

---

**Victor Lazzarini and Joseph Timoney**

An Grúpa Theicneolaíocht Fuaime agus  
Ceoil Dhigitigh

(Sound and Digital Music Technology  
Group)

National University of Ireland, Maynooth  
Maynooth, Co. Kildare, Ireland

Victor.Lazzarini@nuim.ie

JTimoney@cs.nuim.ie

## New Perspectives on Distortion Synthesis for Virtual Analog Oscillators

The term “virtual analog” (VA) first appeared in the 1990s with the commercial introduction of digital synthesizer instruments that were intended to emulate the earlier analog subtractive synthesizers, such as those produced by, among others, Moog or Sequential Circuits (Smith 2008). In creating a “virtual analog” digital model, two approaches are possible: the first is to build an explicit digital model derived from the various electrical components that form the original analog circuit; the second is to use digital processing structures that will produce outputs that mimic those of the analog system. To date, the second approach has been used for the implementation of the elements of virtual analog subtractive synthesizers, as real-time computational constraints limit the efficacy of circuit models in usable software instruments (Civolani and Fontana 2008). Among the features of a subtractive synthesis system, one aspect that has received more attention than others is algorithms for real-time digital generation of the periodic waveforms associated with voltage-controlled oscillators (VCO), such as sawtooths, square waves, and triangle waves, also known as the classic analog waveforms (Stilson 2006).

Although the original analog waveforms theoretically had an infinite bandwidth, the digital versions must be bandlimited to half the sampling frequency (Stilson 2006). If this is not the case, the audio output can exhibit excessive aliasing distortion, which can severely corrupt the sound quality. This distortion is manifested as audio disturbances that include inharmonicity, beating, and heterodyning. It can be particularly severe when the ratio of the waveform fundamental frequency to the sampling frequency is not an integer, as the aliased components will then fall in between the signal harmonics (Välimäki and

Huovilainen 2007). Furthermore, although a crude technique such as using oversampling with a trivial waveform generator followed by downsampling to the audio rate afterwards can help to ameliorate some of the distortion (Chamberlain 1987), the drawback is that this adds significantly to the computational cost of the implementation. This then impacts on other synthesizer features such as the permissible degree of polyphony. Therefore, for VA subtractive synthesizers, a key goal currently is to develop algorithms that can generate bandlimited digital versions of the classic periodic waveforms at a low computational cost.

To date, a number of algorithms have been proposed for the production of bandlimited digital periodic waveforms. However, even though the variety of approaches appears diverse (Välimäki and Huovilainen 2007), in fact they can be unified in that all can be shown to be connected to nonlinear distortion synthesis (Dodge and Jerse 1985). Perhaps one of the most important discoveries of early digital sound synthesis, distortion-based methods dominated the research landscape of computer music in the 1970s and 1980s. A framework for these techniques was developed through the pioneering work of John Chowning on frequency modulation (FM) synthesis (Chowning 1973); Godfrey Winham, Ken Steiglitz, and Andy Moorer on discrete summation formulae (DSF; Winham and Steiglitz 1970; Moorer 1976, 1977); and Daniel Arfib and Marc LeBrun on digital waveshaping (Arfib 1978; LeBrun 1979). These techniques were demonstrated to stem from the same principles and to have interchangeable interpretations. Perhaps the key aspect that made them so interesting in the early days of computer music was their low computational cost. A number of variants, especially of FM synthesis, were keenly explored (e.g., Schottstaedt 1977; Palamin, Palamin, and Ronveaux 1988; Chowning 1989). In the 1990s, interest in these techniques diminished,

even though some important novel methods were still being proposed, such as that of Puckette (1995). More recently, distortion techniques have been used in new synthesis algorithms (Lazzarini, Timoney, and Lysaght 2008b) and for audio effects (Lazzarini, Timoney, and Lysaght 2007, 2008a, 2008c).

The development of research into VA oscillator models has somewhat rekindled the interest in techniques that can be interpreted from a nonlinear distortion perspective. However, the literature has so far been limited in terms of discussing these new methods from that angle. In the following discussion, we hope to address this issue and propose new nonlinear distortion synthesis algorithms for classic analog-waveform generation. Such signals are normally represented by the sawtooth wave (containing all harmonics with  $1/n$  weights,  $n$  being the harmonic number), the square wave (all odd harmonics,  $1/n$ ), and the triangle (all odd harmonics,  $1/n^2$ ).

## Distortion Synthesis and Analog Waveform Models

We now present a brief survey of key methods for the digital generation of the classic periodic waveforms of subtractive synthesis. Our aim is to show how these fall within the framework of nonlinear distortion synthesis, demonstrating its potential for further designs.

### Lane's Analog Model Oscillator

The technique of nonlinear waveshaping is the basis for one of the early VA models of oscillators (Lane et al. 1997). The algorithm is based on the use of the modulus and absolute-value functions followed by filtering to suppress the aliased portions of the spectrum and correct the spectral rolloff. The waveshaping expression is, with  $\omega = 2\pi ft$ , given by

$$x(t) = f\left(\sin\left(\frac{\omega}{2}\right)\right) = abs\left(\sin\left(\frac{\omega}{2}\right)\right) \quad (1)$$

The spectrum of such a waveshaped signal is not bandlimited, but the most objectionable aliased components in the vicinity of half the sampling rate are reduced by a low-pass filter. Theoretically, the waveshaper output will contain all components

from  $\omega$  upwards, in decreasing strength, plus a certain amount of DC energy (Lane et al. 1997):

$$x(t) = 2 \left[ \frac{1}{\pi} - \frac{2}{\pi} \left( \frac{\cos(\omega)}{1.3} + \frac{\cos(2\omega)}{3.5} + \frac{\cos(3\omega)}{5.7} + \dots \right) \right] \quad (2)$$

By using an adaptive high-pass filter, the steeper rolloff can be corrected to yield an approximate sawtooth spectrum. If we generate a second sawtooth-like wave at  $\omega/2$  and subtract it from the original signal, then square and triangle wave approximations are obtained. This method seems to provide an efficient algorithm for the digital generation of subtractive synthesis oscillator waveforms, but with some aliasing penalty.

### Differentiated Parabolic Wave

Differentiated parabolic wave (DPW) synthesis (Välimäki 2005; Välimäki and Huovilainen 2006, 2007) can be described in terms of nonlinear distortion, as it is based on a parabolic waveshaping of a non-bandlimited sawtooth wave. Similarly to the method in the previous section, the DPW method is also a non-bandlimited but alias-suppressed algorithm. Instead of starting with a sinusoidal signal, the function mapping here uses a complex, already aliased input generated by a modulo counter, which is for all practical purposes a non-bandlimited sawtooth wave. Following on from the waveshaping process, a high-pass filter (a first-order difference) is used to approximate a sawtooth shape. This method seems to provide an efficient algorithm for VA oscillators, but with some aliasing penalty.

### Band-Limited Impulse Train and Discrete Summation Formulae

DSF and band-limited impulse train (BLIT; Stilson and Smith 1996) are similar techniques. In fact, the sinc-based BLIT and the original Winham and Steiglitz DSF pulse are effectively two ways of expressing the same closed-form summation. Both of these methods fall within the nonlinear distortion framework. Moorer's DSF algorithm in particular

Figure 1. Phase-distortion synthesis, original design (solid line). A 50-harmonic bandlimited sawtooth is plotted as comparison (dotted line).

can have an interesting re-casting as a waveshaping process (LeBrun 1979), where one of its four original formulae,

$$s(t) = \sum_{n=1}^{\infty} a^n \cos(n\omega) = \frac{1 - a \cos(\omega)}{1 + a^2 - 2a \cos(\omega)} \quad (3)$$

can be reinterpreted as a cosine wave mapped by a transfer function,

$$f(x) = \frac{1 - ax}{1 + a^2 - 2ax} \quad (4)$$

This fact demonstrates some interesting points of connections between waveshaping and summation-formulae methods. DSF is also strongly connected with FM synthesis, as the latter can be seen as a closed-form summation formula (Dodge and Jerse 1985).

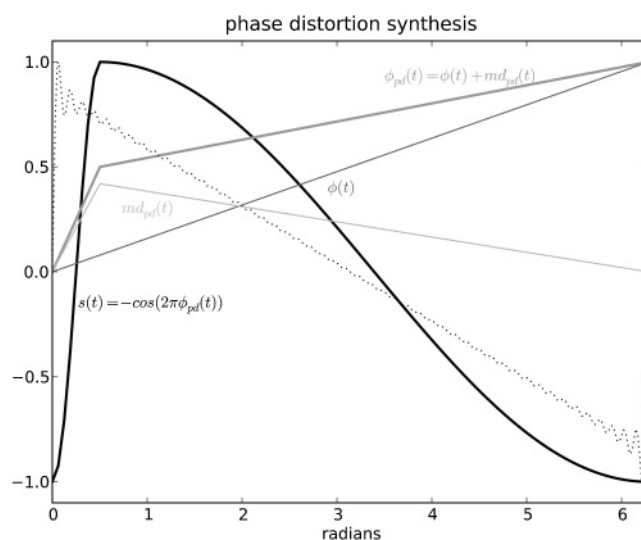
### Phase Distortion or Phaseshaping

The technique of phase distortion (PD), implemented in the Casio CZ series of synthesizers (Massey, Noyes, and Shklair 1987; Roads 1996), has been somewhat ignored in the literature even though it has been shown to be a means of digitally generating the classic periodic waveforms of subtractive synthesis (Ishibashi 1987). Because this method has escaped a more formal treatment, we will try to sketch some of its main points in the following discussion. In addition, we will propose some new alternative approaches to its implementation.

The main technical reference for it is a patent (Ishibashi 1987) where the basic method is described. PD is, effectively, complex frequency or phase modulation in disguise. Using a sinusoidal oscillator, the output signal is produced by using a nonlinear phase increment,

$$s_{pd}(t) = -\cos(\phi_{pd}(t)) \quad (5)$$

where  $\phi_{pd}(t)$  is our distorted phase increment. (Here, we use  $-\cos()$  to keep the original formulation as per Ishibashi (1987) in line with our convention of unipolar  $2\pi$ -modulo phase.) The phase is in fact made up of a linear increment  $\phi(t)$  and a modulation



function  $md_{pd}(t)$ :

$$\phi_{pd}(t) = \phi(t) + md_{pd}(t) \quad (6)$$

We can see in Figure 1 that the function shape is a smooth version of a trivial sawtooth wave, and thus it should have lower energy in its higher frequencies with a consequent reduction in aliasing. One of the problems is that, given the straight angles of the modulating function, we will not be able to produce a strictly bandlimited output, and some aliasing may occur. In practice, by limiting the amount of distortion, it is possible to reduce the foldover to acceptable levels.

Another solution is, of course, to try to produce a roughly bandlimited phase distortion, either by polynomial or Fourier-series approximation of the modulation function. In fact, we can see in Figure 1 that the function shape is very close to that of a bandlimited sawtooth. Because we know how to describe PD synthesis as complex FM (LeBrun 1977), we can use that theory to estimate the output spectrum, which can be made, for all practical purposes, bandlimited. We start with our basic PD expression of Equation 5, but now, for simplicity, we normalize it:

$$s_{pd}(t) = -\cos(2\pi\phi_{pd}(t)) \quad (7)$$

so that

$$\phi_{pd}(t) = \phi(t) + 0.5md_{pd}(t) \text{ with } \phi(t) = f_0 t \quad (8)$$

Now we can define our modulating function to be a raised, scaled, bandlimited (and phase-shifted) sawtooth, described by

$$md_{pd}(t) = 0.5 + \frac{0.5}{MAX(N)} \sum_{n=1}^N \frac{1}{n} \sin\left(2\pi n\phi(t) - \frac{n\pi}{N+1}\right) \quad (9)$$

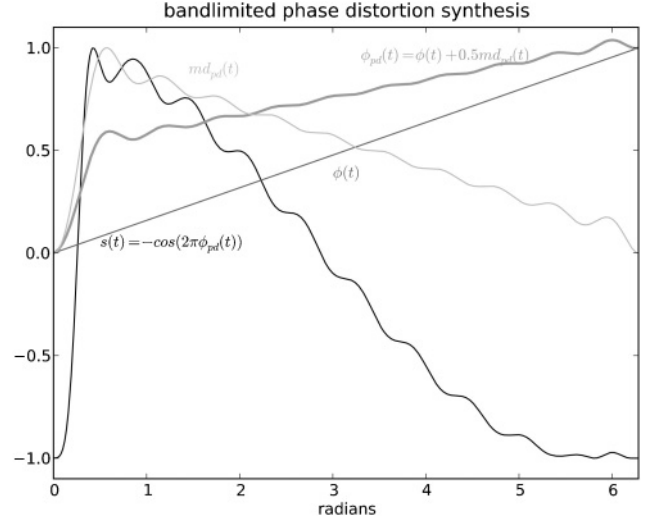
where  $MAX(N)$  is a normalization factor the depends on the number of Fourier components  $N$  used to describe the sawtooth. Now, we can just turn to the theory of FM synthesis (Chowning 1973) to get the correct expansion for our bandlimited PD synthesis (with  $J_n[k]$  standing for the Bessel function of the first kind of order  $n$ ):

$$\begin{aligned} s_{pd}(t) &= -\cos\left(2\pi\left[\phi(t) + 0.25\right.\right. \\ &\quad \left.\left.+ \frac{0.25}{MAX(N)} \sum_{n=1}^N \frac{1}{n} \sin\left(2\pi n\phi(t) - \frac{n\pi}{N+1}\right)\right]\right) \\ &= \sin\left(2\pi\phi(t) + \frac{\pi}{2MAX(N)}\right. \\ &\quad \left.\times \sum_{n=1}^N \frac{1}{n} \sin\left(2\pi n\phi(t) - \frac{n\pi}{N+1}\right)\right) \\ &= \sum_{m_N} \dots \sum_{m_1} \left(\prod_{n=1}^N J_{m_n}\left(\frac{\pi}{2nMAX(N)}\right)\right) \sin\left(2\pi\phi(t)\right. \\ &\quad \left.+ \sum_{n=1}^N m_n \left[2\pi n\phi(t) - \frac{n\pi}{N+1}\right]\right) \quad (10) \end{aligned}$$

A rough measure of bandwidth is given by Carson's rule for FM signals (Van Der Pol 1930; Peiper 2001). The highest significant sideband  $m_N$  will be at  $k+1$ , where  $k = \pi(2nMAX(N))^{-1}$ , with  $k$  rounded to the nearest integer. We can estimate that  $m_1$  and  $m_2$  will be two at most, and the maximum value of all other  $m_n$  will be unity. Given that signal frequencies are a sum of the carrier, which is also the fundamental  $f_0$ , and modulator frequencies  $m_n n f_0$ ,

Figure 2. Bandlimited phase-distortion plots, showing the phase modulation and PD functions  $md_{pd}(t)$  and

$\phi_{pd}(t)$ , as well as the output waveform  $s(t)$ , with the number of harmonics in the modulation waveform  $N = 10$ .



as indicated by Equation 10, the highest significant component of our spectrum will then be at

$$f_N = \left[1 + 1 + 2 + \sum_{n=1}^N n\right] f_0 = \left[4 + \frac{N(N+1)}{2}\right] f_0 \quad (11)$$

It is safe to assume that the highest components will be of very little amplitude, because the value of  $J_1(I)$  in those cases will be very close to 0. So, in fact, the bandwidth might be slightly less than predicted by the expression above. Figures 2 and 3 show the result of this method in the time and frequency domains.

In general, PD can be considered a special case of FM synthesis in which the modulator function fundamental frequency is an integral multiple of the carrier frequency. For the sawtooth approximation, these are the same, whereas for a square-wave simulation, we would want the modulator frequency to be twice that of the carrier. In that case, for a bandlimited approach, we would model the modulator as a triangle wave.

From a complementary angle, if we understand PD to be a method of nonlinear *phaseshaping*, in analogy to waveshaping, we can produce similar-sounding outputs by developing a much simpler algorithm. One way of approximating the original “kinked” phase increment function of Figure 1 can

Figure 3. Spectrum of a bandlimited PD signal, with  $f_0 = 440$  Hz and  $N = 24$  (solid line), in comparison to an ideal (additive synthesis) bandlimited sawtooth wave (dots), with sampling rate  $sr = 44.1$  kHz.

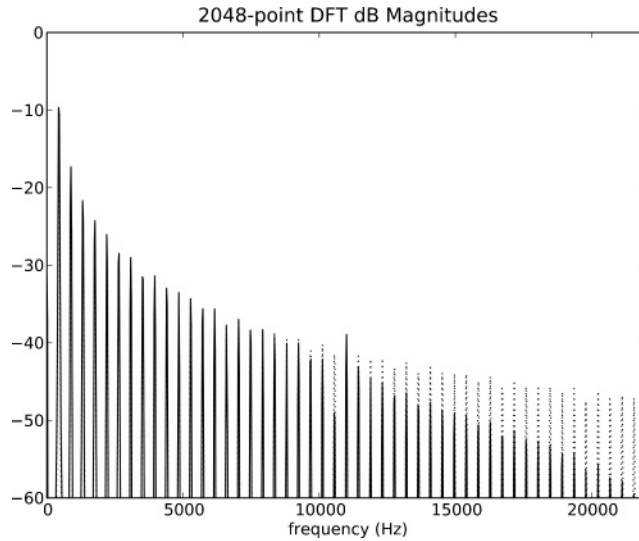
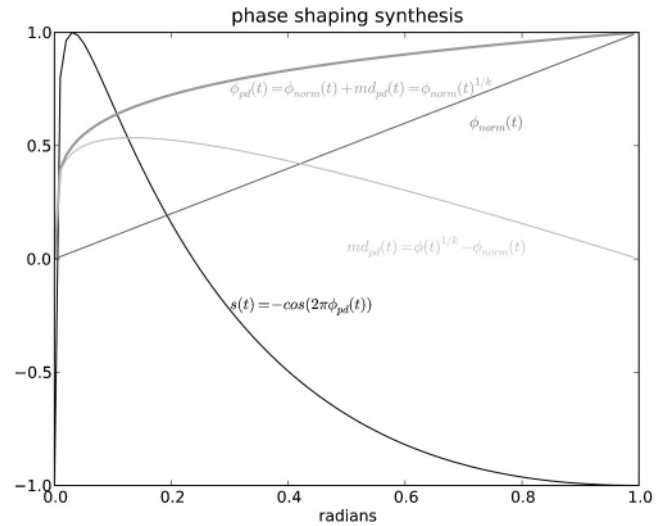


Figure 4. Phase-shaping synthesis,  $k = 5$ .



be simplified by the following expression:

$$s_{pd}(t) = -\cos(2\pi\phi_{norm}(t)^{1/k}) \quad (12)$$

where  $\phi_{norm}(t)$  is the normalized phase increment (moving from 0 to 1), and the phaseshaping transfer function is  $f(x) = x^{1/k}$ . The value of  $k$  is now our phase-distortion index and will increase the bandwidth of the signal. This method is the PD equivalent of polynomial-transfer-function-based waveshaping. The plots for all the relevant modulation functions and the resulting waveform are shown in Figure 4 for  $k = 5$ . However, unlike the previous method, this algorithm is non-bandlimited, but aliasing may be controlled by judicious choice of distortion index. The value of  $k$  can be selected by experimentally observing the maximum significant component at various fundamental frequencies. These can be tabulated and looked up to produce an alias-suppressed output over a range of pitches.

## Novel Approaches

As demonstrated herein, distortion synthesis methods are ubiquitous for the digital generation of the periodic waveforms of subtractive synthesis. The waveshaping technique in particular seems to offer great potential for the development of efficient

algorithms. However, it might have been somewhat overlooked, perhaps due to the shortcomings of the polynomial transfer function approach. The main flaws with this method are twofold: first, the target spectrum is only matched with a specific distortion index; and dynamic spectral evolutions, obtained by varying the distortion index, sound quite unnatural.

There are, however, alternatives to waveshaping based on polynomial transfer functions. In fact, the original approach (Risset 1969) was to heuristically draw a function table that will approach a certain waveshape. It is also possible, as seen in the case of DPW, to use a non-sinusoidal input signal. Both of these approaches, however can often lead to problematic aliasing issues, as the output spectrum will not always be bandlimited, as well as being sometimes hard to predict.

In the methods discussed here, we have chosen to concentrate on sinusoidal inputs, as this will lead to a more easily describable output spectrum. However, this choice means that certain waveshapes, such as the sawtooth, are not directly achievable by any transfer function, as they have asymmetric half-periods. As we will demonstrate later, this is overcome by some small modifications to the basic waveshaping algorithm.

Of the heuristic approaches mentioned above, there is one case of note, which is the use of a

hard-clipping *signum* function defined as

$$\text{sgn}(x) = \begin{cases} +1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \quad (13)$$

This function, when used in waveshaping, produces a non-bandlimited square wave. By smoothing the transition around the origin, it is possible to reduce the amount of aliasing that this transfer function produces. The first of the methods proposed in this section, based on hyperbolic tangent waveshaping, explores the implications of this principle.

In addition, there are alternative approaches for transfer function design, based on functions with infinite polynomial expansions (Taylor's series), such as  $\cos()$ ,  $\sin()$ ,  $\arccos()$ ,  $\arcsin()$ , and  $\exp()$ . They allow for the synthesis of nearly bandlimited spectra, which is a very useful quality for VA algorithms. FM synthesis has also been demonstrated to use such an approach, as it can be put in terms of a waveshaping expression using sinusoidal transfer functions (LeBrun 1979; Lazzarini, Timoney, and Lysaght 2008b). However, the major problem in this case is that the resulting spectra will be weighted by Bessel functions of the first kind. These will result in equally unnatural spectral evolutions, comparable to the results of polynomial waveshaping.

The exponential function, however, is an interesting case. The resulting spectrum in this case is defined in terms of modified Bessel functions of the first kind, which produces a more natural timbral evolution. The second method discussed subsequently takes advantage of this fact to generate bandlimited pulse waveforms as the basis for classic analog waveform synthesis.

### Hyperbolic Tangent Waveshaping

The use of the hyperbolic tangent function,  $\tanh()$ , in waveshaping is quite widespread, especially in non-linear amplification modeling (Huovilainen 2005). This method has not been previously explored for the design of VA oscillator algorithms, however. In this section, we present a method for generating classic analog waveshapes based on this technique. The

main advantage of using sigmoids (i.e., functions exhibiting an "s" shape), in general—and the  $\tanh()$  function in particular—is the fact that their shape approximates the signum shape, with a smoothed transition. The hyperbolic tangent has a partial Taylor's series description, which can be useful for predicting the output spectrum of waveshaping (Zucker 1965):

$$\begin{aligned} \tanh(x) &= x - \frac{x^3}{3} + \frac{2x^5}{15} - \frac{17x^7}{315} + \dots \\ &= \sum_{n=1}^{\infty} \frac{2^{2n}(2^{2n}-1)B_{2n}x^{2n-1}}{(2n)!}, \quad |x| \leq \frac{\pi}{2} \end{aligned} \quad (14)$$

where  $B_i$  represents the Bernoulli number  $i$ , defined as

$$B_{2n} = (-1)^{n+1} \frac{2(2n)!}{(2\pi)^{2n}} \left[ 1 + \sum_{m=0}^{\infty} \frac{1}{m^{2n}} \right] \quad (15)$$

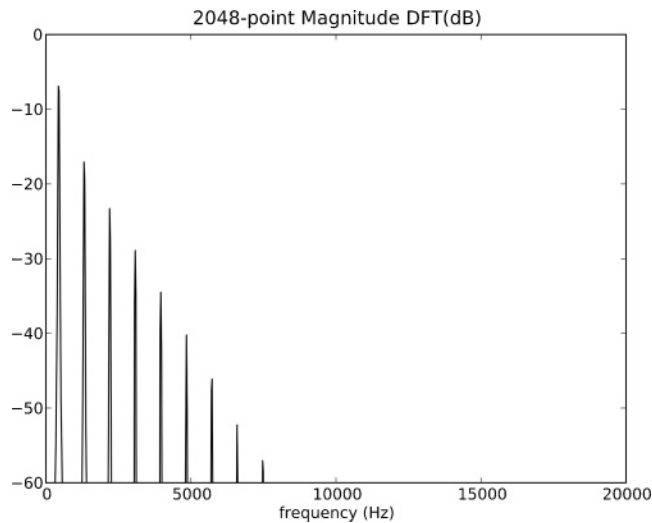
If we derive this function with an input sine wave, we will obtain the following spectrum, which is in practice bandlimited:

$$\begin{aligned} &\tanh\left(\frac{\pi}{2} \sin(\omega)\right) \\ &= \sum_{n=1}^{\infty} \frac{2^{2n}(2^{2n}-1)B_{2n}(\pi/2)^{2n-1}}{(2n)!} \sin^{2n-1}(\omega) = \\ &= \sum_{n=1}^{\infty} \frac{2^{2n}(2^{2n}-1)B_{2n}(\pi/2)^{2n-1}}{(2n)!} \times \\ &\quad \left[ \frac{2}{2^{2n-1}} \sum_{k=0}^{n-1} (-1)^{n-k-1} \binom{2n-1}{k} \sin([2n-2k-1]\omega) \right] = \\ &= \sum_{n=1}^{\infty} \sum_{k=0}^{n-1} (-1)^{n-k-1} \frac{2B_{2n}(2^{2n}-1)(\pi/2)^{2n-1}}{n[k!](2n-k-1)!} \\ &\quad \times \sin([2n-2k-1]\omega) \end{aligned} \quad (16)$$

where  $\omega = 2\pi ft$ .

This will produce a signal with odd harmonics, but with a very steep spectral rolloff (see Figure 5), which is not what we want when we are trying to produce a digital version of an analog square-wave oscillator. However, if we consider that the hyperbolic tangent approximates a signum function

Figure 5. The hyperbolic tangent waveshaping spectrum (as driven by a sine wave with amplitude  $\pi/2$  and frequency 440 Hz).



for high values of  $k$  in

$$\tanh(kx(t)) \approx \text{sgn}(x(t)), \quad k \gg 0 \quad (17)$$

we can then drive the waveshaper with suitable values for the distortion index  $k$ , to obtain a closer model of the square wave. The choice of  $k$  will of course depend on the acceptable levels of aliasing. We have found empirically that for a driving sine wave scaled by  $0.5\pi$ , sampled at 44,100 Hz, we can define  $k$  in terms of the fundamental frequency as

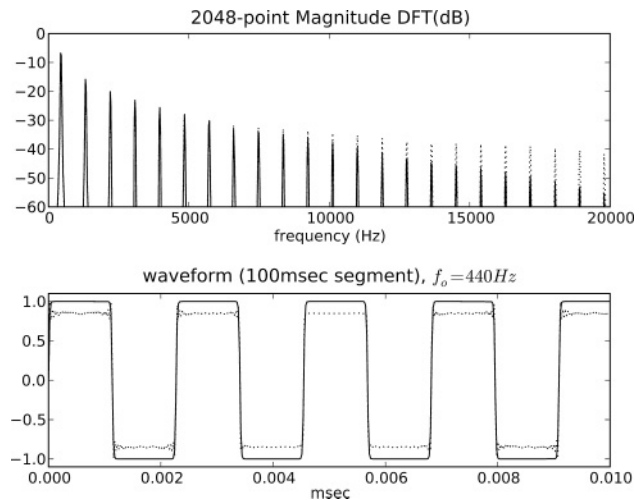
$$k = \frac{12000}{f_o \log_{10} f_o} \quad (18)$$

With this technique, the highest aliasing levels will be kept roughly at  $-60$  dB. In the practical ranges of fundamental frequency values (for instance, up to 5 kHz), aliasing should be tolerable. If oversampling is used, then we can relax this constraint on  $k$  quite significantly. In this case, values of  $k$  producing high-frequency components will not lead to aliasing, as the digital baseband of the signal is higher.

This method is an efficient way to implement a low-aliasing digital version of an analog square-wave oscillator (see Figure 6), as we can use a table lookup waveshaper, driven by a simple oscillator. The computational costs are small; the only extra component required is a second function that can also be tabulated to normalize the output for varying values of  $k$ . This algorithm is capable of

Figure 6. Magnitude spectrum and waveform of hyperbolic tangent waveshaper square wave, with  $f_0 = 440$  Hz. An ideal (additive synthesis)

bandlimited square wave is plotted in the dotted line (spectrum and waveform), with sampling rate  $sr = 44.1$  kHz.



producing dynamic spectra with smooth changes from sinusoidal to square waveshapes.

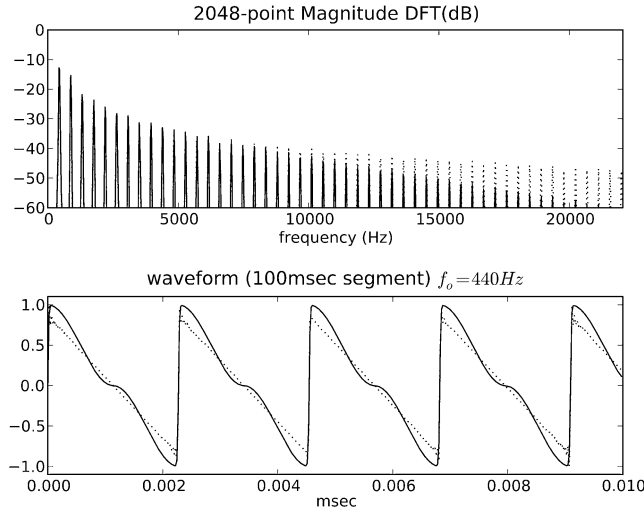
## Sawtooth from Square

In order to produce a nearly bandlimited sawtooth output with the previous method, it is necessary to add some means of heterodyning the waveshaper output. This is the classic combination of nonlinear distortion and ring modulation, which is also an alternative formulation for FM synthesis. Here, in order to produce a sawtooth wave, we will first try and add the missing even harmonics and then combine the odd and even components to generate the output.

Generating even components from odd ones is again another simple procedure. All we need to do is to ring-modulate the square wave with a cosine at the same fundamental frequency. Because we expect the square-wave components to be in the sine phase, we can easily predict the output of this process:

$$\begin{aligned} s(t) &= \cos(\omega) \text{square}(\omega) \\ &= \cos(\omega) \sum_{n=0}^{\infty} \frac{1}{2n+1} \sin([2n+1]\omega) \\ &= \frac{1}{2} \sum_{n=0}^{\infty} \frac{1}{2n+1} (\sin(2n\omega) + \sin(2[n+1]\omega)) \quad (19) \end{aligned}$$

Figure 7. Magnitude spectrum and waveform of hyperbolic tangent waveshaper sawtooth wave, with  $f_0 = 440$  Hz. An ideal (additive synthesis)



$$= \frac{1}{2} \left[ \left(1 + \frac{1}{3}\right) \cos(2\omega) + \left(\frac{1}{3} + \frac{1}{5}\right) \cos(4\omega) + \dots \right]$$

$$= \sum_{n=0}^{\infty} \frac{2n+2}{4n^2+8n+3} \sin(2[n+1]\omega)$$

The resulting signal  $s(t)$  is itself made up of even and odd harmonics of  $2\omega$ , twice the square wave's fundamental frequency, and it is not too far from a sawtooth shape. If we add together the odd and even components, we will have a sawtooth-like wave at  $f_0$ , defined by (excluding a normalizing factor of 0.5):

$$saw(t) = \text{square}(\omega)(\cos(\omega) + 1)$$

$$= \sum_{n=0}^{\infty} \frac{1}{2n+1} \sin([2n+1]\omega)$$

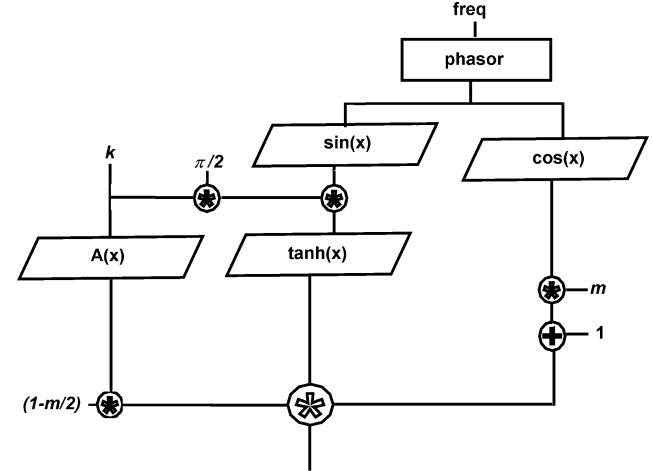
$$+ \frac{2n+2}{4n^2+8n+3} \sin(2[n+2]\omega) \quad (20)$$

This models the sawtooth shape relatively well (see Figure 7). Remembering that in our notation the expected amplitudes of the sawtooth's even harmonics are  $(2n+2)^{-1}$ , the approximation error can be calculated by

$$err_{\text{even\_harmonics}}(n) = 20 \log_{10} \frac{\left(\frac{2n+2}{4n^2+8n+3}\right)}{\left(\frac{1}{2n+2}\right)} \quad (21)$$

bandlimited sawtooth wave is plotted in the dotted line (spectrum and waveform), with sampling rate  $sr = 44.1$  kHz.

Figure 8. Hyperbolic tangent waveshaper oscillator flowchart.



The only significant difference is at the second harmonic, where the error is 2.5 dB. From the fourth harmonic upward, the error is less than 0.5 dB. This is a general method that can be used with any bandlimited square-wave input. We can therefore insert our hyperbolic-tangent waveshaper square into it to produce a sawtooth. In addition, we can also define a control  $m$ ,  $0 \leq m \leq 1$ , that will affect the blend of even and odd harmonics. This, together with the waveshaping modulation index, can be used to model the shape control customarily found in analog oscillators (Moog 2002).

The complete expression for the algorithm becomes

$$s(t) = A(k) \left(1 - \frac{m}{2}\right) \tanh\left(\frac{\pi k \sin(\omega)}{2}\right) [1 + m \cos(\omega)] \quad (22)$$

where  $A(k)$  is a scaling function used to normalize the signal for different values of  $k$ . Smooth changes in the shape of the wave can be achieved by varying the value of  $m$  from 0 to 1 in Equation 22. As  $m$  approaches 0, the expression becomes closer to Equation 17, thus producing only odd harmonics. The larger the value of  $m$  (within its correct range), the more prominent even harmonics become. The signal flowchart for this instrument is shown in Figure 8.



## Modified FM Synthesis

Modified FM (ModFM; Timoney, Lazzarini, and Lysaght 2008) is a technique derived from classic FM synthesis. The main difference between the two techniques is that its expansion is based on modified Bessel functions, rather than the ordinary Bessels found in the expansion of FM equations. The relationship between the two techniques is better explained first by manipulating the simple FM formulation (expressed in terms of cosines rather than sines):

$$\begin{aligned} s_{FM}(t) &= \cos(\omega_c + k\cos(\omega_m)) \\ &= \cos(k\cos(\omega_m))\cos(\omega_c) - \sin(k\cos(\omega_m))\sin(\omega_c) \\ &= J_0(k)\cos(\omega_c) + \sum_{n=1}^{\infty} (-1)^{\text{int}(\frac{n}{2})} J_n(k) \\ &\quad \times (\cos(\omega_c - n\omega_m) + (-1)^n \cos(\omega_c + n\omega_m)) \quad (23) \end{aligned}$$

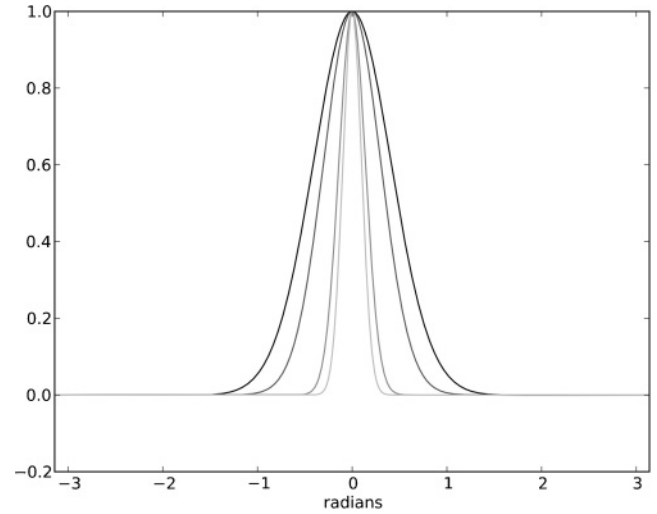
where  $J_n[k]$  is the Bessel function of order  $n$ , and  $\text{int}(n)$  is the integer part of the number  $n$ . Bearing in mind the expression above, ModFM is then expressed as

$$\begin{aligned} s_{ModFM}(t) &= e^{k\cos(\omega_m)} \cos(\omega_c) \\ &= I_0(k)\cos(\omega_c) + \sum_{n=1}^{\infty} I_n(k) (\cos(\omega_c - n\omega_m) \\ &\quad + \cos(\omega_c + n\omega_m)) \quad (24) \end{aligned}$$

with  $I_n[k] = i^{-n}J_n(ik)$ , the modified Bessel function of order  $n$ , which is a special case of that function for purely imaginary arguments (Watson 1944).

If FM synthesis can be seen as a combination of sinusoids ring-modulated by sinusoidal-waveshaper signals, ModFM is then based on a sinusoid ring-modulated by an exponential-waveshaper signal. Other points of connection between the two expressions above are discussed in Moorer (1976) and Palamin, Palamin, and Ronveaux (1988), where variations on a similar algorithm are explored. The main advantage of ModFM when applied to the production of the waveforms of subtractive synthesis is that, unlike FM, ModFM exhibits a smoother and more natural-sounding spectral evolution for

Figure 9. ModFM pulse waveforms for  $k = 5, 10, 50$ , and  $100$ . The wider pulses are generated with lower values of  $k$  (the modulation index).



time-varying modulation index values. This is in contrast to the Bessel functions that describe the output of classic FM synthesis, which exhibit an oscillatory amplitude pattern as the modulation index increases (Chowning 1973).

If we scale the basic ModFM signal output by

$$g(k) = e^{-k} \quad (25)$$

then the ever-increasing modified Bessel functions are a very good choice of scaling coefficients, producing spectra that are, for all practical purposes, bandlimited. For high values of  $k$  and  $c:m = 1$ , the algorithm will produce a pulse train. The equation for the ModFM-based pulse can be written as

$$s_{pulse}(t) = e^{[k\cos(\omega) - k]} \cos(\omega) \quad (26)$$

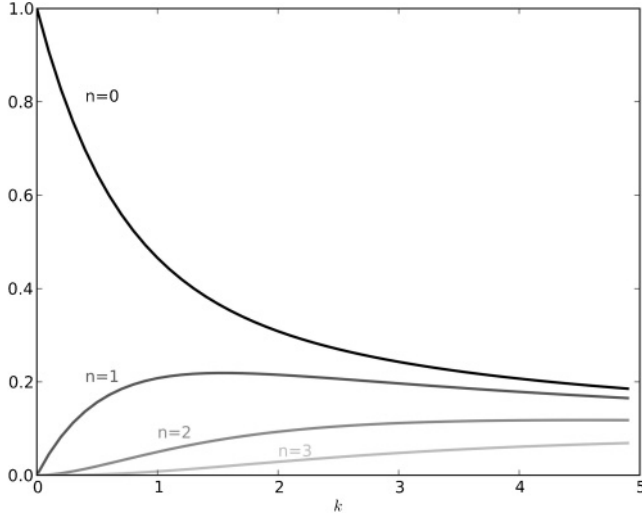
The width, and therefore smoothness, of the pulse is determined by the modulating index  $k$ , and lower values of  $k$  will give a broader pulse shape. This is illustrated in Figure 9 for various modulation index values. The pulse formed with the lower modulation index is wider.

Taking the ModFM expansion (Equation 24), the spectrum of the pulse train defined herein is given by

$$s_{pulse}(t) = \frac{2}{e^k} \sum_{n=1}^{\infty} I'_n(k) \cos(n\omega) \quad (27)$$

where  $I'_n(k) = 0.5[I_{n-1}(k) + I_{n+1}(k)]$  (Watson 1944).

Figure 10. Plot of the scaling functions  $2e^{-k}I_n(k)$  for  $n = 0$  to 3, found in the ModFM expression.



The spectrum of the modFM pulse has a low-pass characteristic. This roll-off pattern suggests that it should be possible to carefully choose parameter values for the modulation index  $k$  in the equation above such that the pulse train waveform is effectively a bandlimited signal.

The rate of rolloff in the spectrum is determined by the modulation index  $k$  and the scaled modified Bessel functions of different orders. A plot of the amplitudes of the scaling functions in the ModFM pulse expansion (Equation 27) is shown in Figure 10. Notice that these curves are free of the characteristic wobble of classic FM Bessel functions.

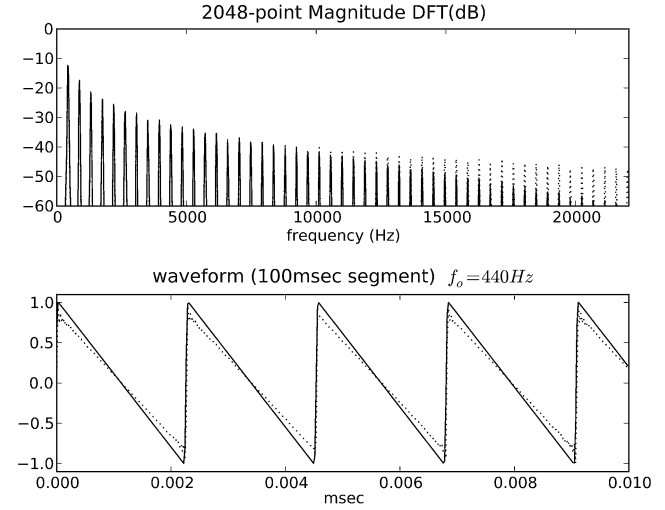
### Bandlimited Sawtooth Generation

If a bandlimited pulse train is available, it is possible to generate a bandlimited sawtooth wave by integration, following the procedure given in Stilson (2006, p. 214). The integration can be carried out using a one-pole filter whose z-transform is

$$H(z) = \frac{1}{1 - z^{-1}} \quad (28)$$

An important factor in the procedure is to remove the mean of the pulse train so that the sawtooth will be centered around a DC level of 0. For this effect in real-time implementations, we chose to use the

Figure 11. Magnitude spectrum and waveform of ModFM sawtooth wave (solid line), with  $f_0 = 440$  Hz, compared to an ideal bandlimited sawtooth wave (dotted line), with sampling rate  $sr = 44.1$  kHz.



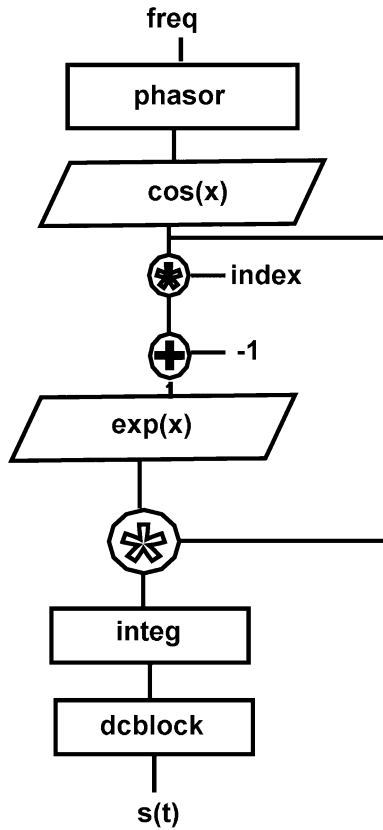
linear-phase DC blocking filter given in Yates and Lyons (2008) after the integration stage. A plot of the resulting signal in the time and frequency domain is shown in Figure 11. The bandlimited-sawtooth-generation system could thus be described by the block diagram in Figure 12.

We must determine the best value for the modulation index  $k$  such that we achieve both our goals of bandlimited output and sawtooth approximation. We demonstrated in the previous section that the higher the modulation index  $k$ , the greater the magnitude of the higher-frequency spectral harmonics. However, from Equation 27, our output signal is theoretically not bandlimited, so high values of  $k$  could introduce aliasing. It is clear a trade-off must be found between a sufficiently bright spectrum that closely approximates a sawtooth and one free of perceptible aliasing. By integrating Equation 27 with respect to frequency, we will find an expression for our ModFM sawtooth wave:

$$s_{saw}(t) = \frac{2}{e^k(n\omega)} \sum_{n=1}^{\infty} I'_n(k) \sin(n\omega) \quad (29)$$

We can now find a maximum value for the modulation index at any sawtooth frequency such that the harmonics that lie beyond half the sampling frequency are less than a defined threshold. In addition, due to the low-pass characteristic of the sawtooth wave, we only need to know the magnitude

Figure 12. The ModFM sawtooth oscillator flowchart.



of the first harmonic that appears above half the sampling frequency. Using a  $-90$  dB threshold (with respect to the fundamental frequency  $f_0$ ), we can find the best values for  $k$  using the following expression:

$$\max_k \left\{ 20 \log_{10} \frac{I'_{n+1}(k)(n+1)^{-1}}{I'_1(k)} \right\} \leq -90 \text{ dB}, \quad n = \left\lfloor \frac{sr}{2f_0} \right\rfloor \quad (30)$$

where  $sr$  is the sampling rate in Hz.

Using Equation 30 to compute the maximum value of modulation index, it was found that its value decreases exponentially with respect to frequency, or linearly with respect to the equivalent MIDI note number. A first-order polynomial was fitted to the line, which then provides the expression for the maximum modulation index

$$k = e^{-0.1513N+15.927} \quad (31)$$

where  $N$  is the MIDI note value of the desired pitch of the bandlimited sawtooth wave to be generated, defined as

$$N = 12 \log_2 (f/440) + 69 \quad (32)$$

with  $f$  in units of Hz. The modulation index is high for low fundamentals, but smaller than unity at the other extreme of the range.

### Generating Other Bandlimited Waveforms

Other waveforms can be produced following a similar method, but using a bipolar pulse as the starting point. This signal can easily be generated by using a ModFM  $c:m$  ratio of 1:2:

$$S_{bipulse}(t) = e^{[k \cos(2\omega) - k]} \cos(\omega) \quad (33)$$

The resulting bipolar bandlimited pulse wave is shown in Figure 13. Integrating this waveform using the first-order filter defined in Equation 28 then produces a bandlimited square wave. The resulting square wave is shown in Figure 14. As the method is basically the same as for the sawtooth wave, changes to the oscillator waveshape are controlled by simple  $c:m$  ratio selection. The only limitation is that continuous smooth changes are not possible. This is because values for  $c:m$  that do not closely approximate ratios of small integers will result in inharmonic spectra.

In addition, if we start from a square wave, we can also produce a sawtooth wave using the method outlined previously in Equations 19 and 20. One advantage of such an algorithm is that the DC-blocking requirement is effectively removed, as the bipolar pulse in general has an insignificant DC component. In this case, the transition between sawtooth and square can be made simpler and linear.

Finally, by further integrating the square wave, we will also be able to produce a triangle waveform. In this case, we will again have to take care of removing the signal mean, using the same DC-blocking filter of the previous section. Other ad hoc shapes can also be produced by the choice of various  $c:m$  ratios.

Figure 13. ModFM bipolar pulse, with  $k = 5, 10, 50,$  and  $100$ .

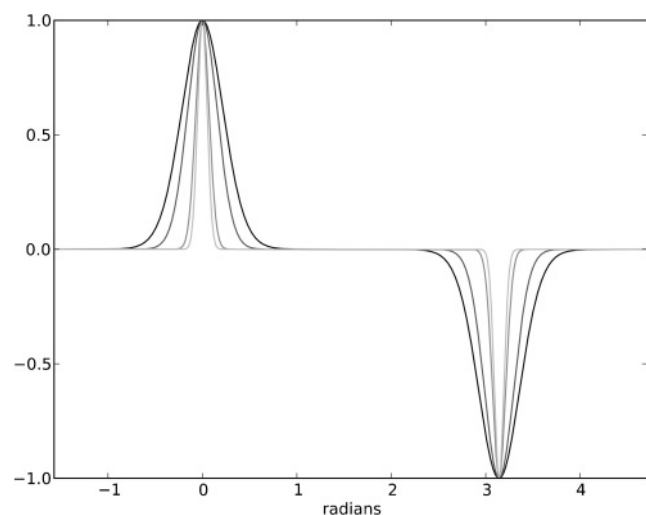


Figure 13

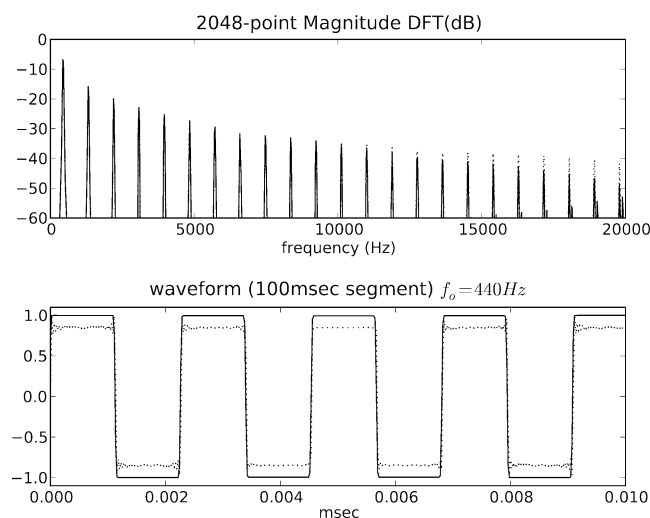


Figure 14

## Conclusion

The framework of distortion synthesis offers some very useful methods for the implementation of bandlimited and low-aliasing VA oscillators. We have demonstrated here a number of techniques that, in addition to established algorithms, provide some interesting alternatives for the synthesis of classic analog waveforms. Particularly, these methods are

Figure 14. Magnitude spectrum and waveform of ModFM sawtooth wave with  $f_0 = 440$  Hz, compared to an ideal bandlimited square wave (dotted line), with

sampling rate  $sr = 44.1$  kHz. Note the small amount of aliasing in the ModFM spectrum, which is nevertheless not objectionable.

possibly low-cost in computational terms. Another advantage of the distortion techniques discussed here is that they allow for tonal control without the need for separate filtering. This makes possible the addition of oscillator shape controls similar to the ones found in analog synthesizers. We are confident the novel methods introduced in this article are an important addition to the arsenal of techniques at the disposal of digital instrument designers.

A number of Python scripts, sound examples, and other resources are available online at <http://music.nuim.ie/synthesis> and will also appear on the 2010 *Computer Music Journal* DVD.

## Acknowledgment

We would like to acknowledge the support of An Foras Feasa, who partially funded the research leading to this article.

## References

- Arfib, D. 1978. "Digital Synthesis of Complex Spectra by Means of Multiplication of Nonlinear Distorted Sinewaves." *AES Preprint No. 1319 (C2)*. New York: Audio Engineering Society.
- Chamberlain, H. 1987. *Musical Applications of Microprocessors*. Indianapolis, Indiana: Hayden.
- Chowning, J. 1973. "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation." *Journal of the Audio Engineering Society* (21):526–534.
- Chowning, J. 1989. "Frequency Modulation Synthesis of the Singing Voice." In M. Mathews and J. R. Pierce, eds. *Current Directions in Computer Music Research*. Cambridge, Massachusetts: MIT Press, pp. 57–63.
- Civolani, M., and F. Fontana. 2008. "A Nonlinear Digital Model of the EMS VCS3 Voltage-Controlled Filter." *Proceedings of the 11th International Conference on Digital Audio Effects*. Espoo, Finland: pp. 35–42.
- Dodge, C., and T. Jerse. 1985. *Computer Music*. New York: Schirmer.
- Huovilainen, A. 2005. "Nonlinear Digital Implementation of the Moog Ladder Filter." *Proceedings of the 8th International Conference on Digital Audio Effects*, Naples, Italy: University of Naples, pp. 61–65.
- Ishibashi, M. 1987. "Electronic Musical Instrument." U.S. Patent No. 4,658,691.
- Lane, J., et al. 1997. "Modeling Analog Synthesis with DSPs." *Computer Music Journal* 21(4):23–41.

- Lazzarini, V., J. Timoney, and T. Lysaght. 2007. "Adaptive FM Synthesis." *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx07)*. Bordeaux: University of Bordeaux, pp. 21–26.
- Lazzarini, V., J. Timoney, and T. Lysaght. 2008a. "The Generation of Natural-Synthetic Spectra by Means of Adaptive Frequency Modulation." *Computer Music Journal* 32(2):12–22.
- Lazzarini, V., J. Timoney, and T. Lysaght. 2008b. "Split-Sideband Synthesis." *Proceedings of the 2008 International Computer Music Conference*. San Francisco, California: International Computer Music Association, pp. 41–44.
- Lazzarini, V., J. Timoney, and T. Lysaght. 2008c. "Asymmetric-Spectra Methods for Adaptive FM Synthesis." *Proceedings of the International Conference on Digital Audio Effects*. Available online at [www.acoustics.hut.fi/dafx08/papers/dafx08\\_42.pdf](http://www.acoustics.hut.fi/dafx08/papers/dafx08_42.pdf).
- Le Brun, M. 1977. "A Derivation of the Spectrum of FM with a Complex Modulating Wave." *Computer Music Journal* 1(4):51–52.
- Le Brun, M. 1979. "Digital Waveshaping Synthesis." *Journal of the Audio Engineering Society* 27(4):250–266.
- Massey, H., A. Noyes, and D. Shklair. 1987. *A Synthesist's Guide to Acoustic Instruments*. New York: Amsco.
- Moog, R. 2002. *Minimoog Voyager User's Manual*. Asheville, North Carolina: Moog Music.
- Moorer, J. A. 1976. "The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulas." *Journal of the Audio Engineering Society* 24(9):717–727.
- Moorer, J. A. 1977. "Signal Processing Aspects of Computer Music: A Survey." *Proceedings of the IEEE* 65(8):1108–1141.
- Palamin, J. P., P. Palamin, and A. Ronveaux. 1988. "A Method of Generating and Controlling Musical Asymmetric Spectra." *Journal of the Audio Engineering Society* 36(9):671–685.
- Peiper, R. 2001. "Laboratory and Computer Tests for Carson's FM Bandwidth Rule." *IEEE 33rd Southeastern Symposium on System Theory*. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, pp. 145–150.
- Puckette, M. 1995. "Formant-Based Audio Synthesis Using Nonlinear Distortion." *Journal of the Audio Engineering Society* 43(1):40–47.
- Roads, C. 1996. *Computer Music Tutorial*. Cambridge, Massachusetts: MIT Press.
- Risset, J. C. 1969. "Introductory Catalogue of Computer-Synthesized Sounds." Reprinted in *Computer Music Currents* 13 (1995). Mainz: Schott Wergo Music Media, 109–254 (CD booklet).
- Schottstaedt, W. 1977. "The Simulation of Natural Instrument Tones using a Complex Modulating Wave." *Computer Music Journal* 1(4):46–50.
- Smith, J. O. 2008. "Physical Audio Signal Processing for Virtual Musical Instruments and Audio Effects." Available online at [ccrma.stanford.edu/~jos/pasp](http://ccrma.stanford.edu/~jos/pasp).
- Stilson, T. 2006. "Efficiently-Variable Non-Oversampled Algorithms in Virtual-Analog Music Synthesis." PhD dissertation, Department of Electrical Engineering, Stanford University. Available online at [ccrma.stanford.edu/~stilti/papers/TimStilsonPhDThesis2006.pdf](http://ccrma.stanford.edu/~stilti/papers/TimStilsonPhDThesis2006.pdf).
- Stilson, T., and J. Smith. 1996. "Alias-Free Digital Synthesis of Classic Analog Waveforms." *Proceedings of the 1996 International Computer Music Conference*. San Francisco: International Computer Music Association, pp. 332–335.
- Timoney, J., V. Lazzarini, and T. Lysaght. 2008. "A Modified FM Synthesis Approach to Bandlimited Signal Generation." *Proceedings of the 11th International Conference on Digital Audio Effects*. Espoo, Finland, pp. 27–33.
- Välimäki, V. 2005. "Discrete-Time Synthesis of the Sawtooth Waveform with Reduced Aliasing." *IEEE Signal Processing Letters* 12(3):214–217.
- Välimäki, V., and A. Huovilainen. 2006. "Oscillator and Filter Algorithms for Virtual Analog Synthesis." *Computer Music Journal* 30(2):19–31.
- Välimäki, V., and A. Huovilainen. 2007. "Antialiasing Oscillators in Subtractive Synthesis." *IEEE Signal Processing Magazine* 24(2):116–125.
- Van Der Pol, B. 1930. "Frequency Modulation." *Proceedings of the Institute of Radio Engineers* 18(7):1194–1205.
- Watson, G. N. 1944. *A Treatise on the Theory of Bessel Functions*, 2nd ed. Cambridge: Cambridge University Press.
- Winham, G., and K. Steiglitz. 1970. "Input Generators for Digital Sound Synthesis." *Journal of the Acoustic Society of America* 47(2):665–666.
- Yates, R., and R. Lyons. 2008. "DSP Tips & Tricks [DC Blocker Algorithms]." *IEEE Signal Processing Magazine* 25(2):132–134.
- Zucker, R. 1965. "Elementary Transcendental Functions: Logarithmic, Exponential, Circular and Hyperbolic Functions." In M. Abramowitz and I. Stegun, eds. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, pp. 65–94.