Audio Engineering Society

Convention Paper 7792

Presented at the 126th Convention
2009 May 7–10 Munich, Germany

Phase and Amplitude Distortion Methods for Digital Synthesis of Classic Analogue Waveforms

Joseph Timoney[1], Victor Lazzarini[2], Brian Carty[3] and Jussi Pekonen[4]

[1,2,3] The Sound and Digital Music Technology Group, NUI Maynooth, Ireland
[1]JTimoney@cs.nuim.ie
[2]VLazzarini@nuim.ie
[3] Brian.M.Carty@nuim.ie

[4] Dept. of Signal Processing and Acoustics, TKK Helsinki University of Technology, Espoo, Finland
[4]Jussi.Pekonen@tkk.fi

ABSTRACT

An essential component of digital emulations of subtractive synthesizer systems are the algorithms used to generate the classic oscillator waveforms of sawtooth, square and triangle waves. Not only should these be perceived to be authentic sonically, but they should also exhibit minimal aliasing distortions and be computationally efficient to implement. This paper examines a set of novel techniques for the production of the classic oscillator waveforms of Analogue subtractive synthesis that are derived from using amplitude or phase distortion of a mono-component input waveform. Expressions for the outputs of these distortion methods are given that allow parameter control to ensure proper bandlimited behavior. Additionally, their implementation is demonstrably efficient. Lastly, the results presented illustrate their equivalence to their original Analogue counterparts

1. INTRODUCTION

Interest in the development of algorithms to simulate Analogue signals in the digital domain has spurred an area of research known as Virtual Analogue (VA) Modelling (also VA Synthesis in the specific case of sound generation) [1,2,3,4,5,6]. The key issues in these works are related to approximating the modelled signals efficiently and avoiding or minimising aliasing distortion.

This paper investigates a number of methods for amplitude and phase distortion to generate classic waveforms for Analogue subtractive synthesis. These methods are very closely related, as they operate on sinusoidal inputs, yielding complex spectra as a result of waveshape distortion. They feature some interesting characteristics that are useful for this particular application: low computational cost, easily described dynamically-variable spectra and simple parametric control. In addition, they provide means of matching their Analogue counterparts as closely as a digital signal will allow.

This paper is organised as follows. We will first discuss the basic techniques of amplitude distortion and their use in the emulation of classic waveforms. This is followed by an exploration of phase distortion. In this discussion we will examine two different methods: the original oscillator phase increment distortion and the novel technique of allpass filter coefficient modulation.

2. AMPLITUDE DISTORTION TECHNIQUES

Amplitude distortion provides perhaps the simplest means of modification of a sinusoidal waveshape. It operates by a non-linear amplification of the original signal. Mathematically stated, it is basically function mapping of some signal (such as a sinusoid with frequency $\varpi = 2\pi f_0 t$ and amplitude k):

$$s(t) = f(k\cos(\varpi)) \tag{1}$$

When $f(.)$ is non-linear, the output spectra will be complex. Another name commonly given to this group of techniques is *waveshaping* [7]. This technique allows for very low-cost implementations, as the function mapping can be implemented with simple table lookup.

The result of waveshaping depends on two factors: the non-linear transfer function shape and the amplitude of the input signal. Generally speaking, increasing the gain of the input sinusoid will cause the distortion to be more pronounced and the output spectra will contain more components. In order to de-couple the amplitude of the signal and its spectral richness, we normally provide a means of normalising the output signal for the full range of input amplitudes. This is done using a normalising function, which in its turn is dependent on the shape of the waveshaping transfer function. Once this is accomplished, the amplitude k in equation 1 becomes the *distortion index* that controls spectral richness.

Traditionally, the method of transfer function selection for waveshaping has been by spectral matching based on (finite) polynomial transfer functions. However, even though this provides a perfectly band-limited and matched spectrum for a given index of distortion, it is not suitable for the emulation of classic Analogue waveforms. The major problem with it is that the use of polynomials has a crippling side-effect: very un-natural sounding dynamic spectra. That is, for values of the distortion index other than the ‘matched’, the spectra can deviate wildly from the desired one.

This is problematic because: (i) we will need to vary the spectra to limit the bandwidth of the output signal to avoid aliasing; (ii) we want to preserve the spectral envelope, so that waves of various fundamentals look and sound alike. The solution is to look for alternatives to polynomial transfer functions that do not exhibit these side-effects. We will not be concerned in finding only waveshaping functions that produce perfectly band-limited spectra, but ones where the bandwidth can be effectively limited by judicious use of the index of distortion.

The cases examined below will all employ functions with infinite Taylor’s series polynomial expansions, which in theory will provide non-bandlimited spectra. However, in practice, we can avoid aliasing by reducing the distortion on higher fundamentals.

2.1. Triangle wave generation

The simplest example of a classic-waveform waveshaper transfer function is represented by the inverse trigonometric functions, *arcsin(.)* and *arccos(.)*. The result of employing these functions with full distortion (k=1) is the production of a non-bandlimited triangle wave. Triangles have a very steep roll-off in their spectral envelope, decaying by N^{-2}, where N is the harmonic number. Given these conditions, in many applications, aliasing is minimised by default. However, we can improve on that. By employing progressively smaller values of k, inversely proportional to the fundamental frequency of the signal, we will be able to band-limit the signal very well.

2.2. Hyperbolic tangent transfer function

The example of the triangle wave provides an interesting method of operation: we can find a function that generates the (non-bandlimited) shape we want, then we can find ways of band-limiting or minimising the aliasing distortion in the signal. If we want, for instance, to produce a square wave from a sinusoid input, it is just a matter of hard-clipping the signal. However, this will introduce aliasing that is much more pronounced and intrusive than in the triangle wave case, as now the spectrum decays as N^{1} and the higher components will be more disruptive.

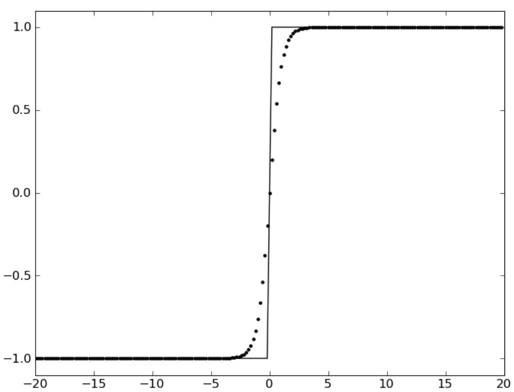

Figure 1. Clipping (continuous line) and tanh (dots) waveshaping transfer functions

If we look at the reasons for this heavy aliasing, we will see that it is the transition around 0, at the join of the two pieces of the clipping function (Figure 1) that causes the problem. What we need here is a function that is smooth around 0, a sigmoid. The hyperbolic tangent *tanh*(.) (Figure 1, dots), a favorite in other waveshaping applications (e.g. overdrive distortion effects) will do very well.

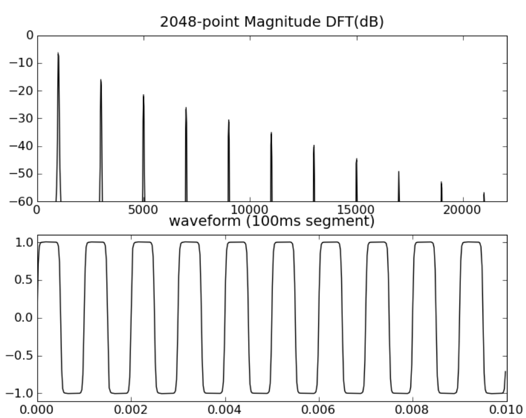

Figure 2. Square wave produced by hyperbolic tangent waveshaping

In addition, we will need to control the index of distortion to minimise aliasing, as *tanh*(.) has again an infinite Taylor's series. This can be measured experimentally and a simple expression to determine the modulation index *k* from the fundamental f_0 is found in

$$k = \frac{12000}{f_o \log_{10} f_0} \tag{2}$$

Intuitively, we can see why this is the case: as the amplitude of the input increases, the sharper, the transition at 0 is, as more of the 'flat' portion of the transfer function is used to waveshape the signal. A plot of the resulting square wave is shown on Figure 2.

2.2.1. Sawtooth wave generation

Sawtooth generation using waveshaping poses a small problem. It is impossible to find a transfer function that directly generates the desired shape from a sinusoidal input. It is simple to demonstrate why: waveshaped signals have a half-period symmetry that does not exist in the sawtooth shape. If we take for instance, just the positive half of a sine wave, after any function mapping, the rising side will always be mirrored by the decreasing one. It is easy to see that we cannot amplitude shape it into a sawtooth because of this (we will however be able to approximate it by phase distortion as discussed later on).

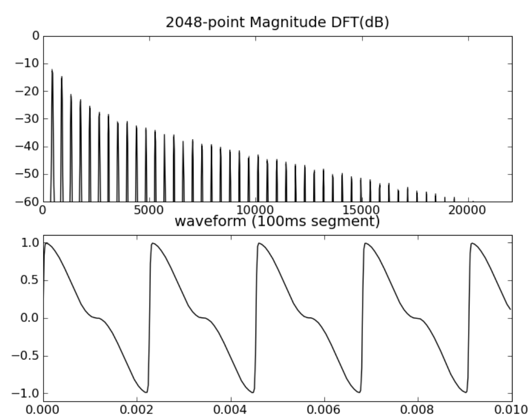

Figure 3. Sawtooth from square wave

However, if we incorporate a means of ring-modulating a square wave produced by waveshaping with a cosine, we can produce the missing even harmonics. By mixing this signal with the original square wave a sawtooth can be produced. The following expression demonstrates this:

$$saw(t) = square(\omega)(\cos(\omega)+1)$$
$$= \sum_{n=0}^{\infty}\left[\begin{array}{l}\frac{1}{2n+1}\sin([2n+1]\omega)+\\ \frac{2n+2}{4n^2+8n+3}\sin(2[n+2]\omega)\end{array}\right] \quad (3)$$

This will indeed approximate the sawtooth shape quite well (Figure 3). There is a slight deviation in even harmonic weights in comparison to the classic sawtooth, but this is only perceptible in harmonic 2, where the error is 2.5 dB. From harmonic 4 upwards, the difference is less than 0.5dB.

2.3. Exponential waveshaping and Modified FM

Waveshaping can also be used to produce pulse waveforms. This is accomplished by using another very common function, which also has an infinite Taylor's series, the exponential, *exp(.)*. The result of distorting the amplitude of a sinusoid with an exponential transfer function is given by a well known expression:

$$s(t) = \exp(k\cos(\omega)) = \sum_{n=-\infty}^{\infty} I_n(k)\cos(n\omega) \quad (4)$$

where $I_n(k)$ are known as Modified Bessel functions. Signal normalisation here is actually simpler than in other cases. The normalising function turns out to be the inverse of the exponential itself, *exp(-k).* Unipolar pulses of various bandwidths can be produced with this expression (Figure 4). In fact, this is the basis of a more general expression for a technique we call *Modified Frequency Modulation* (ModFM) synthesis [8]:

$$s_{ModFM}(t) = \exp(k\cos(\omega_m)-k)\cos(\omega_c) = $$
$$= \frac{1}{\exp(k)}\sum_{n=-\infty}^{\infty} I_n(k)\cos(\omega_c+n\omega_m) \quad (5)$$

This technique, based on a variation of the original FM expression [9], has, like that method, a number of applications, depending on its parameters *c, m* and *k* (the carrier and modulator frequencies; and the modulation index). In addition, unlike FM, it is also useful for the synthesis of classic waveforms. This is because one of its characteristics is an ordered, natural, dynamic spectral evolution. FM, on the other hand, features very synthetic-sounding spectral changes, as a consequence of how Bessel functions scale its components. The effect is comparable to the problems with polynomial waveshaping.

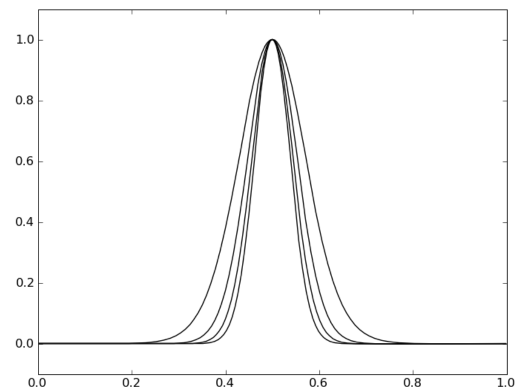

Figure 4. Exponential waveshaping pulse waveforms of various widths, for $k=5$ (wider) through to 20 (narrower)

2.3.1. Pulse integration

ModFM can be used to generate other types of classic waveforms. This is accomplished by integrating the pulse waveform generated by certain configurations of the algorithm [8]. For instance, with $c{:}m = 1$, we have a unipolar pulse, which, when integrated will produce a sawtooth waveform. One advantage of the method is again the ability of controlling the bandwidth of the signal, therefore minimising aliasing distortion. Setting $\omega_m = \omega_c$ in equation 5 and integrating it with respect to frequency yields an expression for the resulting spectrum of a ModFM sawtooth (Figure 5):

$$s_{saw}(\omega) = \frac{2}{\exp(k)(n\omega)}\sum_{n=1}^{\infty} I_n'(k)\sin(n\omega) \quad (6)$$

where $I_n'(k) = 0.5[I_{n-1}(k)+I_{n+1}(k)]$. The integration is easily implemented by a 1st order IIR filter:

$$H(z) = \frac{1}{1-z^{-1}} \quad (7)$$

To limit the values of k so that we have a full-spectrum sawtooth without aliasing, we can solve the following optimisation problem (*sr* denotes the sampling rate):

$$\max_{k}\left\{20\log_{10}\frac{1}{(n+1)}\frac{I_{n+1}'(k)}{I_1'(k)}\right\}\leq -90\,\mathrm{dB},\qquad (8)$$
$$n=\left\lfloor\frac{sr}{2f_0}\right\rfloor$$

Similarly, we can obtain a square wave by producing a bipolar pulse, with $\omega_m = 2\omega_c$.This, when integrated in the same way as above, will yield a square wave. Further integration can be used to generate triangle waveshapes. The only extra care we have to take when implementing ModFM for classsic waveforms is that the generated signals will contain a lot of energy at 0Hz. For practical applications, we will need to correct this with either some way of extracting the mean of the pulse signal or by applying a good DC blocker at the output [10].

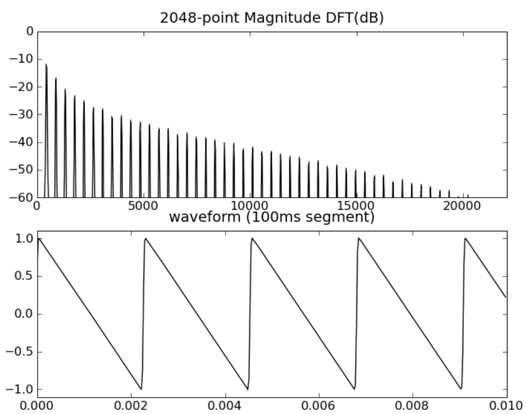

Figure 5. ModFM sawtooth wave

3. PHASE DISTORTION TECHNIQUES

Another way of generating complex waveforms by distorting sinusoids is to apply a non-linear mapping to its phase, instead of amplitude. This will have a similar effect of modifying the waveshape and as a consequence, introducing higher components to the signal. In fact, any such operation is a subset of a more general technique, Phase Modulation (PM), a synthesis technique related to FM. The original methods of Phase Distortion (PD) were based on the manipulation of a wavetable oscillator phase increment [11]. Recently, a new method, based on the modulation of allpass filter coefficients, was proposed as an alternative [12].

3.1. Phase increment distortion

The technique of PD as implemented in the Casio CZ series of synthesizers is an inexpensive means of generating digital versions of classic Analogue waveforms. The output signal is produced by using a non-linear phase increment to a sinusoidal oscillator, as in

$$s_{pd}(t) = -\cos(2\pi\phi_{pd}(t)) \qquad (8)$$

where $\phi_{pd}(t)$ is the distorted phase increment. The non-linear phase can be broken down in terms of a linear increment $\phi(t)$ [normalised in the range of 0-1] and a modulation function $md_{pd}(t)$:

$$\phi_{pd}(t) = \phi(t) + md_{pd}(t) \qquad (9)$$

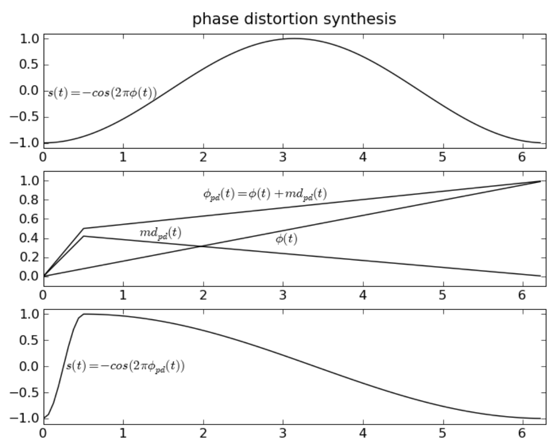

Figure 6. Phase Distortion: on top, the original waveform; in the middle plot, the original phase, the modulation function and the non-linear phase function; the bottom graph shows the PD output waveform

The PD function for the generation of a sawtooth-like waveshape is shown on Figure 6. For this case, the non-linear phase increment is affected by the use of a sawtooth wave modulator. The spectral characteristics of the output will depend on the exact shape of this sawtooth, particularly the balance between the length of its rise and decay portions. Here, we effectively have a form of complex PM, with a sawtooth modulator and a *c:m* ratio of 1. This will generate a significantly rich spectrum, however, the amount of distortion will need to be carefully limited to avoid aliasing. Spectral evolutions here are smooth, because of the use of very small values for the effective PM index (a useful characteristic of using complex modulators). Other waveshapes, such as the square wave, can be created using different non-linear phase increment functions.

3.2. Phase shaping

In complement, if PD is a method of non-linear *phaseshaping*, in analogy to waveshaping, we can produce similar-sounding outputs by choosing alternative phase 'transfer functions'. For instance, we can approximate the 'kinked' phase increment function of Figure 6 using the following expression:

$$s_{pd}(t) = -\cos(2\pi\phi(t)^{1/k}) \tag{10}$$

where $\phi(t)$ is the normalised phase increment (moving from 0 to 1) and the phaseshaping transfer function is $f(x) = x^{1/k}$. The PD index is now k, controlling the bandwidth of the signal. All relevant modulation functions and the resulting waveform are shown on Figure 7, for k =5. In similar fashion, several other phaseshaping functions can be proposed to produce a variety of output waveforms.

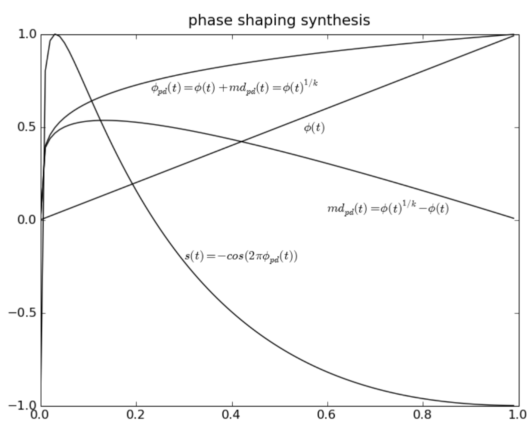

Figure 7. Phase shaping functions and output waveform

3.3. Allpass filter coefficient modulation

The modulation of the coefficient of a single-stage allpass filter was shown in [12] to introduce phase distortion into a sinusoidal input signal. This property can be exploited to create a dynamic phase distortion system for waveform generation. The transfer function of a time-variant first-order allpass filter is [12]

$$H(z) = \frac{-m(t) + z^{-1}}{1 - m(t)z^{-1}} \tag{11}$$

where $m(t)$ is the modulation function. The phase distortion introduced by this filter at time t and at a frequency ω is

$$\phi(\omega, t) = -\omega + 2\tan^{-1}\left(\frac{-m(t)\sin(\omega)}{1 - m(t)\cos(\omega)}\right) \tag{12}$$

Assuming that the desired phase distortion function is known, equation 12 can be rewritten to give an expression for the unknown coefficient modulation function:

$$m(t) = \frac{-(\phi(\omega, t) + \omega)}{2\sin(\omega) - (\phi(\omega, t) + \omega)\cos(\omega)} \tag{13}$$

where the simplifying approximation for the tangent has been employed [13]

$$\tan\left(\frac{\phi(\omega, t) + \omega}{2}\right) \approx \frac{(\phi(\omega, t) + \omega)}{2} \tag{14}$$

To create a sawtooth waveform using this technique the phase modulation function of Figure 6 must be shifted and scaled appropriately by

$$\phi(\omega, t) = \frac{md_{pd}(t)((1 - 2P)\pi - \omega)}{(1 - 2P)\pi} - (1 - 2P)\pi - \omega \tag{16}$$

where $P \in (0,1)$, and is the fraction of the waveform period during which the function $md_{pd}(t)$ is increasing. For example, to produce a sawtooth wave a value of P=0.25 is suitable. Substituting equation 16 into equation 13 gives the required coefficient modulation function $m(t)$ for the allpass filter. Smoothing of this modulation function is desirable to inhibit output waveform glitching caused at points of rapid change in the coefficient value. A useful smoothing function was found to be:

$$m'(t) = \alpha + \tanh(m(t) - \beta) \tag{17}$$

where $\alpha = 1.45$ and $\beta = 1.5$ was determined empirically to reduce this phenomenon significantly [14].

Applying a cosine to the input of this allpass filter whose coefficient modulation is given by equation 17 gives the resulting spectrum and waveform shown in the lower panel of Figure 8. The shape of the waveform is that of a sawtooth, albeit smoothed, whose frequency components are given in the plot of the spectral magnitude in the upper panel of Figure 8. The power of the highest frequency components of this wave (> 10,000Hz) is below -60dB. This steeper roll-off of spectrum is related to the modulating smoothing given in equation 17. Adjustment of the values for α and β can be made to alter the bandwidth of the signal spectrum.

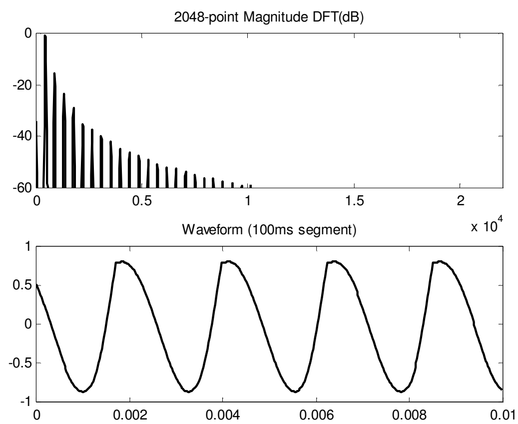

Figure 8. Modulated allpass filter output sawtooth wave

To produce a square wave using this technique equation 16 should be modified to:

$$\phi(\omega, t) = \frac{mds_{pd}(t)((1-P)\pi + \omega)}{(1-P)\pi} - (1-P)\pi - \omega \qquad (18)$$

where the modulation function $mds_{pd}(t)$ is at twice the frequency of $md_{pd}(t)$, i.e. 2ω. Again, smoothing of the resulting modulation function can be performed using equation 17. A plot of the output square wave spectrum and waveform are given in Figure 9. A trace of the glitching phenomenon is observable at the corners of the square wave; however, no significant alias components can be seen at any frequency.

Having obtained a square wave it is straightforward to produce a Triangle wave using the procedure outlined in [15].

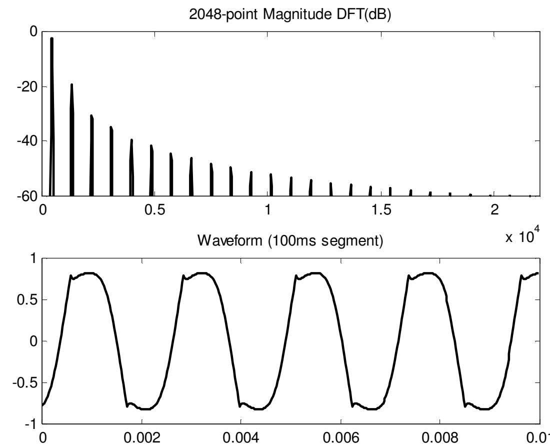

Figure 9. Modulated allpass filter output square wave

4. CONCLUSION

This paper has presented a set of novel techniques for the production of the classic oscillator waveforms of Analogue subtractive synthesis that are derived from using amplitude or phase distortion of a mono-component input waveform. All techniques are straightforward to implement and demonstrated graphically to produce good results. Future work will look at assessing the perceptual accuracy of these various techniques in reproducing the sound of their respective Analogue counterparts. Additionally, a comparative study of their computational complexity will be performed.

5. ACKNOWLEDGEMENTS

This work was partly supported by An Foras Feasa. Additionally, it was partly funded by the Academy of Finland, project no. 122815.

6. REFERENCES

[1] V. Välimäki, "Discrete-Time Synthesis of the Sawtooth Waveform with Reduced Aliasing", IEEE Signal Processing Letters, vol. 12, issue 3, pp. 214-217, 2005.

[2] V. Välimäki, A. Huovilainen, "Oscillator and Filter Algorithms for Virtual Analog Synthesis", Computer Music Journal, vol. 30, issue 2, pp. 19-31, 2006.

[3] V. Välimäki, A. Huovilainen, "Antialiasing oscillators in subtractive synthesis", IEEE Signal Processing Magazine, vol. 24, issue 2, pp. 116–125, 2007.

[4] J. Pekonen and V. Välimäki, "Filter-Based Alias Reduction in Classical Waveform Synthesis", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, NV, pp. 133-136, Apr. 2008.

[5] D. Lowenfels,, "Virtual Analog Synthesis with a Time-Varying Comb", Proceedings of the Audio Engineering Society 115th Convention, New York, USA, 2003.

[6] J. Lane, D. Hoory, E. Martinez, P. Wang, "Modeling Analog Synthesis with DSPs", Computer Music Journal, vol. 21, issue 4, pp. 23-41, 1997.

[7] M. Le Brun, "Digital Waveshaping Synthesis", Journal of the Audio Engineering Society, vol. 27, issue 4, pp. 250-266, 1979.

[8] J. Timoney, V. Lazzarini, T. Lysaght, "A modified FM synthesis approach to bandlimited signal generation", Proceedings of the International Conference on Digital Audio Effects, Espoo, Finland, Sept. 1-4, 2008.

[9] J. Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation", Journal of the Audio Engineering Society, vol. 21, issue 7, pp. 526-34, 1973.

[10] R. Yates and R. Lyons, "DC Blocker Algorithms", IEEE Signal Processing Magazine, vol. 25, no. 3, pp. 132-134, Mar. 2008.

[11] M. Ishibashi, "Electronic Musical Instrument", U.S. Patent no. 4,658,691, 1987.

[12] J. Pekonen, "Coefficient Modulated first-order allpass filter as a distortion effect", Proceedings. 11th Conf. Digital Audio Effects (DAFx), Espoo, Finland, Sept. 1-4, 2008.

[13] K. Steiglitz, A Digital Signal Processing Primer, Prentice Hall, USA, 1996.

[14] J. Timoney, V. Lazzarini, J. Pekonen, V. Valimaki, "Spectrally rich phase distortion sound synthesis using an allpass filter", accepted for publication at Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Tapei, Taiwan, 2009.

[15] T. Stilson, J. Smith, "Alias-free digital synthesis of classic analog waveforms", Proc. Int. Computer Music Conf., Hong Kong, China, pp. 332–335, 1996.