Controlling Live Generative Electronic Music with Deviate

Sarah Keith
Dept. of Media, Music, Cultural Studies
Macquarie University
NSW 2109, Australia
sarah.keith@mq.edu.au

Abstract
Deviate generates multiple streams of melodic and rhythmic output in real-time, according to user-specified control parameters. This performance system has been implemented using Max 5 [1] within the genre of popular contemporary electronic music, incorporating techno, IDM, and related forms. The aim of this project is not musical style synthesis, but to construct an environment in which a range of creative and musical goals may be achieved. A key aspect is control over generative processes, as well as consistent yet varied output. An approach is described which frees the user from determining note-level output while allowing control to be maintained over larger structural details, focusing specifically on the melodic aspect of this system. Audio examples are located online at http://www.cetenbaath.com/cb/about-deviate/.

Keywords: generative, performance, laptop, popular music

1. Introduction
As a relatively young and technologically-oriented domain, current laptop performance displays diverse approaches to live music creation. Algorithmic, generative, and other computer-aided methods for creating live music have come to the fore in recent years. Earlier precedents for real-time algorithmic generation include Mathews' 1970 GROOVE program [2], Vercoe's 1984 Synthetic Performer [3], and Xenakis' evolving UPIC system [4]. The last decade has also seen popular contemporary electronic music performance center around the computer. The turntables of the vinyl DJ have shifted to hardware samplers and sequencers, and finally to composition and performance software such as Ableton Live [5], now a well-established performance tool.

A number of performers have countered such playback-oriented modes of laptop performance through the use of algorithmic techniques. Collins [6] provides a thorough overview of the problems and possibilities, both technical and musical, afforded to those who would attempt to incorporate live coding practice or algorithmic processes in a live context. The divergent fields of algorithmic composition and popular electronic music thus now share the sphere of laptop performance.

The aim of this project is to employ algorithmic approaches within the form of popular contemporary electronic music. Similar projects involving generative programs for contemporary or popular music forms include LEMu [7], Bloom [8], and Infno [9]. The musical context cited for this project is deliberately open-ended; aiming for synthesis of a specific musical style would limit the scope of interactions possible in performance, and furthermore the system is envisioned as a flexible and re-usable environment, unconstrained by close limitations of genre, and capable of producing diverse output.

2. Data generation
Generation of melodic data (as MIDI note events) takes place according to data maps, which consist of manually constructed distributions concerning **timing**, **velocity**, **pitch** and **tonality**. Maps are selected, rendered according to various user-specified parameters, and altered in real-time according to various control parameters.

Timing maps consist of 32 indexed values, representing demisemiquaver resolution in a 4/4 beat structure. Different maps may be substituted to yield different rhythmic results. Each value assigned represents the ‘weight’ of event occurrence; similar statistical representation of temporal events has been used in analysis of musical rhythms, and is able to reflect the rhythmic hallmarks of various genres [10]. Using standard normal distribution, these values are then altered according to a variable sigma value to create deviation from the initial value. These values are then rendered to a score, which is subject to filtering (cutoff) control by the user to determine event density: this and further control aspects will be discussed in the following section. Each of the 32 events generated in also has an associated **velocity** value, which is determinate and serves to reinforce metric structure.

Pitch selection employs data maps similar to the key profiles obtained by Krumhansl and Kessler [11], denoting a statistical distribution of pitch suitability within a given tonal context. To ensure consistency, varying pitch maps are used to generate pitches according to the tonal requirements of the metric structure [12], whereby events possessing metric emphasis are assigned tonal pitches and less metric events are assigned more chromatic pitches. An additional parameter allows the user to determine the tendency towards overall **tonality**, altering the potential for nondiatonic and secondary scale tones to be generated.

Variation is thus achieved through selecting different data maps and by redistributing events using the Gaussian function, as opposed to changing rules within the system itself. This provides flexibility and extensibility, and aids in developing a cognitive grasp over the system.

3. Data control

Each of the 32 events generated per module has an associated weight, as discussed in the previous section. A filtering **cutoff** value set by the user outputs all events with a higher given weight than the cutoff value, controlling the density of note output. The cutoff function can be set as a constant, or as a time-variant function. Further control parameters affect **root note**, **transposition**, **duration** and **volume**. The user can also select from a **chord** menu, resulting in chorded, rather than single-note, note output.

The process explained so far describes a single melody-generating module that produces one bar of monophonic output. In application, four modules are used to provide four sequential 'bars' of musical output that can be selected and arranged by the user. In turn, four of these four-bar structures are layered to allow the simultaneous output of four melodic voices. **Voice enabling** permits voice rows to be enabled or muted, while **arrangement** alters the playback order of modules. A provision for **data storage** and **recall** has also been implemented, allowing the use of reusable 'presets'.

4. Evaluations

From an empirical standpoint, *Deviate* has produced promising musical output ranging from the bland to the unpredictable, as negotiated by the user. As with any musical instrument, a level of practice is necessary in order to develop performance methods that ensure the generation of consistent material. Further directions for this project include a facility for timbral change, including automatic voice selection, equalization, and audio effects. Automatic learning through supplied data is another area for exploration, as data maps could be based on computational analysis of an existing MIDI score or performance. Using this approach, it is likewise possible for a degree of flexibility to be incorporated within a given musical text. Lastly, augmenting the laptop with a physical control interface is an area in the early stages of exploration.

5. Conclusion

The primary aim of the melodic aspect of *Deviate* as presented in this paper is to create new and musically acceptable output in real-time, within the genre of contemporary electronic music. One advantage of the approach taken here is the foundation of generative processes on data maps, rather than musical rules. This method allows for new input to be easily supplied, and permits the development of an extensible library of source data. Other aims include the incorporation of controls that are comprehensible and effective in a performance context.

This project investigates further possible applications of generative music performance, and highlights the proximity of computer music to popular electronic music, as well as the links between computer music and basic music psychology. These areas provide fruitful ground for further exploration and collaboration.

References

[1] "Max 5" [Web site] 2009, [2009 Mar 27], Available: http://www.cycling74.com/products/max5

[2] M. V. Mathews and F. R. Moore, "GROOVE: A Program to Compose, Store, and Edit Functions of Time," *Comm. of the ACM*, vol. 13, no. 12, pp. 715-721, 1970.

[3] B. Vercoe. "The Synthetic Performer in the Context of Live Musical Performance," in *Proc. of the International Computer Music Conference (ICMC)*, 1984, pp. 199-200

[4] M. Marino, M. H. Serra, J. M. Raczinski, "The UPIC System: Origins and Innovations," *Perspectives of New Music*, vol. 31, no. 1, pp. 258-269, 1993.

[5] "Ableton Live 7," [Web site] 2008, [2008 Nov 13], Available: http://www.ableton.com/live

[6] N. Collins, "Generative Music and Laptop Performance," *Contemporary Music Review*, vol. 22, no. 4, pp. 67-79, 2003.

[7] R. Wooller, N. Coleman, A. Brown. "LEMu" [Web site] 2009, [2009 Mar 27], Available: http://www.clatterbox.net.au/instruments/lemu

[8] B. Eno, P. Chilvers. "Bloom | GenerativeMusic.com" [Web site] 2009, [2009 Mar 27], Available: http://generativemusic.com

[9] N. Collins. "Infno: Generating Synth Pop and Electronic Dance Music On Demand" in *Proc. of The International Computer Music Conference (ICMC)*, 2008 (forthcoming).

[10] D. Huron, *Sweet Anticipation: Music and the Psychology of Expectation*, Cambridge, Massachusetts: MIT Press, 2006, pp. 180-186.

[11] C. L. Krumhansl and E. J. Kessler, "Tracing the Dynamic Changes in Perceived Tonal Organization in a Spatial Representation of Keys," *Psychological Review*, vol. 89, no. 4, pp. 334-368, 1982.

[12] M. R. Jones, "Dynamics of Musical Patterns: How do Melody and Rhythm Fit Together?" in *Psychology and Music: The Understanding of Melody and Rhythm*, T. J. Tighe, W. J. Dowling, Eds. Hillsdale, NJ: Lawrence Erlbaum Associates, 1993, p. 68.