

2016

A multi-point 2D interface: Audio-rate signals for controlling complex multi-parametric sound synthesis

Stuart James

Edith Cowan University, s.james@ecu.edu.au

Follow this and additional works at: <https://ro.ecu.edu.au/ecuworkspost2013>



Part of the [Music Commons](#)

James, S. (2016). A Multi-Point 2D Interface: Audio-rate Signals for Controlling Complex Multi-Parametric Sound Synthesis. *Proceedings of New Interfaces for Music Expression*.

Available [here](#)

This Conference Proceeding is posted at Research Online.

<https://ro.ecu.edu.au/ecuworkspost2013/3356>

A Multi-Point 2D Interface: Audio-Rate Signals for Controlling Complex Multi-Parametric Sound Synthesis

Stuart James
West Australian Academy of
Performing Arts
Edith Cowan University
s.james@ecu.edu.au

ABSTRACT

This paper documents a method of controlling complex sound synthesis processes such as granular synthesis, additive synthesis, timbre morphology, swarm-based spatialisation, spectral spatialisation, and timbre spatialisation via a multi-parametric 2D interface. This paper evaluates the use of audio-rate control signals for sound synthesis, and discussing approaches to de-interleaving, synchronization, and mapping. The paper also outlines a number of ways of extending the expressivity of such a control interface by coupling this with another 2D multi-parametric nodes interface and audio-rate 2D table lookup. The paper proceeds to review methods of navigating multi-parameter sets via interpolation and transformation. Some case studies are finally discussed in the paper. The author has used this method to control complex sound synthesis processes that require control data for more than a thousand parameters.

Keywords

audio-rate control, 2D multi-parametric interface, geometry, interleave, de-interleave, mapping, nodes, table lookup, interpolation, transformation

ACM Classification

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing, H.5.2 [Information Interfaces and Presentation] User Interfaces, G.4 [Mathematics of Computing] Mathematical Software

1. INTRODUCTION

The use of audio signals for control is common for sound synthesis. Modular and semi-modular synthesisers use electronically generated audio signals for controlling a sound synthesis process. Modular and semi-modular synthesisers use 3.5-mm Jack or TT patch leads to send control voltages from the output of one signal generator module to the input of another. This is commonly used for modulation synthesis techniques such as *Amplitude Modulation* (AM)¹, *Ring Modulation* (RM)², and *Frequency Modulation* (FM)³ synthesis, but in reality can be used to control any operator or process.

Cort Lippe and Zack Settel have also documented an approach to controlling FFT-based processes using audio signals [2, 3]. In 1999 they named this process *low dimensional audio rate control* [1]. FFT-based processes are computed in frames of samples (often 256, 512, 1024, or some other power-of-2), in what is sometimes referred to as the frequency-domain. In this way a time-domain audio control signal is also

computed in an audio buffer of the same size in samples. For example, for a 1024 size FFT, an audio control signal is buffered in groups of 1024 samples, and assigned to a multi-parameter process, in this case they are assigned to control the state of 512^4 parameters of the FFT. In the case of FFT-based processes, the use of audio control signals ensures that control data is both synchronized, and maintains precise resolution with the synthesis process.

These two different approaches of using audio signals to control sound synthesis processes present two generalised methods of mapping audio signals (as controller data) to a sound synthesis process. The first is an explicit one-to-one mapping of audio signal to operator or process, and the second is a one-to-many mapping of an audio signal to a vector-based process of many parameters. In order to explore the potential of this control method, the author extended this one-to-many mapping approach for the real-time and simultaneous control of up to 32,768 parameters for a 32-channel implementation of timbre spatialisation in the frequency domain [4].

2. CONTROLLING MULTI-PARAMETER PROCESSES USING AN AUDIO SIGNAL

Whilst a digital audio signal may be considered a single 1D list of values, there are many ways in which such lists may be constructed or deconstructed. For example, in mathematics an interleave sequence is obtained by shuffling two sequences. If S is a set, and (x_i) and (y_i) , $i = 0, 1, 2, \dots$, be two sequences in S . The interleave sequence is defined as the sequence $x_0, y_0, x_1, y_1, \dots$

In reverse, digital signals can also be de-multiplexed or de-interleaved into several different signal streams, and in this way may be used to control several processes simultaneously. The code presented in Figure 1 shows a simple *gen~* process in *MaxMSP* that is responsible for de-interleaving a single audio input signal into four separate audio output streams.

```
g1, g2, g3, g4 = gate(plusequals(int(1), 0, min=1, max=5),
int(1), choices=4);
out1 = latch(in1, g1);
out2 = latch(in1, g2);
out3 = latch(in1, g3);
out4 = latch(in1, g4);
```

Figure 1. The *gen~* code in *MaxMSP* responsible for de-interleaving a single audio signal into four separate signals

¹ Dependent on the discovery of resistance by George Ohm, and realized and further explored with the invention of the first audio amplifier in 1906

² Discovered by Harold Bode in 1947

³ Discovered by John Chowning in 1967-8

⁴ This is half the fast Fourier transform (FFT) frame size due to the complex conjugate symmetry of the FFT

Figure 2 shows this code embedded inside a *gen~* object and controlling four different oscillators of an additive synthesis system. The audio input signal could be generated synthetically or sampled as a time-domain signal, or it could be interleaved from several different input sources. Note that the modulation potential of such a system relies on the morphology and transformation of this audio signal. In Figure 2, as pictured, we simply have a static system. This paper will discuss such potential for changing the state of multiple parameters in Section 3.

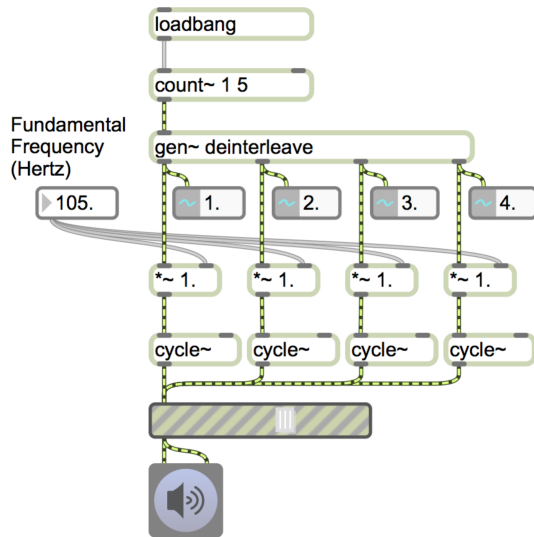


Figure 2. Controlling simple additive synthesis, that is, controlling the pitch of four different oscillators using one audio input signal

If this example were to be extended to control one thousand oscillators or more, it would not be practical to duplicate thousands of instances in a graphical sound synthesis environment, but rather it would be better to iterate the process. In *MaxMSP* this could be achieved either using the *poly~* object, or implementing the process in procedural code such as Java using the *mxj~* object.

2.1 Synchronisation

The advantage of working with digital audio signals, those being digitally controlled components in analog systems and digital signals in computing, is that the timing of these signals can be synchronized. This means that interfacing two different processes, for example a control layer and a synthesis layer, is possible if they are both running off the same master clock; like any digital timing protocol such as *MIDI time code* (MTC), *SMPTE*, *DMX*⁵, *Alesis Digital Audio Tape* (ADAT) format, *Multichannel Audio Digital Interface* (MADI), *DANTE* and *audio-over-IP* (AoIP) technology, these will therefore be synchronized in their recommended configuration, as they are running off the same scheduler.

Several real-time signal processing environments distinguish between audio-rate and control-rate signals [6, 7, 8]. However different applications approach the scheduling of control-rate and audio-rate signals differently. *MaxMSP* has two runtime schedulers: the Max “control” scheduler (based in milliseconds), and the MSP “audio” scheduler timed at the audio sampling rate [5]. *CSound*, *Supercollider*, and Native Instruments *Reaktor* on the other hand set the control rate value

as a numerical factor of the sampling rate (based on the notion of clocking all signals to a master clock or super-clock). Lippe and Settel suggest that significant and continuous modification of a spectrum, as in the case of a sweeping band-pass filter, is not possible using control data of low timing resolution or precision. Depending on the platform, they may not be precisely timed, and their resolution may not keep up with the task of providing 1024 parameter changes at the FFT frame rate of 43 times a second (using FFT buffers of size 1024 at the audio sampling rate of 44,100 samples per second).

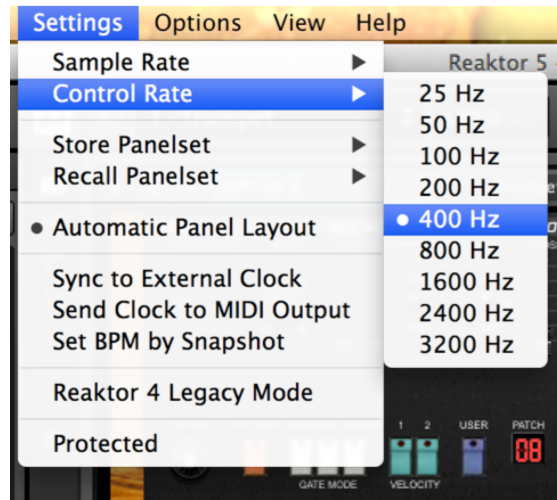


Figure 3. Control signal sub-sample rates available in Native Instruments Reaktor.

Table 1. Specifying control rate signals and audio rate signals in *CSound* and *Supercollider*

<i>CSound</i> Score Header	<i>Supercollider</i>
<pre>sr=44100 kr=4410 ksmps=10 Nchnls=2</pre>	<pre>FSinOsc.ar(800, 0.0, 0.2); // create a sine oscillator at 800 Hz, phase 0.0, amplitude 0.2 FSinOsc.kr(800, 0.0, 0.2); // create a sine oscillator at 800 Hz, phase 0.0, amplitude 0.2</pre>

In the case of environments like *MaxMSP*, control rate messages and video cannot always guarantee exact timing synchronicity. OSC suffers from the same kind of clocking issues, as the timing of events is not tightly synchronised, and using certain network protocols such as UDP (user datagram protocol) may not be guaranteed to arrive at all. Audio signals as control signals have the benefit of being time synchronized, and can transmit high-precision numbers of 16-, 24- or 32-bit floating point resolution at rates generally faster than control rates offered in programmable synthesis environments. However we must also ask ourselves what the parameter is we are controlling, since if we were controlling a video processing algorithm, it would be more efficient to update the value once per processed video frame [10].

When it comes to applying these principles to an FFT process, synchronisation is straightforward as the FFT has its own internal scheduler (the frame index). Other methods don't have this, and therefore some kind of clock is necessary to determine this. Ideally any of these processes should be controlled by an external master clock which both the synthesis process and control layer can access.

⁵ DMX512-A

2.2 Mapping

The issue of mapping manifests in the choice of strategies for transference of physical input gestures to sonic results, which includes both the decision of what parameter associations to make as well as the behaviour of this transference itself [11]. There are many styles of digital music performance that do not have the communicative aspects of human gestures as their primary concern, yet real-time control and organisation of sound materials is often still of paramount importance [12].

Andy Hunt and Marcelo Wanderley (2002) define several mapping strategies that have developed over the years: one-to-one, one-to-many and many-to-one, and several of these approaches can be used in combination for a variety of many-to-many mappings. Doug Van Nort and Wanderley state that a mapping can be *explicit* or *implicit* [12]. The former refers to a situation in which the mapping is known and can be expressed analytically, whereas the latter is based on internal adaptation of a system, and can be seen as a ‘black box’ model [14]. This latter approach is seen as promising in that it can allow a user to adapt a performance system to their unique gestures. However, the explicit approach is beneficial in that having knowledge about the way that the mapping occurs allows one to tune, alter and expand it over time and for different musical contexts.

Audio signals consist of a finite amount of data within timed intervals, dependent on the audio sampling rate. The number of parameters that can be assigned within a time interval may vary depending on the number of samples stored within each audio vector. The relationship here with the audio sampling rate is such that if we unpack 10 interleaved values, we effectively end up with control data at a sub-sampling rate. In this case if the audio sampling rate is 44,100Hz, the control rate is 4,410Hz, and we result in a one-to-ten explicit mapping as a result of the de-interleave process.

In this way, control signals can be derived either according to the number of control parameters required, or by the time necessary to compute the next value for a relevant parameter of the sound synthesis process the signal is mapped to.

$$t = \frac{F}{sr}$$

where t is the time in seconds, F is the vector or frame size in audio samples, and sr is the audio sampling rate. This could be extended to determining the frequency of the audio vectors.

$$f = \frac{sr}{F}$$

where f is the frequency of the audio vectors.

Table 2. A comparison between the window size, frequency, and time taken at 44,100Hz sampling rate. The number of samples are equivalent to the number of parameters that can be controlled.

Window Size (in samples)	Frequency (in Hertz)	Time (in milliseconds)
1 sample	44,100Hz	0.02 ms
10 samples	4,410Hz	0.23 ms
100 samples	441Hz	2.27 ms
1024 samples	43.07Hz	23.22 ms
88,200 samples	2 Hz	2000 ms

In most situations one either knows the computation time for the synthesis process, such as grain time for granular synthesis, or if the synthesis process is calculated in frames, such as the number of grains processes simultaneously for granular synthesis or the size of the FFT frame. If one knows that they

want to control 1000 grains simultaneously, then one is able to compute this every 22.68ms, or at a frequency of 44.1Hz.

Multiple audio streams may be used to determine multiple values per parameter, for example the state of each spectral bin requires both amplitude and phase. Spatial coordinates require at least 2 or 3 different values determining azimuth, distance, and elevation. It would make sense to manage these using 2 and 3 audio signals respectively.

3. A 2D/3D INTERFACE

In terms of the application of such an approach requires a visualization that shows the state of 2 or 3 different audio signals, and color coding each sample according to the total number of interleaved samples (that is, the color range is normalized to the total number of interleaved samples). Figure 4 shows the color coding as it might be applied to the parameters of an FFT process (i.e. bin index).

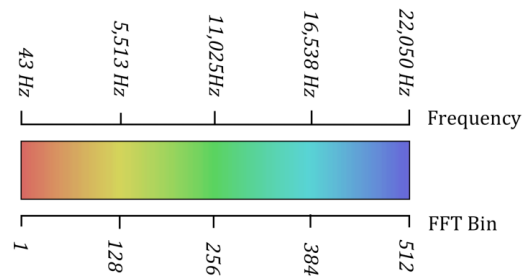


Figure 4. Color coding for bin indices of an 1024 size FFT

These audio signals are plotted parametrically using this color coding either in 2D or in a virtual 3D environment (available through the use of the OpenGL API).

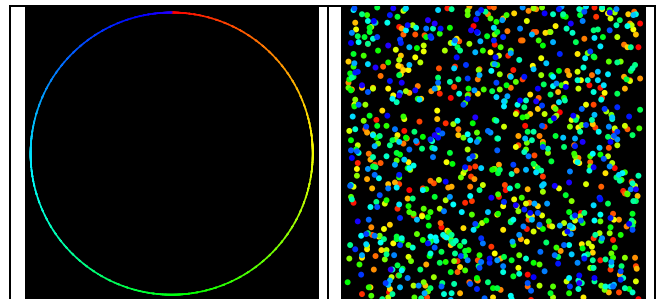


Figure 5a. Two continuous sinusoidal audio signals plotted parametrically where the subsequent signals (once de-interleaved) are also continuous.

Figure 5b. Two discontinuous random audio signals plotted parametrically where the subsequent signals (once de-interleaved) are also discontinuous.

There are many ways in which such 2D or 3D signals may be generated. Some of these computational and algorithmic techniques include algebraic, trigonometric, iterative, procedural and vector-based processes. Other processes include particle systems, which involve the moving of particles through space using vector fields, vector math and quaternions. These are often classed within a field called ‘kinematics’ [15]. Such vector-based systems also give rise to behavioural systems such as flocking and swarming algorithms like the Boids algorithm. Other vector-based systems can apply geometric movements to each individual parameter within a vector. Figure 6, inspired by

Trevor Wishart's writing on the counterpoint of spatial motion, outlines one block procedure by the author that focuses on this kind of vector-based sound movement, and subsequently used for controlling complex sound synthesis [19, 4].

In order to create smooth continuous parameter changes, both audio signals used for control have to be periodic, and have the same frequency as the interleave cycle. Discontinuous parameter changes may emerge from most other scenarios, particularly with discontinuous audio signals that are also non-periodic or repeating, or if audio signals have a different frequency than the interleave cycle.

```

case 21:
  //Circular Harmonics
  for(i = 0; i < o1.length;i++)
  {
    var2 = (int)in1[i];
    var4 =
(float)((float)var2/(float)1023)*(Math.cos((p1*((int)var2 %
(int)p2)/1023)+(myInitial[var2]]));
    var5 = (float)((float)var2/(float)1023)*(Math.sin((p1*((int)var2
% (int)p2)/1023)+(myInitial[var2]]));
    o1[i] = var4;
    o2[i] = var5;
    o3[i] = (float)(Math.sin((p1*((int)var2 % (int)p2)/1023)));
    myInitial[i] = (p1*((int)var2 % (int)p2)/1023)+myInitial[var2];
  }
  break;

```

Figure 6. The Java block procedure responsible for generating a counterpoint of circular trajectories of different speed, based on the natural harmonic series.

3.1 2D/3D Nodes and Signal Interpolation

For continuous and periodic audio signals, a control signal will result in static and non-evolving changes to parameters, much the way static oscillators sound. On the other hand, just as a sound designer may modulate sounds in order to create time-varying sounds, control signals can also be significantly influenced by the time-varying nature of a waveform in the same way.

Interpolation between several different generative or algorithmic systems proved to be a powerful morphological tool. In order to achieve this, from a control perspective, the 2D visualization introduced above was also coupled with a 2D nodes interface (as shown in Figure 7), which was largely responsible for providing a simple 2D space through which a user can intentionally contort and significantly change the topology and geometry of 2D and 3D forms displayed. Since this method shows no precedence for continuous versus discontinuous, linear versus non-linear, or algorithmic versus procedural processes, such an interface proves to be both intuitive and flexible in allowing the user to explore the in-betweenness of different generative systems.

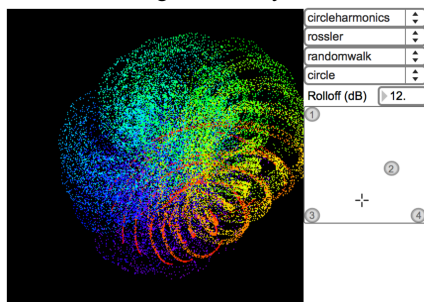


Figure 7. The nodes interface (on the right) allowing the user to interpolate between various generative systems, in this case a counterpoint of circular motions as shown in Figure 6, a circle, random walk and the Rössler chaotic attractor.

This research adapted the DBAP panning technique [16] to take multiple trajectory sources and pan across these input sources to generate a single output control signal. Each input source can be arbitrarily positioned within this virtual navigable space. The advantage of DBAP is its ability to both adapt appropriate loudness curves where different sound sources might normally intersect, ensure that loudness roll-off curves are extended for where sources do not intersect, and ensure that unity gain is maintained with respect to the resulting output signal. This is a slight point of departure from traditional DBAP panning as the author re-appropriated the DBAP algorithm from a one-to-many mapping often synonymous with spatial panning algorithms to a many-to-one mapping instead [4, 17].

3.2 Signal Transformation

Audio control signals can be transformed using time domain or vector-based methods. There are a number of time domain transformations that have been explored as part of this research project. These include affine transformations such as scaling, translating or the rotation of 2D and 3D plots of audio signals [18]. This research has also involved experimentation with a range of other time domain transformations including smoothing functions, foldover and wrapping, bit-rate reduction, phase distortion, feedback, crosstalk, and aliasing. Such methods can introduce non-linearities in the way in which the trajectory evolves over time. We can use all of these techniques as a means of shaping the 2D and 3D arrangement of points. *Waveshaping* distortion can also be used to change the harmonic content of a trajectory using Chebyshev functions. Transformations can be applied additively in series or parallel.

Vector-based transformation methods can also be useful. Inspired by systems like kinematics, particle systems and swarm systems, or those kinds of distributions determined by vector mathematics, this kind of transformation is concerned with the continuous nature of each independent parameter contained within a vector.⁶ Vector-based transformations may also be applied to trajectories such that each point of an existing geometry is subjected to an independent shift in geometric location.

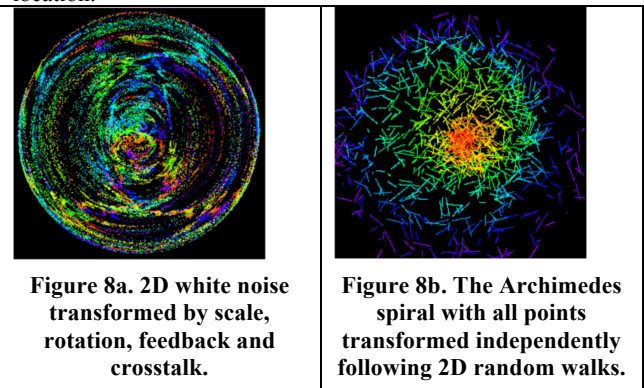


Figure 8a. 2D white noise transformed by scale, rotation, feedback and crosstalk.

Figure 8b. The Archimedes spiral with all points transformed independently following 2D random walks.

3.3 Extending the Interface with table lookup

This further stage allows a 2D or 3D set of audio signals (i.e. 2 channels or 3 channels of audio) to access a 2D or 3D lookup table in order to account for additional control information. The lookup process can be useful in both generating more control data, but may alternatively be useful in introducing further non-

⁶ Although the other methods use a time domain signal for determining evolution of the system, this method is in fact a frequency domain or vector-based method that determines the evolution of each coordinate in the vector independently.

linearities in an existing signal stream. The lookup stage has also shifted this control process enough for it to be considered synonymous with Wave Terrain Synthesis. In this way, by using 2D or 3D table lookup, we are effectively using Wave Terrain Synthesis as a framework to control other complex multi-parameter systems [4, 18].

3.4 Synchronised versus Asynchronised Control of Parameters

Some time-domain audio signals, if they are not generating a repeating periodic signal of a wavelength that matches the vector (in samples) will result in asynchronous mappings of control data to parameters for sound synthesis. There are many different instances that give rise to asynchrony, and such scenarios may be classed in two categories: low- and high-frequency asynchronous mappings, due to the audio signals evolving misalignment with the parameters of a sound synthesis process.

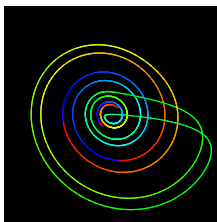


Figure 9. Two dimensions plotted of a 3D chaotic attractor where the 2D paths (pictured) once de-interleaved generate discontinuity. In this case this results in low-frequency asynchrony.

Some transformational processes also force a signal out of alignment, such as a one sample delay, which would cause the data to rotate in its parameter assignment. In the context of an FFT for example, this would involve rotating the frequency bins that control data is mapped to.

Modulating the clocks would also be another way to force a synchronous system to fall out of synchronisation with another. For example one clock might be offset by another, or may slow in relation to the main master clock. Here the clock driving the control data will differ from the clock controlling the sound synthesis process.

4. CONTROLLING MULTI-PARAMETRIC SOUND SYNTHESIS

In order to apply this control data and interface to real world scenarios, some important questions must be asked about how intuitive a particular method may be, how manageable it is for real-time applications, and how flexible and open to expression it is. Insook Choi, Robin Bargar and Camille Goudeseune [20] describe the process through which a performer learns to associate acoustic properties with the actions that produce them as feedback, allowing the performer to measure the consequences of a series of actions. They suggest that *intuition*, within this context of human-machine interaction, describes the experience of participating in feedback systems where the performer can draw compatible conclusions from a small number of trials, quickly learning to differentiate a large number of states in a complex system.

A clear relationship between the actions performed in software and the resulting change of auditory state is vital, and yet when approaching complex sound synthesis involving thousands of control parameters, there are both cognitive and logistical problems associated with this. The user cannot be

intentionally responsible for every parameter, but in this case is responsible for the global distribution of parameters. An analogy in audio engineering is the notion of a group fader, where the performer or engineer is not responsible for riding the level of multiple individual faders on the mixer, but rather a single fader that controls the proportional level of multiple audio channels. In this way, parameter management reduces the necessary burden to one parameter change, making the system easier to manage and simple enough cognitively for the performer or engineer to focus on the sounds themselves, and the musicality of the spatial gestures applied.

4.1 Additive Synthesis / Timbre Morphology

Additive synthesis is one such sound synthesis method that has been bound by problems of a simple interface. Approaches to additive synthesis involving 1024 oscillators may use the 2D audio signal to control relative amplitudes and phase of each sinusoidal component (this may be determined spatially over the 2D visualization the Polar or Cartesian coordinates of each point). At a sampling rate of 44,100Hz, parameters are updated at approximately 43Hz. Since this frequency impinges on the audible frequency spectrum, it is possible to also produce sidebands for each component sine wave too, allowing for increasingly complex timbres to be produced.

4.2 Granular Synthesis

Controlling granular synthesis via such an interface may take grain time or grain size into consideration. In order to control 1000 simultaneous grains, parameters would be updated at 44.1Hz. Depending on the implementation of the synthesis model, parameter assignments are multifarious. For example 2D data could determine the grain pan and grain length of individual grains.

4.3 Swarm-based Spatialisation

Whilst many implementations of swarm-based spatialisation implement control algorithms such as the flocking algorithm and the Boids algorithm, particle systems, the *spatial sound synthesis technique* could also use many of the other multi-point systems described here. In this case the 2D/3D data would be assigned to the spatial position of individual grains. In this case the space-filling properties of the 2D/3D audio signal will also correlate with the level of immersion of the resulting sound spatialisation.

4.4 Spectral Spatialisation

In the case of spectral spatialisation, each frequency bin is assigned an independent spatial trajectory. The 2D/3D interface here may be used to control the spatial coordinates of 1024 simultaneous frequency bands, updated spatially at the sample. The author has published examples of this previously [4].

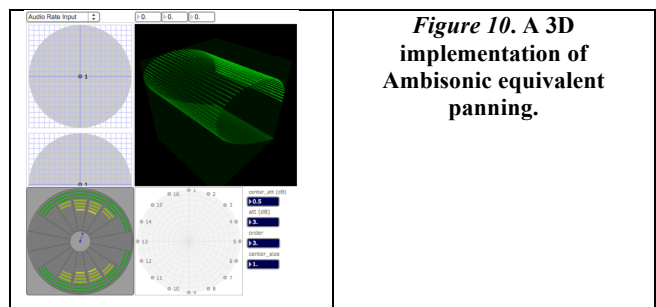
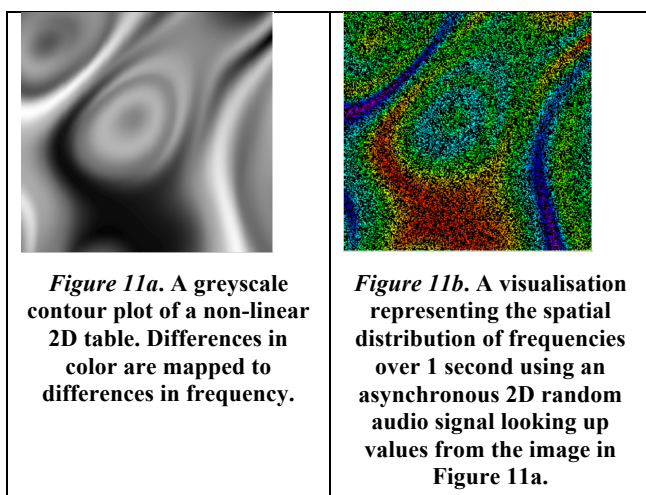


Figure 10. A 3D implementation of Ambisonic equivalent panning.

4.5 Timbre Spatialisation

In this case the vector of 1024 values is mapped to a spatialisation function responsible for decoding the various frequency bands into a series of SPF functions used for convolving a live input source to a series of loudspeakers. The difference between the spectral spatialisation scenario and this implementation of timbre spatialisation, is that this approach also involves a table lookup stage which determines how the frequencies are distributed across space, and the 2D/3D audio signal determines the spatial coordinates to render across the spatial soundfield.



5. CONCLUSIONS

Further research into the use of audio signals to control complex sound synthesis will focus on the possibilities regarding the gestural control of such a system, as well as the diversities of geometries and parametrisations possible. The performance evaluation of such techniques for controlling sound synthesis are of significant importance, particularly in relation to how intuitive, manageable, and flexible such a control system is. Is this method expressive enough, and does it not bombard the user with a myriad of unnecessary parameters?

6. REFERENCES

- [1] C. Lippe, & Z. Settel. Low Dimensional Audio Rate Control of FFT-Based Processing. *Institute of Electrical and Electronics Engineers (IEEE) ASSP Workshop*, Mohonk, New York (1999), 95-98.
- [2] Z. Settel, & C. Lippe. Real-time Timbral Transformation: FFT-based Resynthesis. *Contemporary Music Review*, 10 (1994), 171-179.
- [3] C. Lippe, & Z. Settel. Real-Time Control of the Frequency-Domain with Desktop Computers. *12th Italian Colloquium on Computer Music*, Gorizia (1998), 25-30.
- [4] S. James. Spectromorphology and Spatiomorphology: Wave Terrain Synthesis as a framework for controlling Timbre Spatialisation in the frequency domain. *Ph.D. Exegesis*, Edith Cowan University, Western Australia, 2015.
- [5] M. Puckette. FTS: A Real-time Monitor for Multiprocessor Music Synthesis, *International Computer Music Conference*. San Francisco: Computer Music Association (1991), 420-429.
- [6] R. Boulanger, editor. *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming*. The MIT Press, 2000.
- [7] J. McCartney. Supercollider: A new real time synthesis language. In *Proceedings of International Computer Music Conference. Hong Kong*, 16 (2000).
- [8] T. Place, T. Lossius, A. Jensenius, & N. Peters. Flexible Control of Composite Parameters in Max/MSP. *Proceedings of the 2008 International Computer Music Conference* (2008).
- [9] N. Mamoulis. Spatial Data Management, *Synthesis Lectures on Data Management* 21, Morgan & Claypool Publishers (2012), 22-23.
- [10] R. Jones, & B. Nevile. Creating visual music in Jitter: Approaches and techniques. *Computer Music Journal*, 29(4) (2005), 55-70.
- [11] D. Van Nort, & M. Wanderley. Exploring the Effect of Mapping Trajectories on Musical Performance. *Proceedings of the 2006 Sound and Music Computing Conference (SMC)* (2006), 19-24.
- [12] D. Van Nort, & M. Wanderley. Control Strategies for Navigation of Complex Sonic Spaces. *Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME)* (2007), 379-383.
- [13] A. Hunt, & M. Wanderley. Mapping Performance Parameters to Synthesis Engines. *Organised Sound*, 7(2) (2002), 97-108.
- [14] M. Lee, & D. Wessel. Connectionist Models for Real-Time Control of Synthesis and Compositional Algorithms. *Proceedings of the 1992 International Computer Music Conference*, San Jose State University, USA (1992), 277-280.
- [15] I. Pecino. Spatial and Kinematic Models for Procedural Audio in 3D Virtual Environments. *Proceedings of the 2014 International Computer Music Conference*, Athens (2014). Retrieved from http://www.academia.edu/8419853/SPATIAL_AND_KINEMATIC_MODELS_FOR_PROCEDURAL_AUDIO_IN_3D_VIRTUAL_ENVIRONMENTS
- [16] T. Lossius, P. Baltazar, & T. de la Hogue. DBAP - Distance-Based Amplitude Panning. *Proceedings of the International Computer Music Conference*, Montreal (2009), 17-21.
- [17] S. James. Sound Shapes and Spatial Texture: Frequency-Space Morphology. *Proceedings of the 2014 International Computer Music Conference*, Athens, Greece (2014).
- [18] S. James. Developing a Flexible and Expressive Realtime Polyphonic Wave Terrain Synthesis Instrument based on a Visual and multidimensional methodology. *Masters Thesis*, Edith Cowan University, Western Australia, 2005. Retrieved from <http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1107&context=theses> (accessed May 15 2011).
- [19] T. Wishart. *On Sonic Art*. Harwood Academic, Amsterdam, 1996.
- [20] I. Choi, R. Bargar, & C. Goudeseune. A Manifold Interface for a high dimensional control space. *Proceedings of the 1995 International Computer Music Conference*, Banff Centre for the Arts, Canada (1995).
- [21] S. James. Spectromorphology and Spatiomorphology of Sound Shapes: audio-rate AEP and DBAP panning of spectra. *Proceedings of the 2015 International Computer Music Conference*, Texas (2015).