

Methods for the Flexible Parameterisation of Musical Material in Ableton Live

Oliver Bown

Design Lab,
University of Sydney, NSW, 2006, Australia
oliver.bown@sydney.edu.au

Sam Britton

School of Arts,
Brunel University, Middlesex, UB8 3PH, UK
sam@icarus.nu

Abstract

We present a set of tools that were developed for an experimental music project in which an album was created with 1,000 distinct variations and released as a digital limited edition. The tools enable the parameterisation of musical compositions created in the popular music software Ableton Live such that timelines of musical elements can be composed as keyframes and interpolated between.

Parameterisation

The parameterisation of human-designed artefacts turns a single design into a navigable ‘design space’. ‘Parameterisation’ can be defined as the extraction of a fixed set of parameter values from a given entity. This allows for rich mappings from parameters to design outcomes as long as these are deterministic and enumerable. Not all generative systems are naturally parametric – for example, L-systems and generative grammars exist as complex structures – but entities in these systems can be defined as strings and can therefore be enumerated, and thus reduced to parameters. For such systems more suitable enumerations may exist. Much parameterisation consists of a more literal identification of parameters, such as the wingspan of an aeroplane design, or the coefficients of a polynomial equation describing wing curvature. A desirable property of a parameter system is smoothness, such that proximity in parameter-space corresponds to proximity in the design space. More generally the relationship between the geometry of the parameter space and the resulting points in the design space may itself be a design challenge.

Parameterisation and Musical Metacreation

Parameterisation is a common stage in optimisation tasks such as artificial evolutionary optimisation. Here the set of parameter values is understood as a genotype and the specific design resulting from a point in the parameter-space is understood as a phenotype. The smoothness of the genotype-phenotype mapping aids the ability for the algorithm to incrementally improve on designs. More generally, parameterisation can be used to provide a simple graphical interface with which a user can interactively explore a design space.

Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Although parameterisation itself provides only a very weak step towards musically metacreative systems it is a fundamental activity in a vast amount of musical metacreation research.

The argument for the relevance of the current work to musical metacreation is that it provides a small step towards enabling the metacontrol of musical works in a regular creative workflow. Its contribution lies in demonstrating a working system that allows non-technical users to parameterise and parametrically control their compositions. A consequence of this is that each such parametric composition provides a search space, albeit very simple, that an automated system can generatively explore. The current work therefore suggests working methods that may be more amenable to collaborative human-machine control of creative output, possibly involving distributed collaborative creativity between multiple practitioners.

Parameterisation in Design

A number of creative industries have embraced parameterisation. For example, the software Grasshopper¹ is a popular plugin to the Rhino 3D design environment, which facilitates the parametric control of 3D structures. Parametric design of 3D forms has become commonplace, with applications ranging from studies of variation of form within the design of a single building, to evolutionary optimisation tasks. Such techniques have been praised for contributing to increasingly ‘organic’ architectural forms. Rhino and Grasshopper typify the productive integration of programmable and reconfigurable algorithmic methods within pro-user creative environments.

Parameterisation in Music

Parameterisation, and more specifically parametric interpolation, has been used to creative effect in a number of experimental computer music works, particularly in the design of intuitive interfaces for composition or performance. For example, the popular music programming environment MaxMSP² has for some time provided a series of objects that allow all numeric user interface variables to be stored as preset states, and further allows users

¹See <http://www.grasshopper3d.com>.

²<http://www.cycling74.com>

to smoothly interpolate from one preset state to another. Other examples of specific experimental interface designs include Collins' SplineSynth (Collins, 1999) and Bencina's MetaSurface (Bencina, 2005). More broadly, suitable strategies for mapping from control parameters to musical elements is a regular topic of discussion (e.g., see Wanderley, 2002). Yet despite music being one of the most fertile domains for creative technology development, tools similar to Rhino/Grasshopper, that combine professional creative environments with generative flexibility, have yet to emerge. This may be due to differences between the structure of music as compared to other creative domains, or to do with typical working practices of musicians. Nevertheless, there would appear to be no major differences in the potential creative gains that could be made by enabling parameterisation in a musical domain.

Parameterisation is itself not sufficient to constitute musical metacreation. However, it provides a way to integrate the workflow of creative musicians with musical metacreative techniques such as evolutionary computation. Increasingly, good design and successful integration of advanced AI into regular workflows is being called for in such domains, such as by the first author (Bown and Saunders, 2013). It also suggests a number of new musical forms (such as the experimental album discussed in the following section) which indicate novel application domains for creative AI.

A Parametric Album

The current work was the result of a project with a specific experimental music goal: to compose an album of music which could be parameterised and distributed in 1,000 variations³. This was with three distinct aims:

1. to explore the potential of the medium of digital distribution to facilitate alternative relationships between a music consumer and a musical work – in this case, each individual who purchased a copy of the album received their own unique version, in effect a 'digital limited edition';
2. to explore the musical possibilities of treating a composition as an entity with multiple manifestations – this is of course a well established notion in general, but not in the specific case of using technology to mechanise the process of creating variation, especially at the production stage;
3. to also explore the effect this context has on the creative process of composition, from a design point of view, in other words to look beyond the current creative project towards general designs for musical parameterisation. An important goal of the album was to create the conditions by which one could work with standard professional tools rather than musically obscure practices such as writing program code.

The conceptual and creative aspects of this work are discussed in Bown and Britton (forthcoming), which gives further detail about the musical parameters that have been used (not discussed here due to space limitations).

³Fake Fish Distribution, by Icarus, available at <http://www.icarus.nu/FFD>.

Musical Parameterisation in Ableton Live

This latter aim was well facilitated by the recent integration of two popular commercial software packages: the digital audio workstation (DAW) Ableton Live⁴, and the algorithmic media environment MaxMSP. As a result of this integration, almost any parameter accessible through the GUI of Ableton Live can be remote-controlled from a MaxMSP patch, including device parameters (devices include audio effects and virtual instruments), mixer controls and clips (clips are chunks of musical content, either MIDI or audio, that can be triggered for playback in live performance or sequenced in a timeline to create a composition).

Keyframe-Based Breakpoint Function Interpolation

An Ableton Live project is itself a parametric musical system with a graphical user interface, and it is the mechanised manipulation of these parameters that is our focus. A standard and effective approach to parametric control is based on the idea of interpolation between keyframes, or more correctly, "key states", where each state is a specific set of values for the given parameter set. However, since we typically also expect to be able to control musical parameters over time throughout the duration of a piece of music, as well as from one variation to the next, we apply interpolation not directly to parameter values but to breakpoint functions (in a DAW these are typically known as automation lanes or continuous controller tracks) that describe the progression of parameters over time.

This requires two steps from the user: (1) the assignment of actual parameters in Ableton Live to breakpoint functions. This can be done directly (e.g., a breakpoint function is used to control a track volume slider) or via some custom mapping (discussed below); (2) the creation of a number of breakpoint keyframes that can be interpolated between.

Our toolkit provides objects written in MaxMSP that fully support the second stage, and facilitate the first stage. A graphical breakpoint editor⁵ is provided that allows the user to create, edit and recall breakpoint keyframes. An interpolation strategy was developed to handle interpolation between functions with differing numbers of points in an appropriate way. For any given parameter, any number of breakpoint functions could be created and an interpolation path through them could be designed (see Figure 1).

Parameter Reduction, Coupling and Custom Mapping

The mapping from breakpoint parameters can be direct, e.g., a breakpoint function controls a single dial or slider in a Live device, or indirect. Live itself actually offers powerful tools for coupling parameters: a 'macro' dial can be mapped to multiple device parameters with different ranges, allowing complex variation in effects chains to be controlled by single parameters. In addition, since our tools are provided as

⁴<http://www.ableton.com>

⁵built around the *ej.function.js* object in Emanuel Jourdan's *ejies* library, available at <http://www.e-j.com>.

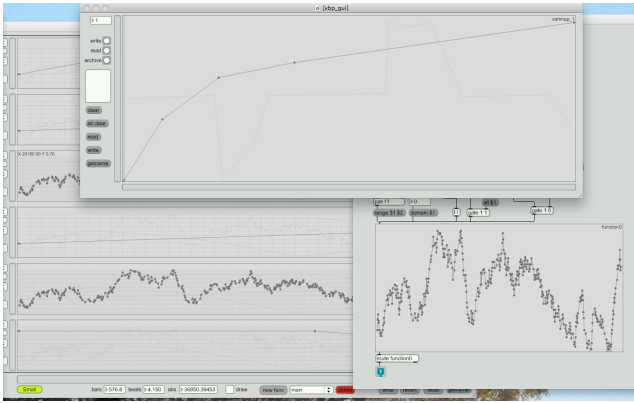


Figure 1: A bank of breakpoint editors controlling parameters (left), another breakpoint editor that is used to manage interpolation between the breakpoint keyframes (top middle), and a resulting interpolation (right).

elements in Max for Live patches, it is easy to add mappings in Max from a single breakpoint function to multiple elements. Furthermore, keyframe interpolations for different breakpoint function parameters can be ultimately reduced to a single parametric control.

As well as continuous parameters discrete Ableton Live clips can also be activated. In order to represent sequences of clips in a way that can be smoothly interpolated these elements can also be controlled by continuous breakpoint functions. Since Ableton allows a quantisation level to be set individually for each clip this means that smoothly varying clip onset times can be constrained to quantised positions. Furthermore, Ableton's clips are able to combine looped and unlooped elements, such as an audio loop combined with a volume ramp, and are also capable of applying simple rules of progression from the end of one clip to the next. This allows for quite complex musical structures to be set up in a way that their arrangement can be parameterised, but without losing the ability to manually control them in detail.

Trajectories as Parameters

A good approach to parameterising the long term structure of a piece of music is to imagine the musical content in the form of a 2D map. Different points on the map correspond to a musical state (for example, the bassline and the drums are playing, the reverb is set to max, etc.), and a smooth trajectory through this space corresponds to one version of the musical work. Using the Max *nodes* object, a partition of a 2D space in regions can be made and mapped directly to parameters. Two breakpoint functions can then be used to control the x and y coordinates of the current position in the musical state space. Interpolating between different instances of these breakpoint functions corresponds to an interpolation between different trajectories through the musical work. This can be highly effective as a design approach, making it easy for a composer to easily conceptualise and create a meaningful range of ways that a musical piece could progress, and being able to carefully ensure that all transi-

tions are well designed.

Rendering Instances

Parameterised musical pieces can easily be rendered out into actual instances, and this can be done in batches by iterating through parameter sets. With Max for Live an unfortunate limitation is that this needs to be done in realtime, since offline rendering cannot be scripted (whereas playback controls can be, and Max can itself be used to record the output to a file). Being a versatile workstation, Ableton's own effects, or third party plug-ins, can be used to do pre-rendering mastering. Further batch processing of the resulting files can easily be done using a range of commandline or end-user tools.

In the case of the album in 1,000 variations, the entire set of parameter controls was reduced to a smoothly interpolating variation control, and 1,000 versions of each track were rendered by incrementally working through this version number. Different applications may require more multidimensional control of the parameters.

Conclusion

These tools were used to make the album Fake Fish Distribution, released in February 2012 on the Not Applicable label, as a digital limited edition of 1,000 variations. They achieved the aim of integrating parametric control into a standard musical workflow in musically effective ways. We believe that the tools will be of general applicability to composers and will enable a new tighter integration between artist workflows and creative uses of AI techniques.

Acknowledgements

This software was partly developed during a residency at STEIM in Amsterdam, The Netherlands. We thank Ableton for their support during the development of the project.

References

- Bencina, R. 2005. The metasurface – applying natural neighbour interpolation to two-to-many mapping. In *Proceedings of NIME'05*.
- Bown, O., and Britton, S. forthcoming. An album in 1,000 variations: Notes on the composition and distribution of a parametric musical work. Leonardo. Accepted February 2013. Preprint available from <http://www.olliebown.com>.
- Bown, O., and Saunders, R. 2013. Multi-feature visualisations of phenotypic behaviour for creative interactive evolution. In *Proceedings of the 2013 GECCO conference*.
- Collins, N. 1999. SplineSynth: An interface to low level digital audio. In *Proceedings of the 1999 Diderot Forum on Mathematics and Music*.
- Wanderley, M. 2002. Mapping strategies in real-time computer music. *Organised Sound* 7(2).