Pulsar Synthesis Revisited: Considerations for a MIDI Controlled Synthesiser

Thomas Wilmering[1], Thomas Rehaag[2], and André Dupke[3]

[1] Centre for Digital Music (C4DM)
Queen Mary University of London
London E1 4NS, UK
thomas.wilmering@eecs.qmul.ac.uk

[2] Intelligent Sounds & Music
Cologne 51065, Germany

[3] Hamburg-Audio
Periana 29710, Spain

Abstract. In this paper we present an implementation of a software synthesiser based on pulsar synthesis to be used in conventional digital audio workstations supporting common plugin standards. After reviewing basic pulsar synthesis, we describe limitations of this synthesis technique and novel parameters we developed to overcome these for the design of a MIDI controlled implementation. The developed keyboard instrument can be easily played by composers and music producers familiar with software synthesisers using traditional synthesis techniques based on virtual oscillators. We also discuss aesthetic considerations in the design, the spectra of complex grain waveforms, and the effect of parameter changes on pulsar train spectra.

Keywords: sound synthesis, pulsar synthesis, granular synthesis

1 Introduction

Sound synthesis and manipulation based on granular synthesis (GS) has been investigated in great detail since its first computer-based implementation by Curtis Roads in 1978 [1]. Newer implementations range from sound generators for grain clouds to digital audio effects based on sound granulation [2]. An extensive overview of granular synthesis techniques is given in [3]. However, although granulation and granular synthesis is a widespread technique in contemporary electronic music composition, synthesisers based on pulsar synthesis (PS) are less common. This is especially the case for implementations that can be easily integrated into digital audio workstations as plug-ins to be used in the same fashion as virtual analog synthesisers

PS is a type of granular synthesis whose name originates from spinning neutron stars that emit periodic signals in the range of 0.24 to 642 Hz [4]. Basic

9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)
19-22 June 2012, Queen Mary University of London

pulsar synthesis generates a periodic pulsar train controlled by various parameters which we describe in more detail in section 3. The aim of our research is to extend PS, and to overcome the limitations inherent in the technique when implemented as a keyboard instrument where the pitch is controlled by MIDI note numbers. The goal is not only the creation of a composition tool for experimental sound design, but also to make PS accessible to, and intuitively usable for music producers of popular electronic music, stressing some of its apparent resemblances to virtual analog synthesis.

After briefly describing granular synthesis and its musical application and reviewing basic pulsar synthesis, we describe the extended PS and the aesthetic considerations taken into account during the development of a plug-in synthesiser in conjunction with *hamburg-audio*[4]. Lastly we discuss the sonic capabilities by means of complex grain waveform spectra, the effect of parameter changes on pulsar train spectra, and sequencing in the microsound domain. Audio examples accompanying this paper can be found online[5].

2 Granular Synthesis

GS is based on the theory of acoustical quanta by British physicist Dennis Gabor, who suggested that every sound can be decomposed to a family of functions derived from time and frequency shifts of a single Gaussian particle. Gabor developed a mathematical representation for acoustical quanta by relating a time-domain signal with a frequency-domain spectrum [5][6]. The duration of a grain of sound is usually in the range of 1 to 100ms, ranging near the threshold of human perception. Furthermore, a grain is characterised by its waveform w shaped by an envelope v.

The combination of large numbers of grains over time makes it possible to create sound patterns and *grain clouds* resulting in atmospheric sounds [3]. The basic form of a grain generator is shown in figure 1.

The first person to use Gabors theory as a composition tool was Xenakis [7]. Musical pieces to mention are for example Metastasis (1954), Concret PH (1958) and Analogique A-B (1959), the latter being described in [8]. Curtis Roads did further research in the field of GS and created various related compositions and computer programs. He developed software implementations such as *Cloud Generator* and *Pulsar Generator* [3][4]. Other important composers in this context are Paul Lansky and Barry Truax. Until the end of the 20th century, GS could mostly be found in the works of composers linked to scientific research institutions and in compositions outside of the world of popular music. However, new genres and subcultures emerged since the beginning of this century with composers having different degrees of knowledge about the institutional framework of computer music. Inspired by the works of the established composers, music styles such as *glitch* or *IDM* (Intelligent Dance Music) are heavily influ-

[4] http://hamburg-audio.com
[5] http://isophonics.net/content/pulsar-synthesis

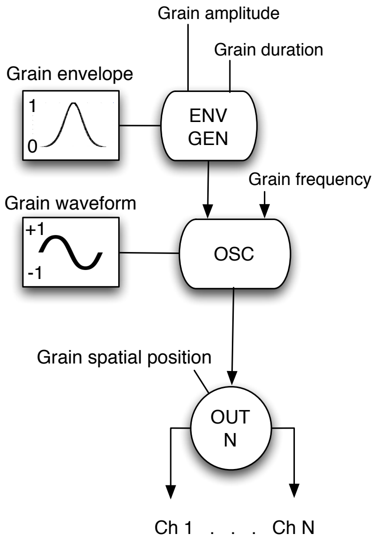


Fig. 1. Basic grain generator consisting of a Gaussian grain envelope generator and a sinusoidal grain waveform. The grains can be spatially placed in N channels.

enced by GS and *Granulation*. This development has also been discussed in the musicological literature [9][10][7].

In recent years granular synthesis and granulation-based effects have found their way into popular music and music production tools with a variety of digital audio effects inspired by the aforementioned newly emerging genres and sub-genres of electronic music. However, PS has as of today gained less attention since its first appearance.

3 Pulsar Synthesis

PS has first been presented as a computer-based granular synthesis technique by Curtis Roads and Alberto de Campo in 1999 [4]. The pulses and pitched tones produced with PS are similar to those of earlier analog musical instruments, e.g. the Ondioline or the Hohner Elektronium which are based on filtered pulse trains [11][4]. Nevertheless, PS is implemented in the digital domain, therefore taking advantage of the processing power and flexibility of modern computer systems.

In its simplest form a pulsar train (as shown in figure 2) is controlled by two main parameters, the fundamental frequency (pulsar frequency):

$$f_p = \frac{1}{p} \tag{1}$$

$$p = d + s \tag{2}$$

where the period p of the pulsar consists of the *pulsaret* width (duty cycle) d, and the intergrain time s.

The duty cycle frequency (formant frequency) is described by:

$$f_d = \frac{1}{d} \tag{3}$$

which determines the width of the pulsaret within the pulsar.

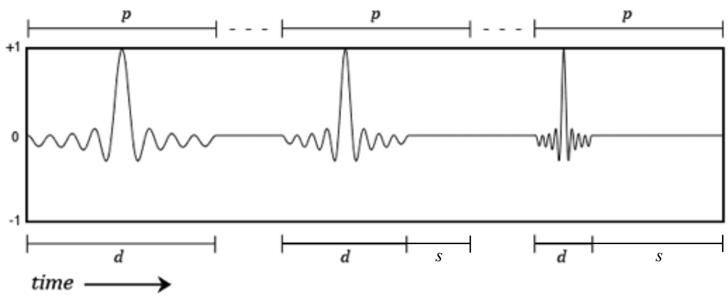


Fig. 2. A pulsar consists of a *pulsaret* of width d and a following silent part (intergrain time s). The period p of the fundamental determining the pitch is an independent parameter from the pulsaret width.

The periodic pulsar train G can be expressed as the convolution of the pulsaret with an impulse train:

$$G_{w,v,d,p}(t) = w_d v_d * \sum_{k=-\infty}^{\infty} \delta(t - kp) \tag{4}$$

where w_d and v_d are scaled versions of the pulsaret waveform and envelope with length d, and δ is the Dirac delta function.

As opposed to pulse width modulation (PWM) in analog synthesisers, where the duty cycle of a rectangular waveform is set by a ratio to the fundamental period, in PS the duty cycle is an independent parameter from the fundamental frequency. Moreover, the pulsaret is characterised by the pulsaret waveform w and the pulsaret envelope v. The pulsaret waveforms and envelopes can be of arbitrary shape. However, Roads [4] proposed some standard waveshapes with the initial introduction of PS and investigated the effect of the grain envelope on the grain's spectrum. The standard waveforms include *sine* and *multicycle sine*, as well as bandlimited pulses and cosmic pulsar waveforms stressing the synthesis technique's relationship to waveforms emitted by neutron stars. Typical envelope shapes are for example Gaussian, sine, linear or exponential decay or attack. The envelope causes a resonant main band and several sidebands, smearing the original waveform spectrum [12]. Figure 2 shows pulsars of constant pitch with varying duty cycle frequency; the pulsaret waveform is a band-limited pulse.

As f_p and f_d are independently variable parameters, we may encounter the case of $d > p$. In this case we can apply overlapped pulsaret-width modulation

(OPulWM), where several grains (pulsarets) overlap. This overlap is defined as the time interval during which two or more grains are played simultaneously, and can be calculated by the difference between the grain rate and grain duration [13]. A different approach to deal with this problem is to cut off the pulsar and spawn a new one without overlapping. However, in practice both approaches have disadvantages. While OPulWM generally leads to cancellation at higher numbers of overlapped grains (and increased CPU load), cutting of a grain may lead to sudden changes in the amplitude, introducing unwanted high frequencies into the spectrum (this effect may be dampened by a cross-fade parameter at the cutoff point). In section 4.6 we describe a hybrid synthesis approach used in our implementation which presents an optional compromise between true PS an playability throughout all octaves.

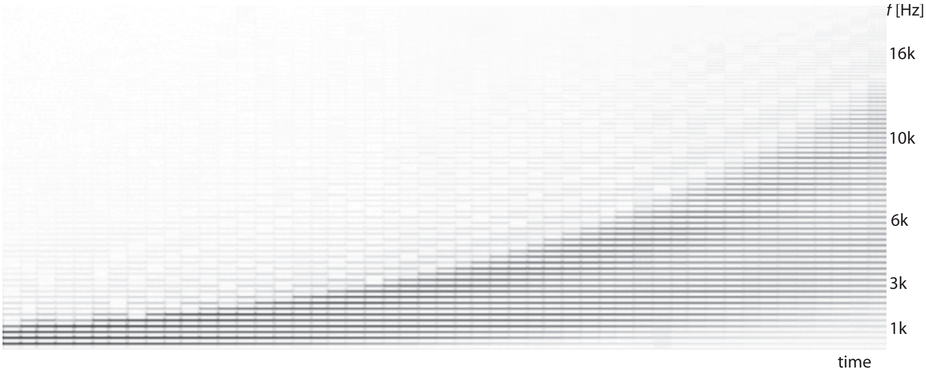


Fig. 3. Spectral effect of increasing the duty cycle frequency at a constant fundamental frequency (pulsaret width modulation).

4 Pulsar Synthesis for a MIDI Controlled Synthesiser

For the design of a PS synthesiser that can be played as a keyboard instrument we introduced a number of novel parameters, some to improve the playability, others purely for aesthetic considerations. In this section we present some of these parameters, the motivation behind their introduction, and the sonic implications. We implemented the synthesis in the *Nuklear* synthesiser as an *Audio Unit*[6]/*VST*[7] plug-in with four parallel pulsar train generators. Its graphical user interface (GUI) is shown in figure 4.

In addtion to the sound generators the synthesiser features various modulation sources which can be mapped to arbitrary parameters. These include 8

[6] https://developer.apple.com/library/mac/#documentation/MusicAudio/Conceptual/AudioUnitProgrammingGuide/Introduction/Introduction.html

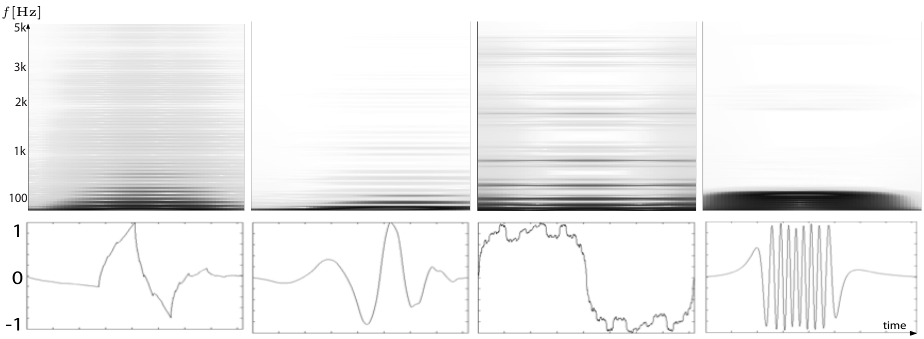
[7] http://ygrabit.steinberg.de/˜ygrabit/public_html/index.html

Fig. 4. Graphical user interface of *Nuklear*, a plug-in synthesiser based on Pulsar Synthesis

ADSHR envelopes, 8 low frequency oscillators (LFO) and control sequences that can be programmed in a 16-step sequencer. Moreover, the sum on the output can be shaped by two filters and an effect section consisting of a delay effect and a distortion effect. In this paper, however, we focus specifically on the pulsar train generators and the parameters we introduced to extend PS to our needs.

4.1 Pulsaret Waveforms

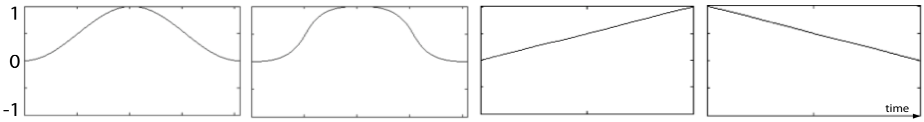
In addition to the standard PS waveforms (see section 3) we implemented several other waveforms known from classic virtual oscillators, such as *sawtooth*, *rectangular* and *triangular*. Furthermore, we added experimental waveforms, among them a selection of shapes based on wavelet functions. It should be noted that in our case we perform a sonification of wavelet functions rather than mathematical operations related to their original purpose, the wavelet transform for multi-resolution signal analysis. We chose the wavelet shapes from an aesthetic point of view and achieved some interesting results with regards to composition and sound design. In our experiments we investigated acoustic characteristics of compactly supported orthonormal wavelets as introduced by Daubechies [14], and biorthogonal wavelets, a family of wavelets with the property of linear phase [15]. The wavelet shapes that are included in our implementation were selected to cover a wide range of sounds. Figure 5 shows some of the waveforms and their respective spectra.



a) b) c) d)

Fig. 5. Pulsaret waveforms with corresponding spectra above. a) based on Daubechies wavelet 2; b) based on Daubechies wavelet 5; c) based on biorthogonal decomposition wavelet 3.5; d) cosmic gravitational wave. The spectrograms are 2048-point fast Fourier transform plots with a Blackmann-Harris window. The duty cycle frequency is 10.87Hz.

To shape the waveforms we implemented the envelope shapes shown in figure 6 in addition to a rectangular envelope with a constant value of 1.



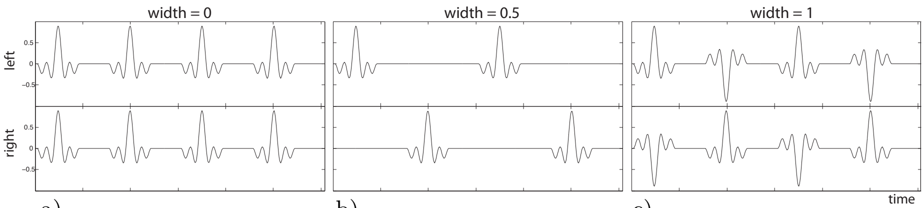
a) b) c) d)

Fig. 6. Pulsaret envelopes in *Nuklear*: a) Hann type I; b) Hann type II; c) linear attack; d) linear decay.

4.2 Stereo Width

A stereo widening effect can be set independently for each of the four parallel pulsar trains. The effect is achieved by alternating the pulsars between the stereo channels (see figure 7). This is implemented as a parameter allowing intermediate settings. Low settings of this parameter only decrease the amplitude in an alternating fashion between the channels, instead of muting every other pulsar completely.

At low fundamental frequencies in the infrasonic range the alternating pulsars are clearly distinguishable, while at higher notes the stereo widening effect is perceived. Moreover, even at low settings harmonics of half the fundamental frequency are introduced to the spectrum. This is due to the fact that effectively an amplitude modulation at half the fundamental frequency is performed with a 180° phase shift between the stereo channels. At width settings above the middle position the alternating pulses are inverted and mixed with the opposite channel.



a) b) c)

Fig. 7. Stereo width parameter: At 0 every pulsar plays on both channels (a). At the centre position the pulsars alternate between the channels (b). At 1 the alternating pulses are mirrored negatively onto the other channel (c). This parameter also allows intermediate settings.

4.3 Pitch-Dependent Pulsar Phase

Another novel parameter we introduce is the *pitch-dependent pulsar phase*. We define this parameter as a time shift of every other pulsar within the fundamental period in the range of 0° to 360°. Figure 8 shows the effect this time-shift has on the pulsar train, both in the time and frequency domain. A phase shift of

360° translates to every second pulsar coinciding with the next pulsar, effectively transposing the pulsar train down by one octave.

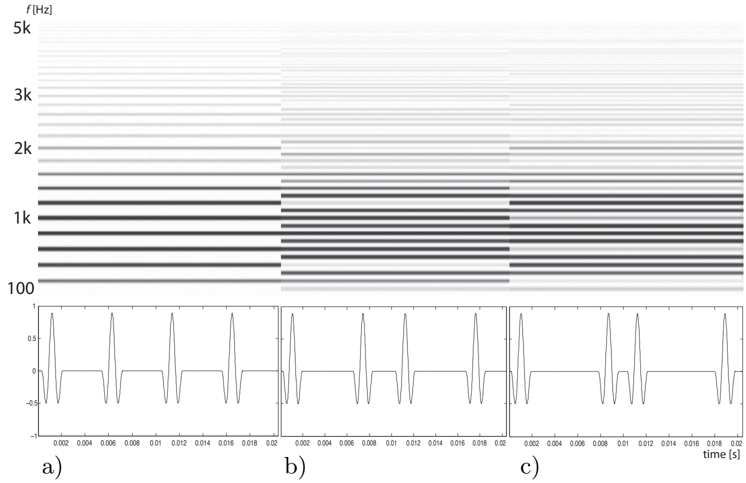


Fig. 8. Effect of the pitch-dependent pulsar phase shift on the pulsar train. The spectra above are produced by playing continuous loops of the pulsar sequences below. a) phase = 0°; b) phase = 90°; c) phase = 180°.

This phase shift technique introduces harmonics at half the original fundamental frequency, with varying magnitudes for the partials depending on the phase-shift amount. Modulating the parameter with an LFO or envelope curve results in an effect reminiscent of a classic *phaser*. However, the common *phaser* effect consists of a time-modulated additive delay line in the range of up to 2ms, independently from the note's pitch [16]. In the infrasonic domain the pulsar phase parameter produces rhythmic changes in the pulsar train.

4.4 Grain Overlap

As mentioned in section 3, when implementing a granular synthesiser one needs to take into account the case of overlapping grains. In the case of PS this is relevant when the fundamental frequency f_p exceeds the duty cycle frequency f_d. Overlapping grains in PS can produce a smooth sound due to the negative intergrain time, while they at the same time largely preserve the overall formant structure of the pulsar train. However, high numbers of overlapping grains may lead to cancellation and high CPU load.

In *Nuklear* it is possible to define an *overlap limit*. When this limit is reached, the pulsarets are scaled in such a way, that a set maximum number of pulsarets fits in to the fundamental cycle. While this technique changes the sound characteristics of true PS, it allows the user to play the synthesiser over all octaves without having to worry about undesired missing notes in the higher registers.

Figure 9 shows a pulsar train with pulsarets scaled to fit into the fundamental period p with an overlap limit of *0* (a), and overlapping pulsarets with unchanged duty cycle frequency f_d (a).

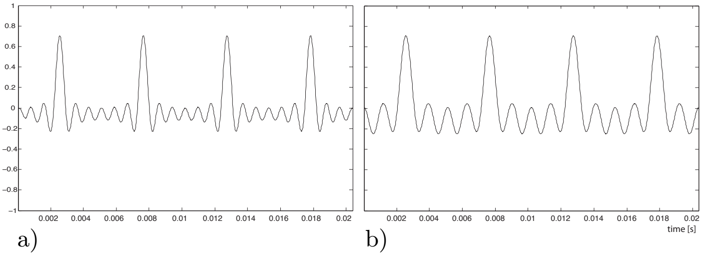


Fig. 9. Pulsar train with a pulsaret waveform of a bandlimited pulse with 6 harmonics. a) overlap limit = 0, the pulsaret is scaled to fit into the fundamental period ($d = p$); b) the pulsarets overlap and keep the original duty cycle frequency f_d.

Figure 10 shows how allowing grains to overlap preserves the formant structure (a). Automatic scaling of the duty cycle period to fit into the fundamental period to avoid grain overlap results in different spectra. In the latter case the pulsar train generator behaves similarly to a virtual oscillator using the pulsaret waveform (in this example d is scaled to $0.95p$).

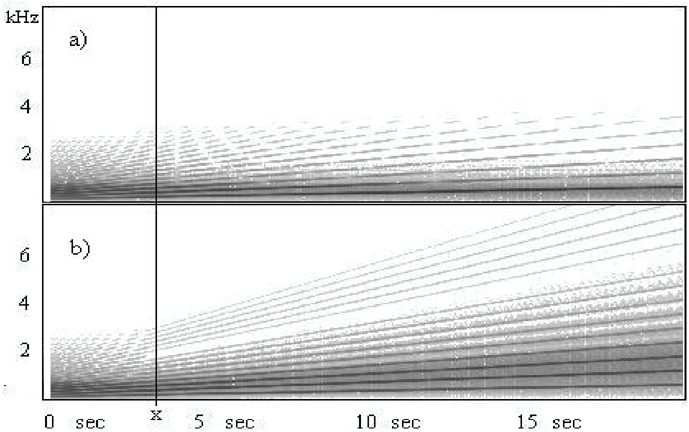


Fig. 10. A pulsar train generated with a prototype implementation. f_p rises linearly from 100 Hz to 600 Hz. The duty cycle frequency f_d is at 200 Hz, the pulsarets are two sine wave cycles shaped by a Gaussian envelope. At the point marked x, f_p is $0.95f_d$. a) the grains start to overlap, b) the pulsarets are scaled to $d = 0.95p$ to avoid overlap.

4.5 Sequencing in the Microsound Domain

Pulsar masking is the controlled deletion of pulsars from the pulsar train replacing it with silence. Roads [4] proposes three forms for this technique: *burst*, *channel*, and *stochastic masking*.

Burst masking mutes pulsars at regular intervals at a given *burst ratio b:r*, where b defines the burst length and r is the rest length in pulsaret periods. For instance, a *burst ratio* of 3:1 produces a sequence of 3 pulsarets and one period of silence, which can be denoted by a binary sequence 111011101110, etc. (Figure 11b). The effect is a form of amplitude modulation, where the fundamental frequency is broken up by a subharmonic factor $b+r$. If the fundamental frequency is in the infrasonic range, rhythmic patterns are perceived. *Channel masking* is defined as the muting of pulsarets on two channels in an opposite manner. Therefore, the stereo width parameter at its middle setting as described in section 4.2 can be seen as channel masking with a *burst ratio* of 1:1. *Stochastic masking* mutes pulsars randomly according to a given ratio, i.e. the ratio of the number of outcomes of 1 (play) to the number of outcomes of 0 (mute), when all outcomes are regarded as equally likely.

Building on the idea of *burst masking* we implemented a binary pulsar masking sequencer we call *microsequencer*. In its current form it allows a masking sequence of up to 8 pulsar cycles in which pulsars can be set to either *on* or *off*. Figure 11c) shows the pulsar pattern 10111001, and the resulting spectrum of a note played with this masking setting.

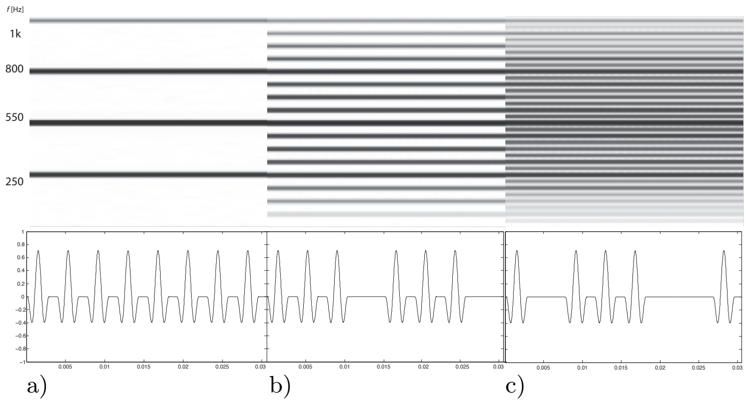


Fig. 11. Spectral effect of the *microsequencer*. The spectra above are produced by playing continuous loops of the pulsar sequences below. a) regular pulsar train without masking; b) masking every 4th pulsar introduces harmonics at $\frac{1}{4}$ of the fundamental frequency (sequence 1110); c) a more complex pulsar sequence (10111001).

4.6 Pulsar/Virtual-Analog Hybrid Synthesis

In addition to true PS our synthesiser offers a synthesis that can be described as a hybrid between PS and synthesis using *classic* virtual oscillators. This development is partly motivated by the problems with regards to notes with high fundamental frequencies and the resulting multiple grain overlap (see section 4.4). Moreover, due to the nature of PS many sounds lack low frequencies in the spectrum and the perceived *warmth* associated with them. We address this problem by lowering f_d relative to f_p, limiting the effect that low notes are not only perceived as a series of high-pitched clicks. By introducing a parameter h, we can gradually move between PS ($h = 1$) and virtual oscillator based synthesis ($h = 0$). Here, the duty cycle frequency f_d takes the form:

$$f_d(h, e, p) = hf_e + (1 - h)f_p \ , 0 \leq h \leq 1 \quad (5)$$

where h is the hybrid synthesis parameter setting and f_e is the duty cycle frequency setting on the GUI.

Note that if $h = 0$ then $f_d = f_p$, i.e. in this setting the pulsar train generator can be used in the same way as a virtual analog oscillator. Moreover, an oscillator may be defined as a special case of a pulsar train generator, where $f_p = f_d$. Thus, all the complex waveforms and parameters developed for PS as described in sections 4.1 to 4.5 are still available in this synthesis mode. Furthermore, by utilising multiple pulsar train generators and the *microsequencer* we can design unusual oscillators, for instance, by rotating through a sequence of waveforms. Figure 12 shows such a signal produced by employing all four pulsar train generators with $h = 0$, each of which use a different waveform and pulsar sequences (1000, 0100, 0010, 0001).

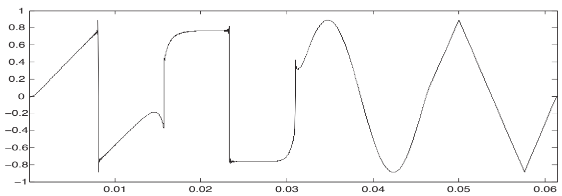


Fig. 12. Signal produced by employing four pulsar train generators each playing a different pulsar sequence with different waveforms. The microsequencer is set to a length of 4 and the sequences 1000, 0100, 0010, 0001. The duty cycle d is of the same length as the fundamental period p

5 Conclusions

We presented an implementation of a software synthesiser based on extended pulsar synthesis (PS). A number of novel parameters were introduced in order to enhance the synthesis technique specifically for use in a keyboard instrument. We demonstrated that PS is well suited for the development of tools for music composition and production, not only within the academic musical framework, but also as an alternative to established virtual synthesisers used in music production studios. To overcome some of the subjective limitations of PS we proposed a hybrid synthesis technique which makes it possible to gradually move between PS and classic virtual oscillator based synthesis. Moreover, due to the nature of PS the synthesiser can be used to demonstrate the relationship of rhythm and pitch by comparing pulsar trains with fundamental frequencies in the infrasonic range with pulsar trains in the harmonic range. The former produces rhythmic patterns, the latter produces pitched notes.

The professional plug-in implementation *Nuklear* is based on this research and capable of producing a range of sounds different in character from established synthesisers. It received positive reviews in the professional literature, and has been awarded with the *Innovation Award* from *Computer Music* magazine [17].

References

1. C. Roads, "Granular synthesis of sound," *Computer Music Journal*, vol. 2, no. 2, pp. 61–62, 1978.
2. U. Zölzer (Ed.), *DAFX - Digital Audio Effects*, J. Wiley & Sons, second edition edition, 2011.
3. C. Roads, *Microsound*, MIT Press, Cambridge, MA, USA, 2001.
4. C. Roads, "Sound composition with pulsars," *Journal of the Audio Engineering Society*, vol. 49, no. 3, pp. 134–147, 2001.
5. D. Gabor, "Acoustical quanta and the theory of hearing," *Nature*, vol. 159, no. 4044, pp. 591–594, 1947.
6. M. J. Bastiaans and A. J. van Leest, "Gabor's signal expansion and the gabor transform based on a non-orthogonal sampling geometry," *6th International Symposium on Signal Processing and its Applications, Kuala Lumpur, Malaysia*, vol. 1, pp. 162–163, 2001.
7. P. Thomson, "Atoms and errors: Towards a history and aesthetics of microsound," *Organised Sound*, vol. 9, no. 2, pp. 207–218, 2004.
8. I. Xenakis, *Formalized Music (Revised Edition)*, Pendragon Press, Stuyvesant, NY, USA, 1992.
9. K. Cascone, "The aesthetics of failure: "post digital" tendencies in contemporary computer music," *Computer Music Journal*, vol. 24, no. 4, pp. 12–18, 2002.
10. P. Sherburne, "12k: Between two points," *Organised Sound*, vol. 9, no. 2, pp. 225–228, 2002.
11. L. Fourier, "Jean-Jacques Perrey and the ondioline," *Computer Music Journal*, vol. 18, no. 4, pp. 18–25, 1994.
12. D. Keller and C. Rolfe, "The corner effect," *Proceedings of the XIIth Colloquium of Musical Informatics, Gorizia, Italy*, 1998.
13. D. L. Jones and T. W. Parks, "Generation and combination of grains for music synthesis," *Computer Music Journal*, vol. 12, no. 2, pp. 27–33, 1988.
14. I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, 1988.

15. A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 45, no. 5, pp. 485–560, 1992.
16. H. Speckmann, "Time-related sound processors," [Online]. Available: http://www9.dw-world.de/rtc/infotheque/sound_processors/soundprocessors.html, 2004.
17. *Computer Music*, Number 172. Future Publishing Ltd., 2011.