guides in this area. De Campo, Frauenberger, and Holdrich (2004) has also been useful as it clearly identifies a number of features that a generalized sonification environment should possess. *String Length and Pitch Interval* (SLAPI) was selected to be the first application developed under the AIMS project because it presented numerous design and programming challenges characteristic of the musical sonification tools we plan to develop. Its interface is shown in Figure 1.

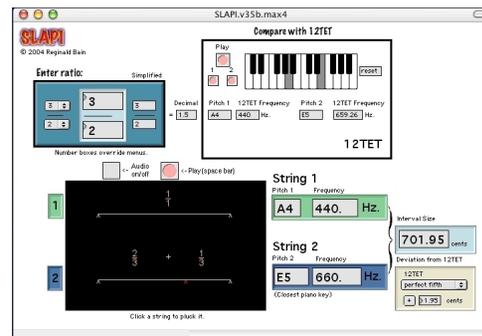


Figure 1. SLAPI—a just interval player application.

SLAPI offers students of tuning theory a simple way to "sonify" *interval frequency ratios* that result from rational string length divisions. These ratios are the basic building blocks of traditional tuning systems such as Pythagorean and just intonation. SLAPI's main function is to provide users with a convenient way to compare the "size" of just intervals (e.g. 3:2, 4:3, 5:4, and 9:8) with their respective twelve-tone equal tempered counterparts. After soliciting feedback from composers using early prototypes of SLAPI, we set out to design a monochord-like interface that would serve the needs of a standalone application. So that some appreciation may be gained of the level of complexity that quickly develops when one attempts to design a human interface for a MaxMSP application, Figure 2 shows SLAPI in MaxMSP's edit mode.

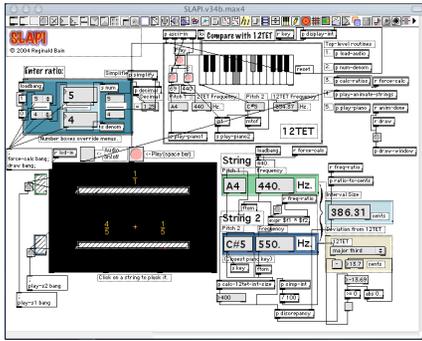


Figure 2. SLAPI shown in MaxMSP's edit mode.

It should be mentioned that most of SLAPI's functionality is encapsulated in objects a level below the top-level shown in Figure 2.

After our experience with the design of SLAPI's sample-based audio implementation, it was decided that the initial AIMS distribution would be implemented using MIDI techniques. Limiting the audio production context to MIDI greatly reduces the complexity of the design issues we must confront during this preliminary stage of the project. As such, this generation of the software only offers control over relatively high-level musical parameters like tempo, pitch collection, canonic interval of imitation, and so forth.

5 The Initial AIMS Distribution

The applications that comprise the initial AIMS distribution are fully documented, open-source and freely available. Most of the applications began life as pre-compositional experiments for full-length algorithmic compositions. Each application explores a single formalism in a melodic context. In the spirit of Xenakis (1992), the melodies and textures produced by these preliminary experiments focus on the rendering of mathematical data and formulations. The initial distribution includes the following applications: *Guido's Text-Setting Method*, *Messiaen's Communicable Language*, *Playing with Pi (And Other Constants)*, the *Gardner-Voss If/Dice Algorithm*, the *Logistic Difference Equation*, and the *Chaos Game*. A brief description of each application is provided below along with a citation from the literature that provides a more in-depth description of each formalism.

5.1 Historical Examples of Mappings

The following two historically significant examples of compositional formalisms explore simple isomorphisms of Latin and French texts, respectively, into melodies. In both cases, the application enables a user to produce a melody by interactively typing a text at the computer keyboard.

5.11 Guido's Text-Setting Method (1026). Guido D'Arezzo's method for setting a Latin text from *Micrologus* (D'Arezzo 1978) is often cited as one of the earliest

examples of a compositional formalism (Loy 1989 and Rowe 1993). It is a straight-forward mapping of the vowels of a Latin text to the notes of a scale to produce a chant-like melody. Guido's mapping, using modern pitch notation (C4 is middle C), is shown in Figure 3.

G2	A2	B2	C3	D3	E3	F3	G3
↑	↑	↑	↑	↑	↑	↑	↑
a	e	i	o	u	a	e	i
A3	B3	C4	D4	E4	F4	G4	A4
↑	↑	↑	↑	↑	↑	↑	↑
o	u	a	e	i	o	u	a

Figure 3. Vowel-to-pitch mapping for Guido D'Arezzo's text-setting method.

Note that the table of correspondences shown in Figure 3 provides the composer with three choices for each vowel (four for the vowel 'a'). As no rhythmic context is specified by the formalism, the interactive AIMS application allows the user to provide a rhythmic context through real-time interaction with the application.

5.12 Messiaen's Communicable Language (1969).

Inspired by Saint Thomas Aquinas' *Summa Theologica*, the French composer Olivier Messiaen (1908-1992) created a so-called "communicable language"—a mapping of letters of the alphabet to pitches and durations, with additional motives conveying special metaphysical meanings—for his organ work *Méditations sur le mystère de la Sainte-Trinité* (Messiaen 1973). The mapping is shown in Figure 4.



Figure 4. Letter-to-pitch (and duration) mapping for Messiaen's communicable language.

Unlike Guido's formalism above, a rhythmic context is provided by Messiaen. The AIMS application allows for a strict implementation of Messiaen's formalism, or alternate musical outcomes via real-time interaction. It should also be mentioned that Messiaen's communicable language is often cited as an example of musical cryptography. (e.g., Sams 2006)

5.2 Sonification of Data

Playing with Pi (And Other Constants) was originally created to explore application designs that require data be read from an ASCII text file. A file containing the digits of the decimal expansion of π (3.14...) is loaded into memory using MaxMSP's `filein` or `table` object. The digits are then mapped (at a rate specified by the user-selected tempo) to

MIDI pitch and key velocity values. To provide an interesting musical context for exploration, a two-voice canon is implemented. The user can specify the levels of pitch and rhythmic imitation for the canon's follower, as well as other parameters. A prototype of the interface is shown in Figure 5.

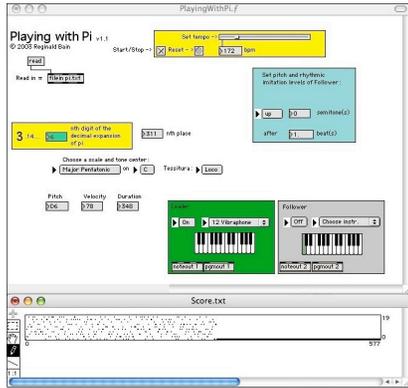


Figure 5. *Playing with Pi*.

Similar data sources, for example, e , Φ , etc., may be explored by supplying the application with appropriately formatted ASCII input.

5.3 Fractals and Chaos

The following examples implement mathematical models associated with the study of fractals and chaos theory.

5.3.1 The Gardner-Voss 1/f Dice Algorithm.

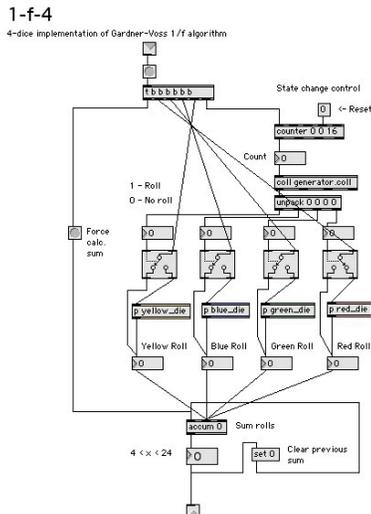


Figure 6. 4-dice subpatch of the $1/f$ algorithm.

Figure 6 (above) shows the MaxMSP subpatch that implements the 4-dice version of the well-known Gardner-Voss algorithm, an algorithm that generates self-similar $1/f$ noise. The application allows the user to create a “ $1/f$

melody” in real time using a variety of pseudorandom dice-roll simulation techniques. Bolognesi (1983) provides a good technical introduction to the subject. It also addresses issues pertaining to the design and analysis of automated composition experiments, a topic integrally related to our own work.

5.3.2 The Logistic Difference Equation and Chaos Game.

Finally, we present two applications which explore mappings of mathematical models from chaos theory. The *Logistic Difference Equation* (Gleick 1987) and *Chaos Game* (Barnsley 1993) were developed to explore application designs that require graphical results be produced in synchrony with the sonification. Figure 7 shows the top-level interface for the *Chaos Game* application.

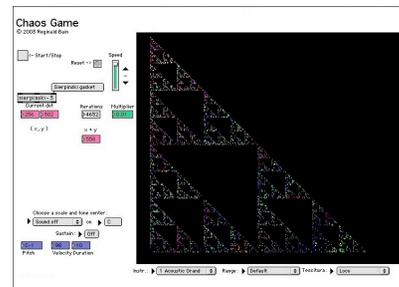


Figure 7. *Chaos Game*.

6 Conclusion

The initial AIMS project distribution described above provides a preliminary library of MIDI-based applications built with Cycling 74's MaxMSP that have allowed us to explore basic issues surrounding the development of creative experiments in musical sonification. Plans for future work include creating applications that implement other historical examples of compositional formalisms, adding experiments with natural functions, and adding audio-based examples that employ various sonification and synthesis techniques.

SLAPI and distributions from the AIMS project are available at:

<http://www.music.sc.edu/fs/bain/software/>

7 Acknowledgments

This work was funded in part by a Research and Productive Scholarship award from the University of South Carolina.

8 References

- Barger, Robin. 1994. Pattern and Reference in Auditory Display. In *Auditory Display*, ed., G. Kramer, 151-65. Reading, MA: Addison Wesley.
- Barnsley, M. 1993. *Fractals Everywhere*, Second Edition. Cambridge, MA: Academic Press Professional.
- Ben-Tal, O. and J. Berger. 2004. Creative Aspects of Sonification. *Leonardo* 37(3), 229-232.
- Bolognesi, T. 1983. Automatic Composition: Experiments with Self-Similar Music. *Computer Music Journal* 7(1), 25-36.
- Bregman, A. S. 1990. *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: MIT Press.
- Burk, P., L. Polansky, D. Repetto, M. Roberts and D. Rockmore. 2005. *Music and Computers: A Theoretical and Historical Approach*. Emeryville, CA: Key College Publishing.
- Childs, E. 2003. Musical Sonification Design. MA thesis, Dartmouth College.
- D'Arezzo, G. 1778. Micrologos. In *Hucbald, Guido, and John On Music: Three Medieval Treatises*, ed., C. Palisca, 47-83. New Haven: Yale University Press.
- De Campo, A., C. Frauenberger, and R. Höldrich. 2004. Designing a Generalized Sonification Environment. In *International Computer Music Conference* poster session. Miami: International Computer Music Association. Available at: <<http://iem.at/projekte/publications/paper/sonenvir/view>>.
- Dodge, C. and T. Jerse. 1997. *Computer Music: Synthesis, Composition, and Performance*. New York: Schirmer Books.
- Dodge, C. 1970. *Earth's Magnetic Field*. Nonesuch Records (H-71250).
- Gleick, J. 1987. *Chaos: Making a New Science*. New York: Penguin Books.
- Kramer, G., ed. 1994. *Auditory Display: Sonification, Audification, and Auditory Interfaces*. SFI Studies in the Sciences of Complexity, Proc. Vol. XVIII. Reading, MA: Addison Wesley.
- Kramer, G., et al. 1997. *Sonification Report: Status of the Field and Research Agenda*. Available at: <<http://www.icad.org/websiteV2.0/References/nsf.html>>.
- Loy, G. 1989. Composing with Computers: A Survey of Some Compositional Formalisms and Music Programming Languages. In *Current Directions in Computer Music Research*, ed. M. Mathews and J. Pierce, 291-318. Cambridge, MA: MIT Press.
- Messiaen, O. 1973. *Méditations sur le Mystère de la Sainte Trinité*. Paris: Éditions Musicales Alphonse Leduc.
- Polansky, Larry. 2002. Manifestation and Sonification: The Science and Art of Sonification, Tufte's Visualization, and the "slippery slope" to Algorithmic Composition (An Informal Response to Ed Childs' Short Paper on Tufte and Sonification). Available at: <<http://eamusic.dartmouth.edu/~larry/sonification.html>>.
- Roads, C. 1996. *The Computer Music Tutorial*. Cambridge, MA: MIT Press.
- Rowe, R. 1993. *Interactive Music Systems*. Cambridge, MA: MIT Press.
- Sams, E. 2006. 'Cryptography, musical', *Grove Music Online*. New York: Oxford University Press. Available at: <<http://www.grovemusic.com>>.
- Winkler, T. 1998. *Composing Interactive Music*. Cambridge, MA: MIT Press.
- Xenakis, I. 1992. *Formalized Music*. Hillside, New York: Pendragon Press.