COMPOSING WITH WAVEHOLES AND MICROSOUNDS.

Pere J Villez
School of Creative Technologies, University of Portsmouth
36-40 Middle Street, Portsmouth, Hampshire, PO5 4BP
pere.villez@port.ac.uk

1 INTRODUCTION

Audio *fusion* and *fission* are essential psycho-acoustic characteristics of *microsound* synthesis (Roads, 2001). The wealth of parameters, however, proposes a challenge, especially in live performance in which there may be a need to control thousands of sonic events every second. Consequently, automated methods for the organization of these events have been developed, notably, *octaviation* in FOF (*Formant Wave Functions*) synthesis (Rodet, 1984) and *pulse masking* in *Pulsar synthesis* (Roads, 2001).

This paper discusses the technological context of microtemporal perception in microsound. It continues to describe the implementation of an alternative microsonic *fusion/fission* method, which employs familiar popular music sequencing and production techniques to the organization of acoustic *particle streams* derived from classic waveforms.
Amplitude masks are generated and synchronized to individual sound *particles*. Finally, it describes how patterns of these masks can be sequenced and morphed into complex audio particle aggregates.

2 WAVES AND MICROSOUND

Sound synthesis can be characterised by two differing representations of sound. The *wave* representation, which, contends that sound waves are continuous phenomena. The second, the *microsound* view (Roads, 2002), which, holds that the continuity of the *wave* representation can be represented using discrete events or acoustical *quanta* (Gabor, 1946; Roads, 2002).

2.1 Wave based synthesis

Classic *wave* synthesis such as *subtractive* or *additive* synthesis (Fig1) have existed as sound design techniques through the history of electronic music spanning many genres and technologies. In subtractive synthesis, a rich harmonic spectrum is filtered to obscure part of the harmonic content of the sound. In *additive* synthesis, however, a rich spectrum is created through the addition of many harmonically simple waves.

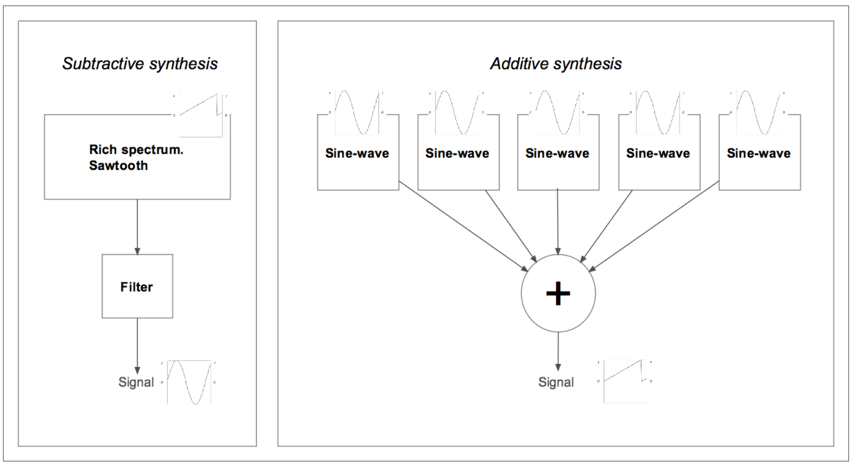

Figure 1. Wave based synthesis

During the 20th century, other wave-based algorithms were developed which added greater complexity to sound synthesis architectures, however, these can often be deconstructed into more generalized forms such as those described above. FOF synthesis, for example, can be viewed as a form of additive synthesis (Rodet, 1984) or subtractive (Giovanni De Poli, Aldo Piccialli, 1991).

In the *wave* representation the wave is perceived as a continuous tone regardless of its fundamental frequency. This sole representation of sound was used until the introduction of the idea of granulated sound by Denis Gabor in his seminal 1946 paper *Theory of Communication* (Gabor 1947).

2.2 Microsound

Microsounds exist below the time scale of the musical note. Microsounds last between a few milliseconds and ~100ms (Roads, 1999).

Microsounds are often referred to as sound *particles* or *grains*. There are a variety of microsonic synthesis techniques and depending on the method of synthesis used they are generally termed, but not limited to, *particles* or *grains*. The term *grain* has tended to be used to refer to events generated using *granular* synthesis (Roads, 1988; Truax, 1988).

This paper refers to *particles* as in *particle* synthesis. In this class are, amongst others, FOF, VoSim (Kaegi Templaars, 1978), PAF (Puckette, 2003) and *Pulsar* synthesis. This type is particular well suited to synthesizing formant rich timbres with a strong independent fundamental frequency, the rate of which is determined by the periodic rate of particle *emission*.

A good visual allegory is to think of a drop of water. In an ideal environment, the sound of a drop of water lasts a few milliseconds, however, if we sequence a series of these droplets so that they follow each other every few milliseconds there becomes a point at which they seem to meld together into a torrent of sound.

2.3 Microtemporal Perception.

Unlike *classic* wave synthesis, microsound techniques exhibit a unique characteristic in that at sub-audio fundamental frequencies the listener is able to perceive the primary acoustic signature of the technique, that is, the *grain*, *particle*, *microsound*.

As particles of sound are generated periodically at up to between 20-40 times per second, they are perceived as discrete events, however, at faster rates the listener begins to hear a *fusion* of *particles* and perceives continuity of tone. The exact threshold of discrimination rate depends on the duration and timbre of the sound, but somewhere in between lies a subjective transitional boundary which the author refers to as the *flutter* region

The boundary between the perception of discrete acoustic events and continuous sound waves is an area, which has been documented variedly since the work of Gabor's paper. Yet, the psycho-acoustic use in musical composition and sound design would seem to need further research.

Many sound classes such as vocal *gurgling* and *rippling* sounds would seem to fall into this category and have been used in compositions such as *Mälarsång* (Clarke, 1987), *Chreode* (Barriere, 1983) and much of Roads's work (Roads, 2001).

There is, however, controversy as too the extent composers have creatively engaged with this region of audio perception and raises the question as to how far we can shake off the legacy and constraints of the traditional elements in music making when applied to electronic music (Miranda, 2002).

A composition, which often crosses the flutter region, is *Mälarsång* where realistically synthesised singing voices, using FOF generators, repeatedly evaporate from solid belcanto vocalisations into *droplets* and eventually into wave like chimes. The listener's perceptual reference is shifted seamlessly from one sound object, the *voice*, to another, the *chime* that seems to have emerged from the depths of the former creating a dramatic sonic narrative.

It is evident then that dynamic manipulation of microtemporal events by *fission* or *fusion* (Roads, 2001, p22) is one of the main compositional interests in microsound, that is, the design of aural illusions (Fig2).

In microsound, the traditional elements of music, pitch, pulse, harmony, dynamics and timbre are extended. These elements can be manipulated and blurred in increasingly creative ways, creating sonic dimensions and aural paradoxes, which challenge the listeners aural and psychological expectations.

The psychoacoustics of this *fusion/fission* is beyond the scope of this paper yet, it can be thought of in terms of the *density of perceived concurrent events*. At higher fundamental frequencies, one event masks the onset of the next (Roads, 2001, p24).

In granular synthesis, it is thought of in terms of *density*. In pitch synchronous particle synthesis such as FOF, PAF or VoSim, it is characterised by rhythm (sparse) or tonality (dense), however, the term *density of* stream would seem an appropriate description for all types of microsonic synthesis. The *stream* being the sequence of microsonic events.

Discrete <---------------> continuous
Particle <----------------------->Wave
<------Fission--------------------------fusion-------->

Figure 2. Discrete to continuous

3 STRATEGIES FOR MANIPULATING PARTICLE DENSITY

Two strategies for the management of this density are directly pertinent to the method presented in this paper.

3.1 Octaviation in FOF synthesis

In the FOF stream it is possible to fade in and out every other particle to create the illusion that the fundamental frequency changing in octaves.

This is achieved by pre-calculating the fundamental frequency required beforehand. The fundamental is then built from consecutive layers of sparser streams of FOF particles and summed into one stream. Density is dependant on how many layers are present. The more layers the higher the fundamental frequency. The duration of the particles are independent of the fundamental frequency which means that at certain fundamental frequencies particles will overlap each other and require further pre-calculation in order to avoid amplitude distortion resulting from summing many events.

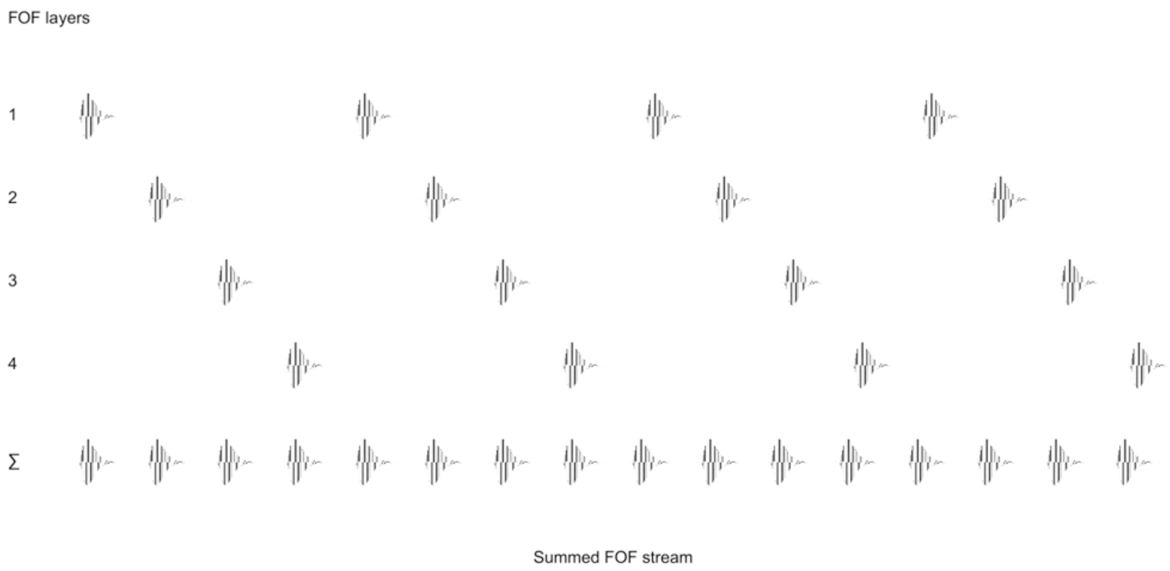

Figure 3. FOF layers

The advantage of this technique is that one can fade layers in and out of the stream (Fig3) creating the illusion that the fundamental frequency is morphing between octaves and without the intervening musical intervals. Because the formants in the spectrum are preserved, this technique allows for the production of dramatic gender morphing vocal timbres and the dynamic resizing of perceived acoustic objects. Octaviation is the mechanism by Michael Clarke creates the evaporation and coalition of FOF events in *Mälarsång*.

3.2 Pulsar Masking

In the *Pulsar Generator* (Roads, 2000), synchronized *pulse masks* can be generated to break up a *pulsar* stream*. Pulsars* are microsonic events, which unlike FOF or *Granular* synthesis have a wide variety of *particle shapes.* Unlike FOF octaviation, this allows for different types of regularized patterns to be generated. *Burst* masking generates periodic bursts of *pulsars* similar to those emitted by early impulse generators (Roads, 2001, p138) Channel *masking* produces concurrent channels of pulsar streams. *Stochastic* masking produces stochastic masks in the stream. Increasing the stochastic mask parameter breaks up the regularity of the pulsar pattern. Periods of activity and rest can be programmed in whole number of pulses; for example, users can program 4 *on* 3 *off* or 8 *on* 8 *off*. Depending on the *emission* rate, users can create interesting rhythms or amplitude modulation effects (Fig4).

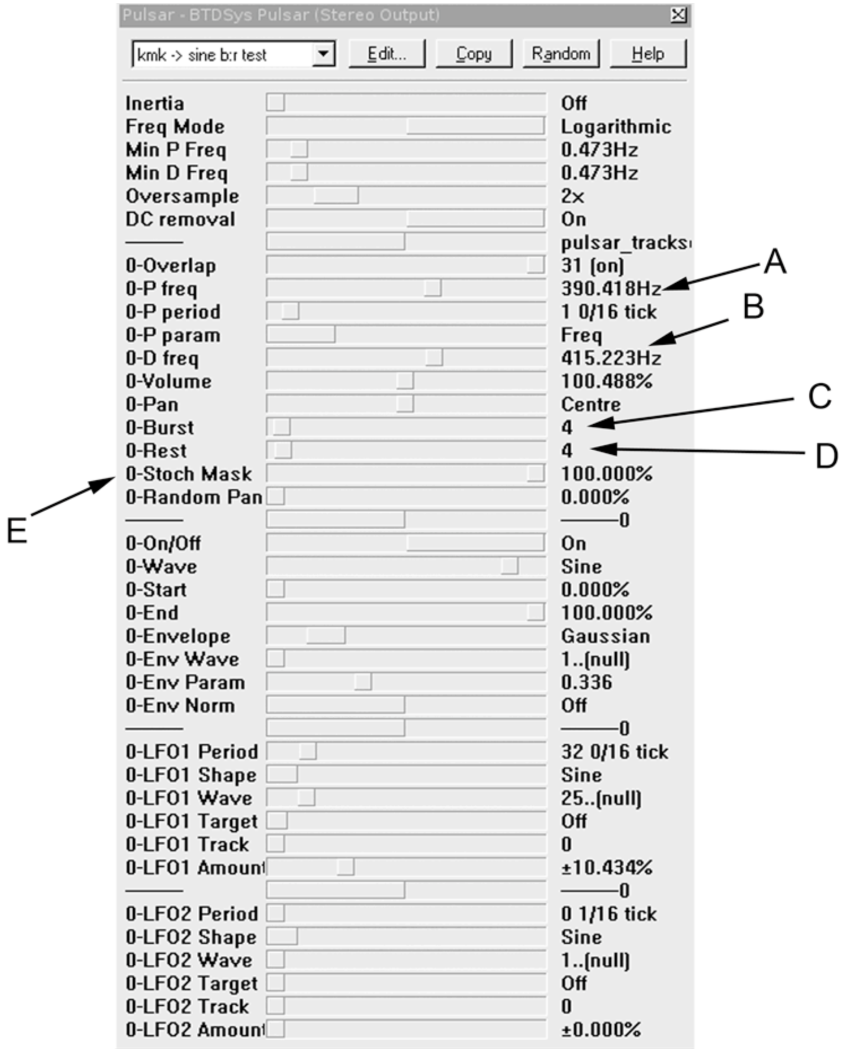

Figure 4. Pulsar Synthesis implementation of Jeskola's Buzz Machines.

A. *Emission* rate (Pulsar synthesis speak for fundamental frequency).
B. *Pulsar* frequency bandwidth (formant frequency)
C. 4 *Pulsars* on
D. 4 *Pulsars off* (masked)
E. Stochastic parameter. Introduces irregularities into the on/off pulsar patterns.

3.3 Increased flexibility

In creating a real-time particle synthesis environment the author decided from the onset to avoid the pre-calculations required in FOF synthesis and the limited programmability of masking in the available implementations of pulsar synthesis.

1. *Why limit masks to stochastic, channel or burst modes?*
2. *Why limit to a number of particles on and a number off ?*
3. *Why not have an arbitrarily programmable system which can be pre-programmed using familiar popular music sequencing techniques i.e. matrix editors / drum editors or created on the fly, generatively, interactively, manually or live?*
4. *Facilitate any number of particles on or off in any sequence order*.
5. *Avoidance of FOF pre-calculations?*
6. *Afford the morphing of any musical interval and not just octaves.*

4 MANAGING SOUND PARTICLES

The strategy for managing particle streams described here is used to control sound particles derived from a *wave* orientated *microsonic* synthesizer. It is termed *wave-orientated* because the sound particles produced are derived from the product of linear unipolar functions such as a *ramp* (positive going sawtooth) (Lynn and Fuerst, 1999) using elementary waveforms such as the sine-wave. The result is a formant rich timbre with a formant frequency independent from its fundamental frequency. In contrast, *Granular* synthesis generates microsonic streams by assigning multiple memory buffers per stream; consequently, it is suitable for granulating digitized sounds.

In the system under consideration, the density is determined by the fundamental frequency. The fundamental is thinned out by masks, which are synchronized to the fundamental. This could be seen as a limitation, however, as seen below, any arbitrary real-time pattern of particles can be produced at any given time.

Masks are generated in sequence with single periods of a reference signal to control single synthesized particles of sound. Because the technique is based on classic wave modulation synthesis, it can also be used to control individual cycles in classic sound synthesis such as *subtractive* or *additive* synthesis thus allowing the system to exhibit the unique characteristics of waves and microsounds.
The synthesis technique has been termed *Particle Wave Stream* synthesis (PWS) (Fig5) and was original presented by the author at the Modular 2002 conference at Thames Valley University in 2002 and 2003. It is part of a larger particle formant synthesizer system called the *Elementary Wave Engine (EWE)* which is continues to be developed at the University of Portsmouth in the UK.

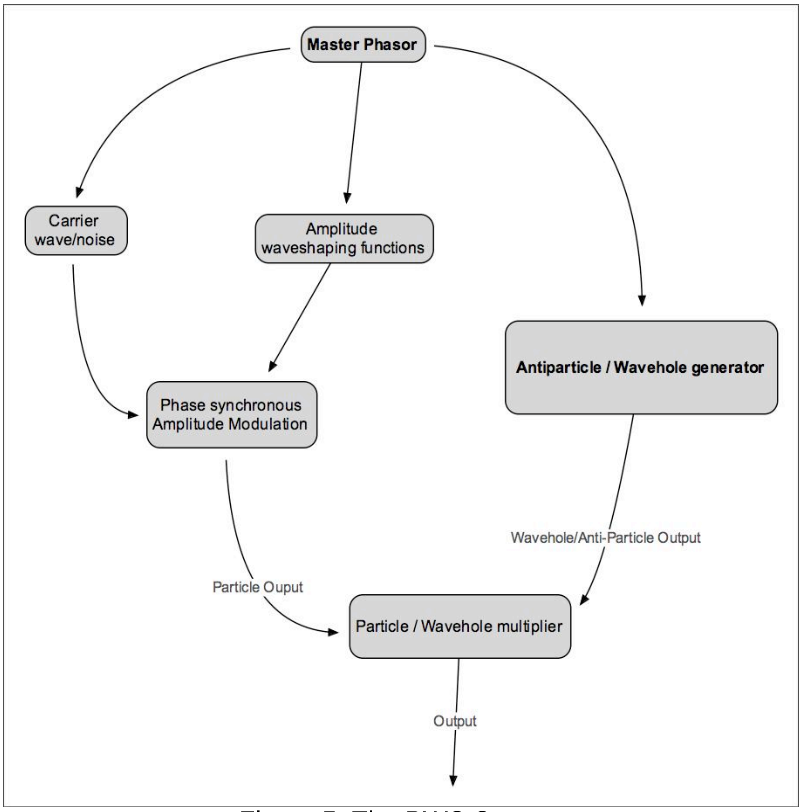

Figure 5. The PWS System

4.1 Data-flow Implementation.

In the original synthesis, implementation of the *EWE* (Fig6) a periodic ramp function ***r(t)*** is used as an indexing function, which is processed by a set of arithmetical operators, and functions to provide a variety of sound particle shapes. An elementary sound particle is created as a product of these amplitude functions and a phase synchronized cosine.

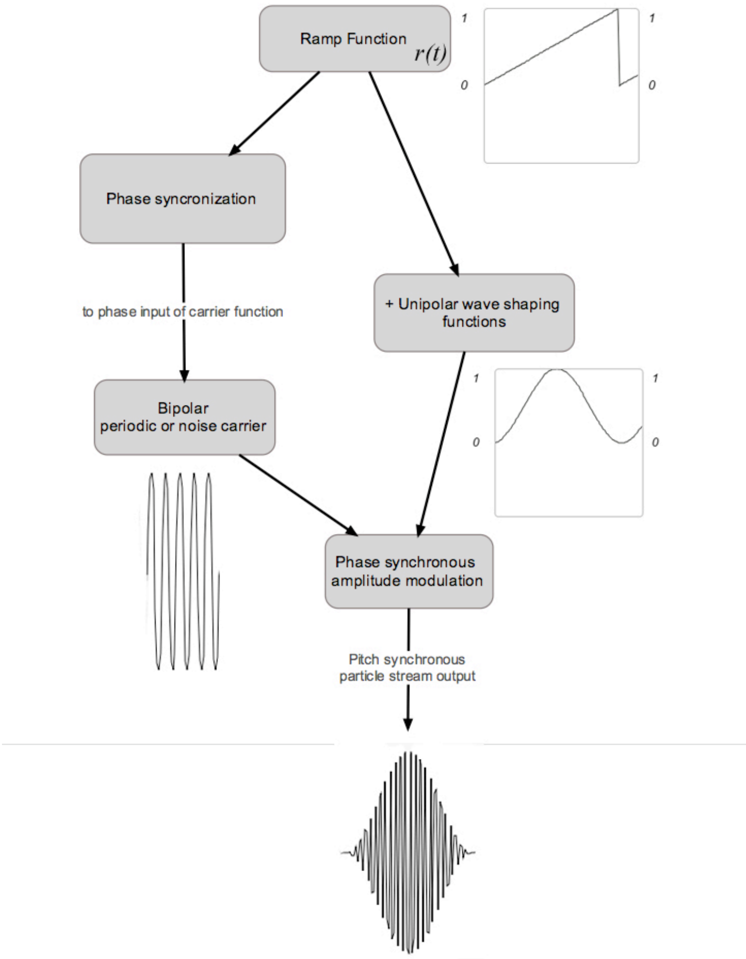

Figure 6. Original PWS implementation

4.2 Ramp to Pulse

A simple amplitude pulse mask ***p(t)*** can be created by processing ***r(t)*** for a positive going unipolar pulse function with a symmetrical duty cycle of 50% (Fig7).
This is done by integrating ***r(t)*** using the following simple logical condition:

p(t) = if *r(t)* > 0.5 then *p(t)* = 1 else *p(t)* = 0

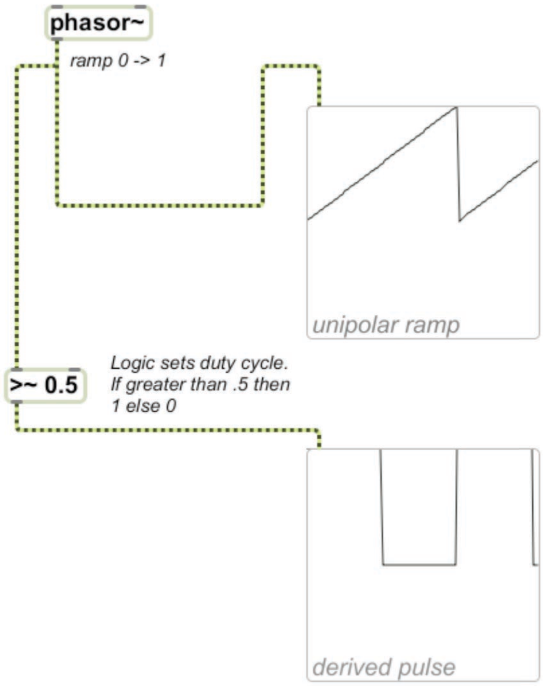

Figure 7. Simple pulse mask

This, however, is not particularly useful for masking the amplitude of a whole cycle of the ramp function as it produces a pulse twice the frequency of ***r(t)***. Instead, we process the ramp by halving its frequency (sampling rate) ***r(t2)*** so that we derive a pulse on every other cycle instead of every other *half* cycle.

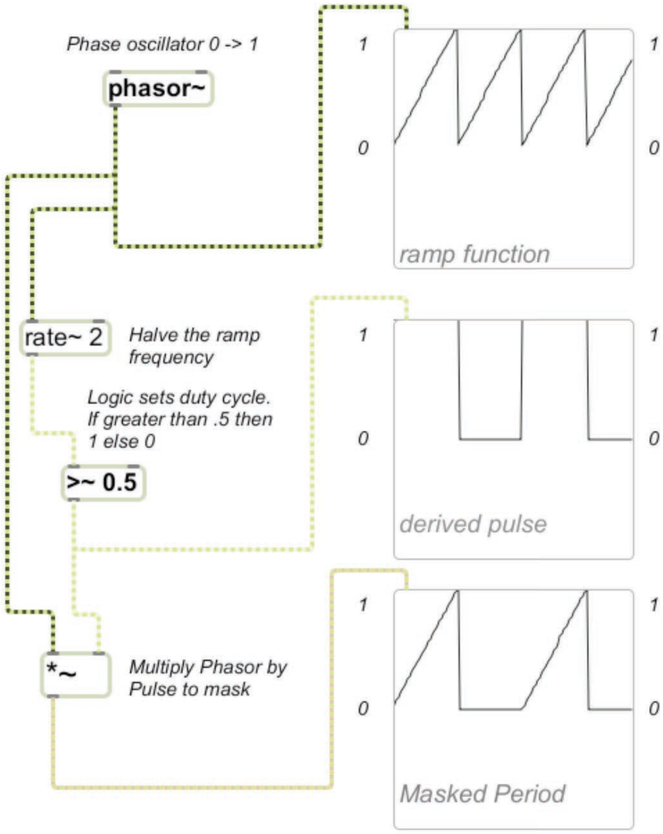

Figure 8. Individual *pulse masks*

Because each half period of ***p(t)*** is synchronized to each cycle of ***r(t2)*** they are pitch synchronous. The result is that ***p(t)*** creates consecutive *holes* in ***r(t).*** The effect of this is to halve the fundamental frequency of ***r(t)***. Though this is useful for illustrating the simplicity with which a *hole* can be created in a waveform this limits its use to creating synchronized sub-octave oscillators as in early subtractive synthesizers such as the *Roland SH-101* or a fixed one octave octaviation in FOF synthesis (Fig8).

4.3 Arbitrary holes in a waveform.

A far more flexible approach is to create pulse masks for each period, which affords arbitrary control, though at first this could seem impractical to implement. Creating completely arbitrary periods requires the user to decide how many periods will be controlled before hand. This poses a technical challenge in that the user might decide they wish to control some thousand periods over a determinate period.

A simple solution is to work with groups or patterns of particles. Repetition is avoided by concatenating varied or distinct patterns overtime.

These could be generated stochastically, generatively or manually. Repetition, however, may be desired. The system phase counter can be slaved to an outer master time source such as a music sequencer or drum machine.

The first step is to decide how many whole periods should fit into one pattern. This becomes the maximum pattern length. To illustrate the mechanism, a pattern of 8 periods has been chosen for this paper, though many more can be easily created. In the current implementation, patterns consist of 32 periods.

With the pattern length decided, we first divide the current fundamental frequency of the master ramp ***r(t)*** function by as many steps ***r(t/n).***

Max/Msp (Dechelle, 2008) offers a simple solution to achieve this with the *rate~* object which offers a simple way of scaling the frequency of an input ramp. Positive numbers scales the frequency downwards so that a ramp with a frequency of 1Hz would be scaled to 0.125 Hz (Fig9).

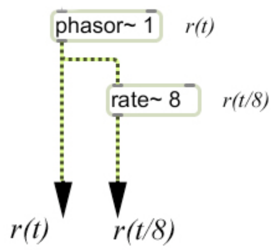

Figure 9. Dividing the master ramp frequency

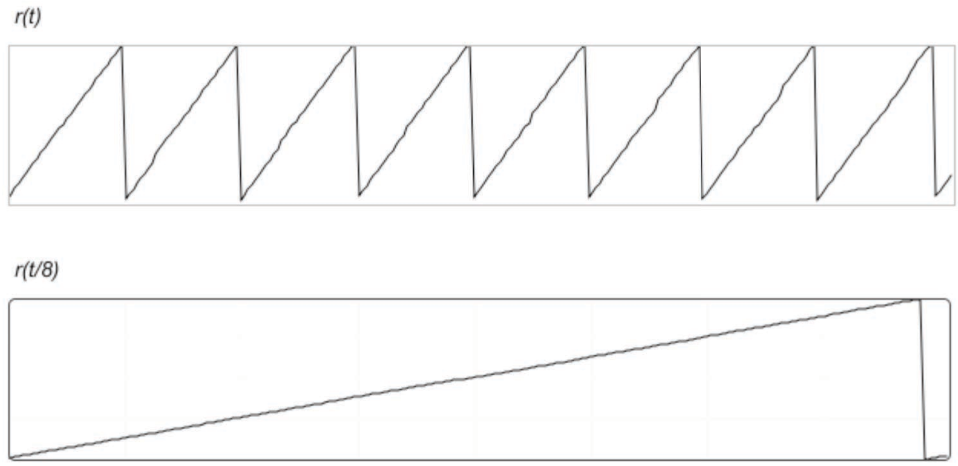

Figure 10. Master ramp r(t) and r(t/8)

r(t/8) is subsequently quantized into as many steps ***a(t)*** as the original division. These are repeated at the beginning of each new cycle of the derived ramp (Fig10).

4.4 Quantization

r(t/8) is then quantized to provide the number of steps, which are needed to generate the pulse masks, which are used to switch *on* and *off* individual periods of the original phase counter (Fig11). This is simply achieved by lowering the bit depth of **r(t/8).** In this case, we lower the bit depth to 3 in order to provide the 8 steps per phase cycle. Logically, increasing the bit depth increases the number of steps by a factor of ***2^n***.

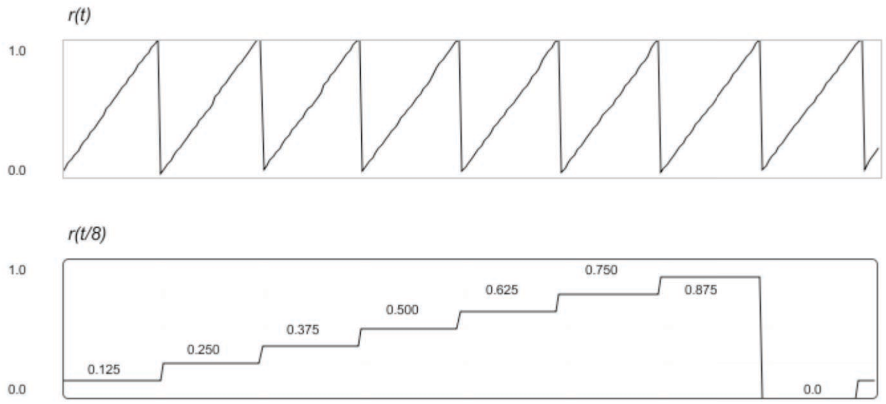

Figure 11. Quantized ***r(t/8***

Each instance of ***a(t)*** is sequentially fed into a parallel array of Boolean gates (Fig12) which generate a pulse based on an elementary logical condition*.* ***If a == a(t) then true (1)***, where ***a*** is the amplitude of the ramp at instance **r(t/8)[n].** These operands provide simple gates whose threshold are the steps created in ***r(t/8).***

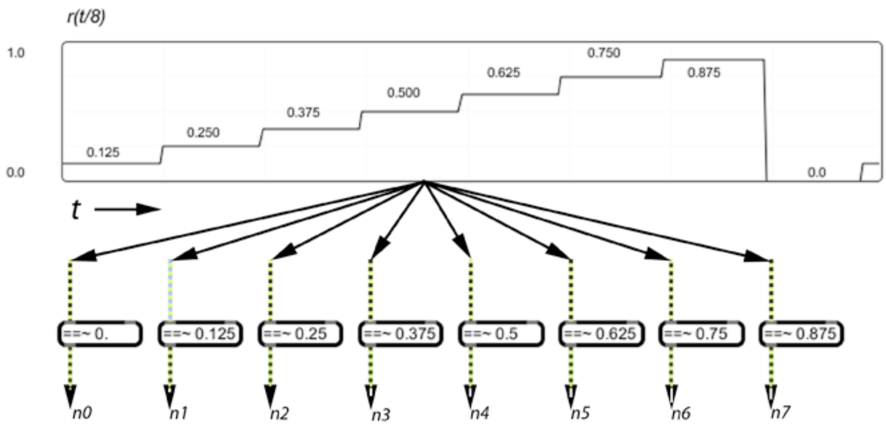

Figure 12. Sequential Boolean Gates

Multipliers are then used to switch the amplitude of each pulse on and off (Fig13).
A *wavehole* is derived from the product of a cycle of the original ramp with a pulse set at of 0.0.

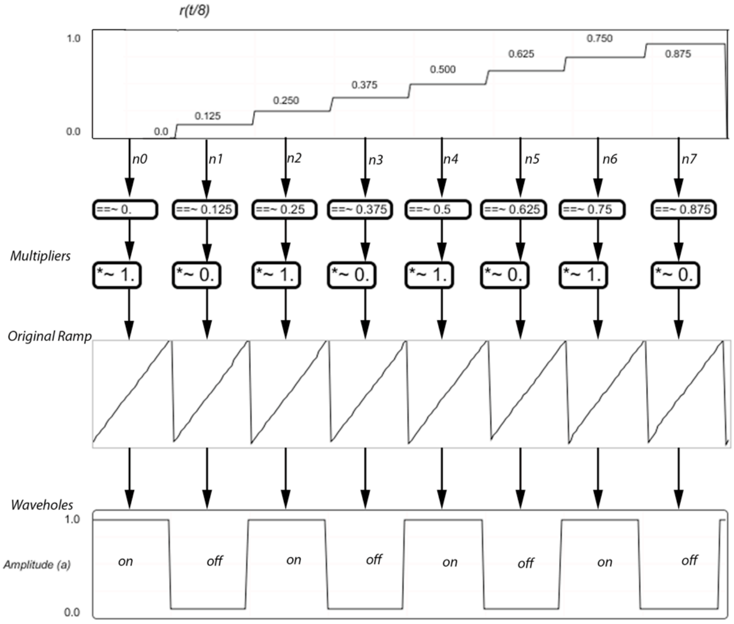

Figure 13. Sequence of on/off masks.

The values used in the example of Fig14 generate the consecutive *waveholes* of Fig14.

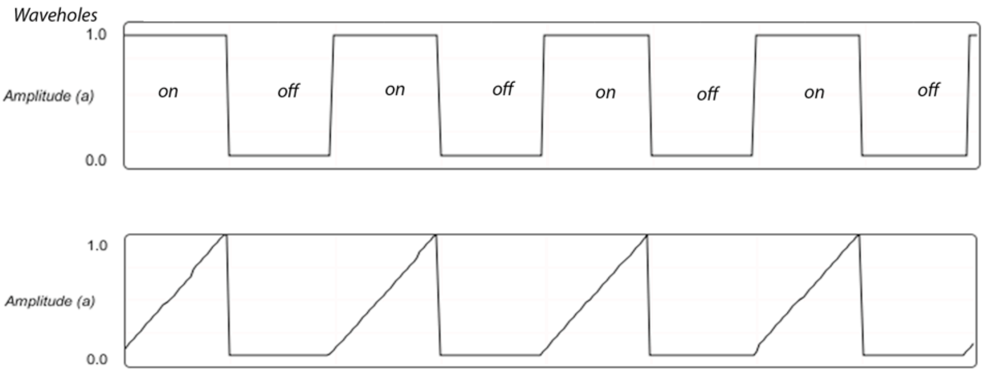

Figure 14. Consecutive *wavehole*

By using arrays of switches, the user can create patterns in which individual cycles of the original ramp are switched *on* or *off* (Fig15).

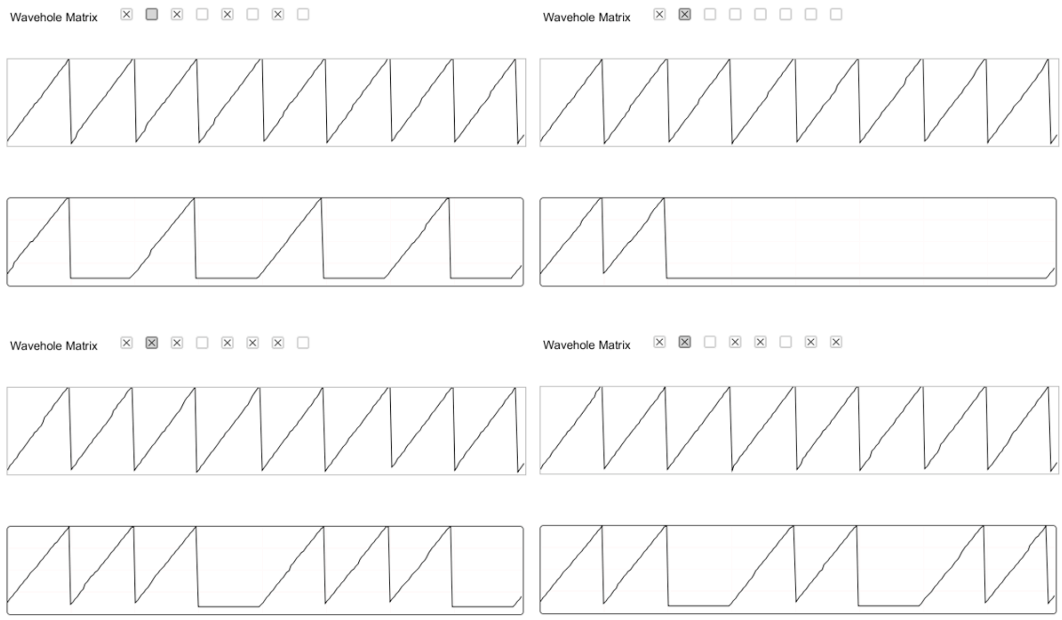

Figure15. Examples of single Wavehole Patterns

Patterns can be concatenated (Fig16) in step or real-time. A scheduling system is in development, which allows long sequences of wavehole patterns to be created. Predetermined rhythmic patterns, generative sequences can be generated in response to live events.

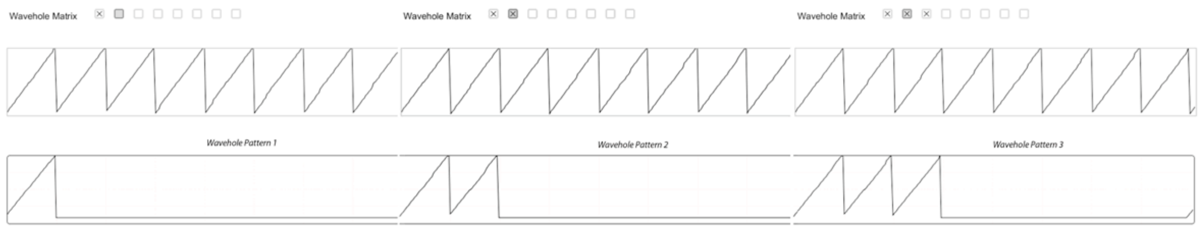

Figure16. Concatenated Wavehole

Similarities to Pulsar synthesis can be seen in this method, however it is the arbitrary nature rather than fixed types of masking in *pulsar masking* and the addition of real-time morphing capabilities, makes the Wavehole generator unique. To the authors knowledge the patterning or modulation of single cycles of trivial waveforms is novel.

4.5 Morphing, switches or rather lack of them

A crucial development in the overall EWE synthesis environment was the decision to substitute the switching system in favour of level controls including simple on and off state parameters. The premise is that for a system to be truly flexible and allow glitch free morphing from any state to another across the complete range of parameters and subsystems it would be necessary to eliminate any kind of binary interruption. The wavehole sill has switches but these are temporarily employed to set one of the two extremes of the wavehole state, however, they are not linked to the morphing capabilities of the system (Fig17).

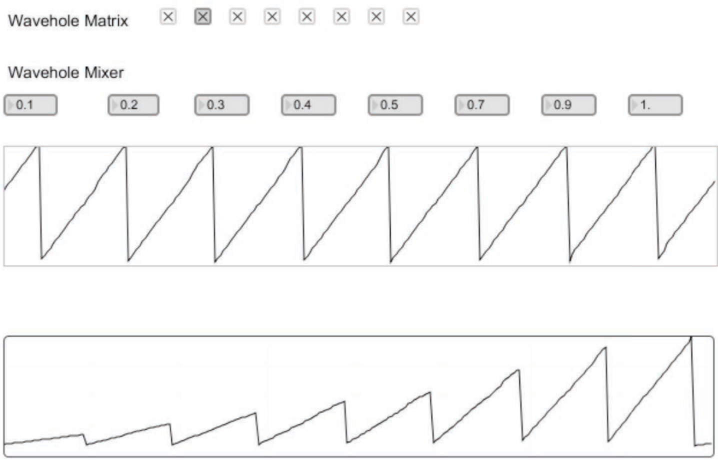

Figure 17. Numerical faders instead of switches to allow smooth morphing of patterns

4.6 Applying waveholes to sound particles

The wavehole generator is employed in the *EWE* system to generate what are conveniently termed *anti-particles*. The flexibility and ease with which sound particles can be made to appear and disappear affords a versatile particle synthesis engine (Fig18).

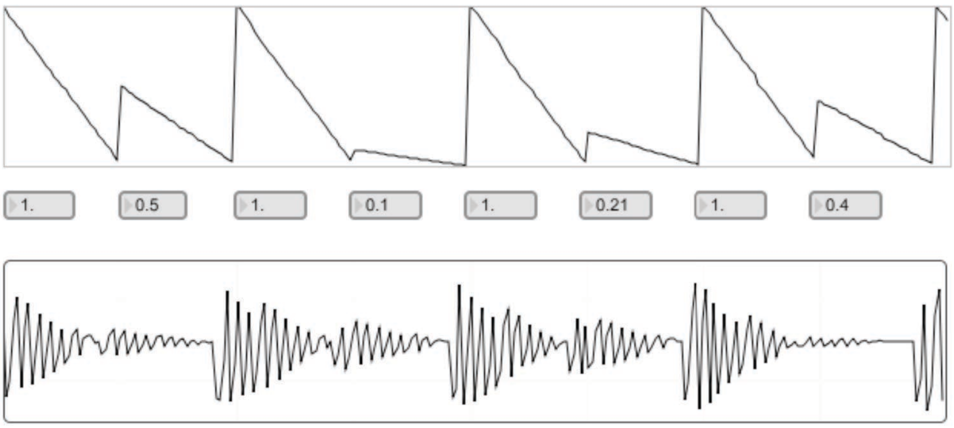

Figure 18. The *waveholes* acting on particles.

Patterns of sound particles and cycles of a waveform can be stored and accessed via a scheduled storage system, which is accessed through pre-programming, generative, algorithmic or live process.

5 FURTHER DEVELOPMENTS

Because each wavehole is synchronized at the sample level to a master phase counter, that is, the original ramp it allows for independent processing of waveform cycles and microsonic events within a stream. *Glisson*s (Roads, 2001, p121), for example, are possible by generating independent pitch trajectories in the formant pitch of individual particles. Single audio events can be panned, reverberated and distributed in an ambisonic system for use in immersive systems.

6 CONCLUSION

The wavehole generator is capable of creating a wide variety of particle fission and fusion textures. It can be particularly effective in re-creating other established fundamental frequency manipulation techniques such as *octaviation* in *FOF* or *stochastic, burst and channel pulse masking* in *Pulsa*r synthesis however, its strength is in the ease with which patterns of sound particles can be programmed sequenced and morphed and is not confined to a single type or category of particle masking class. Users are free to create their own. The system avoids the pre-calculation associated with FOF synthesis and uses established sequencing found popular music production techniques for pure sound synthesis.

The EWE was first developed in 2001 and demonstrated at the Modular 2002 at Thames Valley University in Ealing, London as part of the Modular 2002 conference. Further developments of this system were shown at the Modular 2003 (see ref electro-musician). The complete EWE system, which is in development at the University of Portsmouth, is expected to be released in the second quarter of 2009.

The object of the development was to create a system capable of producing a wide variety of micro-sonic timbres for sound design and composition purposes retaining much of the functionality of traditional wave based synthesis techniques, yet, providing a hybrid of both wave and microsound functionality.

The system is a hybrid, a cross between a wave-shaping / additive synthesiser biased towards the production and control of synthesized sound particles through a variety of wave-shaping techniques.

The current wavehole generator was prototyped using the graphical data-flow programming environment Max/Msp/Jitter (Zicarelli Pukette); however, it can easily be implemented in other programming languages. A low level implementation is expected to follow in late 2009.

It is relative easy to plug the wavehole generator into existing linear trivial oscillators or other wave orientated particle synthesizers such as VoSim. By linear the author means where an index counter is used as the source driver rather than an electronic analogue modelled sawtooth which exhibits noise or other signal irregularities.

7 REFERENCES

D. Gabor, "Theory of communication," J. IEE (London), vol. 93, pp. 429-457, 1946.

Roads Computer Music Journal Automated Granular Synthesis of Sound
Curtis Roads p61 Volume 2, Number 2 September 1978

Kaegi, Werner, and Tempelaars. Vosim - a new sound synthesis system. Journal of the Audio Engineering Society, 26(6):418--425, 1978.

Giovanni De Poli , Aldo Piccialli, Pitch-synchronous granular synthesis, Representations of musical signals, MIT Press, Cambridge, MA, 1991, pp 140-141

Roads Computer Music Journal Introduction to Granular Synthesis
Curtis Roads p11-13 Volume 12, Number 2 September 1988

Truax Computer Music Journal Real-Time Granular Synthesis with a Digital Signal Processing Computer Barry Truax p 14-26 Volume 12, Number 2 September 1988

Rodet, Xavier (1984). “Time-domain formant wave-function synthesis”. Computer Music Journal, 8(3): 9–14.

C. Roads 1999, “Time Scales of musical structure,” in Actes V. Academie Internationale de musicque electro-acoustique, F Barriere and G. Bennett, Eds. (Editions Mnemosyne, Bourges, France

C. Roads 1999, “Time Scales of musical structure,” in Actes V. Academie Internationale de musicque electro-acoustique.

F Barriere and G. Bennett, Eds. (Editions Mnemosyne, Bourges, France, 1999)

Introductory Digital Signal Processing with Computer Applications, SOL 2 Rev t/a by Paul A. Lynn and Wolfgang Fuerst (Paperback - Nov 1, 1999)

Roads, C. Composing with Pulsars, AES Journal of the Audio Engineering Society, Volume 49 Number 3 2001 March pp 134-147

E. R. Miranda, E. (2002), Computer Sound Design: Synthesis techniques and Programming, Oxford: Focal Press, pp 101-102

Microsound. (Roads, 2002). March 2002 ISBN 0-262-18215-7

M. S. Puckette. Theory and Technques of Electronic Music. Retrieved online on OCtober 7, 2008 at http://crca.ucsd.edu/~msp/techniques/latest/book.pdf, December 8 2003.

Dechelle, F, A brief history of MAX. Retrieved online on October 8, 2008 at 15:55pm http://freesoftware.ircam.fr/article.php3?id_article=5

8 DISCOGRAPHY

Jean Batisite Barrierre, Cheode 1, Unpublished, Festival International des musiques expérimentales de Bourges, 1 June 1983

Clarke, M, Mälarsång, Refractions (MPSCD003), 1987.