Collaborating with the Behaving Machine: Simple Adaptive Dynamical Systems for Generative and Interactive Music

Alice Eldridge

Submitted for the degree of D. Phil.

University of Sussex

February, 2008

Declaration

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for a degree.

Signature:

Acknowledgements

This is perhaps the smallest of many projects which was spawned by the energy and enthusiasm of Drew Gartland-Jones. I must thank him for support, encouragement and inspiration. To Phil Husbands I am indebted not only for a powerfully zen approach to supervision, but for being an inspirationally Renaissance man in modern academia. Thanks must go to Jon Bird for useful discussion and priceless support in times of need. Thanks also to Chris Thornton for lending his ear in the absence of Phil, and to Thor Magnusson and Peter Copley for discussion of less than conventional ideas. Beyond Sussex, I am indebted to the Live Algorithms for Music group which has provided a rare community for exchange and debate of matters core to this thesis and in particular to Ollie Bown whose intelligent enthusiasm encouraged me to pursue this project and continues to inspire its future moves. Finally I would like to thank members of blip, and all the people and places that have hosted performances of the work described here.

Preface

Much of the material in this thesis has been previously published and presented in various Journals, conferences, festivals and concerts.

In particular, parts of Chapter 6 have appeared in:

- Eldridge, A. (2002b). Adaptive systems music: Algorithmic process as musical form. In *Proceedings of the 2002 Generative Art Conference*, Milan.

Some of the ideas in Chapter 3 and models from Chapter 5 were presented in:

- Eldridge, A. (2005b). Extra-music(ologic)al algorithms for automated composition. In Rothlauf, F. et. al., editors, *Evomusart workshops 2005, LNCS 3449*, Heidelberg. Springer- Verlag Berlin.

The work presented in Chapter 8 has been previously published in:

- Eldridge, A. (2005c). Fond punctions: Generative processes in live improvised performance. In Edmonds, E., Brown, P., and Burraston, D., editors, *Proceedings of Generative Art Practice*, pages 41-51, Sydney, Australia. Creativity and Cognition, Creativity and Cognition Studios Press. andChapter

- Eldridge, A. (2005a). Cyborg dancing: generative systems for man-machine musical improvisation. In Innocent, T., editor, *Proceedings of third Iteration.*, pages 129-142, Mel- bourne, Australia. Centre for Electronic Media Art (CEMA).

Results from the study presented in Chapter 4 were discussed in a different context in:

- Eldridge, A. (2006). Issues in auditory display. *Artificial Life*, 12(2):259-274.

All of these papers should be considered as my own work. In addition, a summary of the ideas presented here are due for publication in a co-written book chapter:

- Husbands, P., Copley, P., Eldridge, A. and Mandelis, J. (2007). An introduction to evolutionary computing for musicians. In Miranda, E. R. and Biles, A., editors, *Evolutionary Computer Music*. Springer-Verlag.

Many of the works presented here have appeared in public exhibitions, festivals and gigs as follows:

AdSyMII (Chapter 6)

- July 2002: Blip5, Bar Sumo, Brighton, UK
- December 2002: Concert of Generative Art 2002, Politecnico di Milano University, Milan, Italy

Organised Entry (Chapter 6)

- October 2005: The Big Blip 05, Brighton Fringe Basement, Brighton, UK

Self-Karaoke Pond (Chapter 8)

- October 2005: The Big Blip 05, Brighton Fringe Basement, Brighton, UK

Ashby's Grandmother's Footsteps (Chapter 7)

- April 2006: Process Revealed at the Artpool Gallery, Budapest, Hungary (in association with the EvoMusArt workshop at EuroGP 2006.)

Performances of *Fond Punctions*, which uses the **Self-karaoke Machine** described in Chapter 8, include the following events and venues:

- July 2005: *Interactive Mind and Art(efacts)*, The Sussex Arts Club, Brighton, UK
- August 2005: The Great Escapade, Sussex, UK
- September 2005: Live Algorithms for Music meeting, The Great Hall, Goldsmiths college, London, UK
- October 2005: Wrong music, The Volks, Brighton, UK
- November 2005: University of Sussex Lunchtime Concert, Friends Meeting House, Brighton, UK
- November 2005: Third Iteration. Monash University, Melbourne, Australia
- December 2005: Generative Arts Practice. University of Technology, Sydney, Australia
- December 2005: Lan Franchis memorial discotheque, Sydney, Australia
- April 2006: Process Revealed, Artpool, Budapest, Hungary
- May 2006: Sunday Relay, The Albert, Brighton, UK

Compositions made with some of the systems here have been publicly played or distributed as follows:

- *Sines* (2002) a piece composed using the basic homeostat described in Chapter 5 was commissioned for the Lux OPEN 2002, Royal College of Art, London, UK
- *It Didn't Happen at Lan Franchis*, *Picket Fence Study 2: The Ant's ear view* and *Picket Fence Study 4: The Larvae's ear view* are due to appear on the cd accompanying *Evolutionary Computer Music*. (Miranda, E.R. and Biles, A.J. 2007). Springer-Verlag

Finally, the 'Behavioural Objects' project outlined in Chapter 9 was presented at:

- Improvising with Computers: IRCAM, Paris as part of NIME 2006

Collaborating with the Behaving Machine: Simple Adaptive Dynamical Systems for Generative and Interactive Music

Alice Eldridge

Summary

Situated at the intersection of interactive computer music and generative art, this thesis is inspired by research in Artificial Life and Autonomous Robotics and applies some of the principles and methods of these fields in a practical music context. As such the project points toward a paradigm for computer music research and performance which complements current mainstream approaches and develops upon existing creative applications of Artificial Life research.

Many artists have adopted engineering techniques from the field of Artificial Life research as they seem to support a richer interactive experience with computers than is often achieved in digital interactive art. Moreover, the low level aspects of life which the research programme aims to model are often evident in these artistic appropriations in the form of bizarre and abstract but curiously familiar digital forms that somehow, despite their silicon make-up, appear to accord with biological convention.

The initial aesthetic motivation for this project was very personal and stemmed from interests in adaptive systems and improvisation and a desire to unite the two. In simple terms, I wanted to invite these synthetic critters up on stage and play with them. There has been some similar research in the musical domain, but this has focused on a very small selection of specific models and techniques which have been predominantly applied as compositional tools rather than for use in live generative music. This thesis considers the advantages of the Alife approach for contemporary computer musicians and offers specific examples of simple adaptive systems as components for both compositional and performance tools.

These models have been implemented in a range of generative and interactive works which are described here. These include generative sound installations, interactive installations and a performance system for collaborative man-machine improvisation. Public response at exhibitions and concerts suggests that the approach taken here holds much promise.

Submitted for the degree of D. Phil.

University of Sussex

February, 2008

Contents

Track Listings

The attached DVD contains:

- Max/ MSP externals for the main algorithms described in Chapter 5. Also included are help files using mappings similar to those described in this chapter.
- A video of one performance made with the Self-karaoke Machine described in Chapter 8.
- Audio examples described in Chapters 5, 6, 7 and 8 are provided on the following tracks:

1	[5:1]	hom-perturb
2	[5:2]	hom-stabilise
3	[5:3]	hom-sines
4	[5:4]	planting-trees-excerpt
5	[5:5]	hom-sam-remix
6	[5:6]	hom-filter
7	[5:7]	hom-wrongbeats1
8	[5:8]	hom-wrongbeats2
9	[5:9]	nosc-change-weights
10	[5:10]	nosc-minima
11	[5:11]	nosc-change-inputs
12	[5:12]	nosc-perc-simple
13	[5:13]	nosc-perc-delta-tau
14	[5:14]	CA-chaotic
15	[5:15]	CA-complex
16	[5:16]	CA-ordered
17	[5:17]	GLV-inuit-pitch
18	[6:2]	AdSyMII
19	[6:3]	Organised Entry
20	[7:1]	Ashby's Grandmother's Footsteps
21	[8:1]	It Didn't Happen at Lan Franchis
22	[8:2]	Picket Fence Study One: The Blackbeetle's Ear View
23	[8:4]	Picket Fence Study Three: The Ant's Ear View
24	[8:3]	Picket Fence Study Two: The Larvae's Ear View
25	[8:5]	Self-karaoke Pond-harp

Glossary: Abbreviations and resources

1D	- One Dimensional
2D	- Two Dimensional
AI	- Artificial Intelligence
ANN	- Artificial Neural Network
CA	- Cellular Automata
CTRNN	- Continuous Time Recurrent Neural Network
DSP	- Digital Signal Processing
EC	- Evolutionary Computation
GA	- Genetic Algorithm
GM	- General MIDI
MIDI	- Musical Instrument Digital Interface
OSC	- OpenSound Control (see below)
USB	- Universal Serial Bus

Arduino is an open-source physical computing platform based on a simple i/o board, and a development environment for writing Arduino software. The Arduino programming language is an implementation of Wiring, itself built on Processing.
`http://www.arduino.cc/en/`

ChucK is a concurrent, strongly-timed audio programming language for real-time synthesis, composition, and performance, which runs on Mac OS X, Linux, and Windows. Code can be added, removed and modified on the fly, while the program is running making it an ideal language for live coding. It was originated by Perry Cook and Ge Wang of Princeton University.
`http://chuck.cs.princeton.edu/`

Csound is a text based music programming language written in the C programming language. A typical Csound program will include an *orchestra* file describing the nature of the instruments and a *score* file describing the parameters of the material (pitch, duration, amplitude etc). Csound then renders these files to produce an audio file or real-time audio stream.
`http://www.csounds.com/`

Jitter extends the Max/MSP programming environment to support realtime manipulation of video, 3D graphics and other data sets within a unified processing architecture.
`http://www.cycling74.com`

Max/MSP	is a graphical development environment for music and multimedia. The program is highly modular and allows the development of third-party externals as objects which can be fully integrated with the native libraries. A typical Max programme, called a 'patch' is based on multiple graphical objects connected into a data flow. Control rate MIDI messages can be combined with a DSP network. Max was originally developed by Miller Puckette and is now developed and maintained by Cycling'74. `http://www.cycling74.com`
Processing	is an open source programming language and integrated development environment (IDE) built for the electronic arts and visual design communities. It builds on the graphical side of Java, simplifying some features and adding new ones. It is developed by Casey Reas and Ben Fry `http://www.processing.org`
Pure Data (Pd)	is a graphical programming language developed by Miller Puckette in the 1990s for the creation of interactive computer music and multimedia works. Though Puckette is the primary author of the software, Pd is an open source project and has a large developer base working on new extensions to the program. It is released under a license similar to the BSD license. `http://puredata.info/`
OpenSound Control	is a protocol for communication among computers, sound synthesisers and other multi-media devices. It is optimised for networking technology allowing very fast data sharing between machines. It can transport over many protocols but is commonly used with UDP or TCP/IP. It can be compared to MIDI, but does not suffer the same time lags and allows an open-ended url-style symbolic naming scheme. `http://www.cnmat.berkeley.edu/OpenSoundControl/`
SuperCollider	is a real time audio synthesis programming language. The Language combines the object oriented structure of Smalltalk and features from functional programming languages with a C programming language family syntax. Originating as proprietary software, it was released in 2002 by its author James McCartney under the free sofware GPL license. `http://www.audiosynth.com/`
Wiring	is a programming environment and electronics i/o board for exploring the electronic arts, tangible media, teaching and learning computer programming and prototyping with electronics. It is an open project initiated by Hernando Barragàn and builds on Processing. `http://wiring.org.co/`

List of Figures