Oscillator and Filter Algorithms for Virtual Analog Synthesis

Author(s): Vesa Välimäki and Antti Huovilainen

Source: *Computer Music Journal*, Summer, 2006, Vol. 30, No. 2 (Summer, 2006), pp. 19-31

Published by: The MIT Press

Stable URL: https://www.jstor.org/stable/3682001

Vesa Välimäki and Antti Huovilainen
Helsinki University of Technology
Laboratory of Acoustics and Audio Signal Processing
P.O. Box 3000, FI-02015 TKK, Espoo, Finland
www.acoustics.hut.fi/
vesa.valimaki@tkk.fi
antti.huovilainen@acoustics.hut.fi

Oscillator and Filter Algorithms for Virtual Analog Synthesis†

Virtual analog synthesis refers to computational methods that imitate the sound production principles used in electronic music synthesizers of the 1960s and 1970s. In practice, it means digital subtractive synthesis. In this paper, we introduce new methods to generate digital versions of classical analog waveforms with reduced aliasing. We also propose modifications to the digital nonlinear model of the Moog ladder filter. These virtual analog synthesis techniques enable the production of retro sounds with modern computers.

Introduction

Virtual analog synthesis refers to computational simulation of the sound generation principles of analog synthesizers of the 1960s and 1970s. In practice, it means digital subtractive synthesis. The basic principle in subtractive synthesis is, first, to generate a signal with rich spectral content, and then to filter that signal with a time-varying resonant filter.

Virtual analog synthesis became a popular commercial term around 1995, when Clavia introduced the Nord Lead 1 synthesizer, which was marketed as an analog-sounding digital synthesizer that uses no samples (Clavia 2002). Instead, all sounds were generated by simulating analog subtractive synthesis. Previously, the Roland D-50 synthesizer of the late 1980s worked in a similar way, although it contained sampled sounds. An early example of an attempt to design a digital synthesizer that sounds analog was Synergy (Kaplan 1981).

Design and implementation of digital subtractive synthesis are more demanding than is generally understood, because imitating analog electronics with digital processing is not as easy as it may seem. One problem is aliasing caused by sampling of analog waveforms that have rapid changes. The spectra of such waveforms contain infinitely high frequencies, and the signals are thus not band-limited. Another difficulty is that analog filters do not obey simple linear theory. With high signal levels they generate distortion. This does not naturally occur in digital processing, but it must be designed and implemented on purpose (Rossum 1992; Huovilainen 2004).

In this paper, we discuss new versions of oscillator and resonant filtering algorithms that can sound like old analog synthesizers. Computationally very efficient oscillator algorithms not requiring wavetables and having reduced aliasing distortion are proposed for classical waveforms used in subtractive synthesis. These algorithms are modifications and extensions of the digital sawtooth waveform algorithm based on the differentiated parabolic wave (DPW) proposed recently by Välimäki (2005).

A new digital resonant filter structure is also proposed for subtractive synthesis. It is a modified version of the nonlinear digital Moog ladder filter introduced previously by Huovilainen (2004). The new structure reduces the computational cost of the nonlinear digital Moog filter by using a single nonlinearity instead of five nonlinear functions inside filter sections. The new digital Moog filter structure also decouples fairly well the cutoff and the resonance parameters and offers several response types by selecting a weighted sum of different output points.

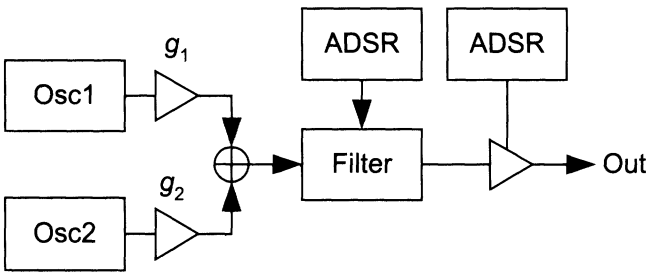
Analog Subtractive Synthesis

The electronic music modules introduced by Robert A. Moog in the mid-1960s are one of the most important innovations in music technology (Moog 1965). A few years later, his company introduced products where the various modules, such as oscillators, filters, and amplifiers, were integrated into a

†This paper is a revised and extended version of the paper "New approaches to digital subtractive synthesis" that was published at the 2005 International Computer Music Conference, Barcelona, Spain, September 2005.

Computer Music Journal, 30:2, pp. 19–31, Summer 2006

Figure 1. A typical block diagram of subtractive synthesis as it was implemented in the Prophet 5 synthesizer in the late 1970s.



single portable unit. Subtractive synthesis was the main principle used in these instruments. The Minimoog was one of the most popular analog synthesizers in the 1970s.

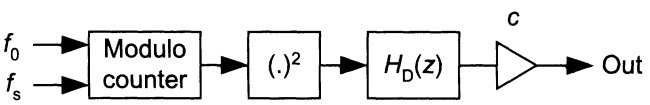
The Prophet 5 synthesizer introduced by Sequential Circuits in 1979 has microprocessor controlled electronics, but it is still an analog synthesizer. Its block diagram, shown in Figure 1, is today a classic example of the subtractive synthesis principle. It includes two oscillators, a resonant low-pass filter, and two envelope generators (ADSR in Figure 1 stands for attack time, decay time, sustain level, and release time). There are a couple of alternative waveforms available together with a noise source.

Digital Oscillators

The sharp corners of geometric waveforms, such as the sawtooth or the square wave, cause aliasing because such signals are not band-limited. Three different classes of methods are known to avoid this problem:

1. Band-limited methods that generate harmonics only below the Nyquist limit, such as additive synthesis (Moore 1990, pp. 270–271; Chaudhary 1998) and its variants, e.g., wavetable synthesis (Burk 2004) and the discrete summation formulae (Winham and Steiglitz 1970; Moorer 1976; Moore 1990, pp. 271–273; Lazzaro and Wawrzynek 2004);
2. Quasi-band-limited methods in which aliasing is low and its level can be adjusted by design to save computational costs, such as in the BLIT (Stilson and Smith 1996a) and the MinBLEP (Brandt 2001) techniques;

Figure 2. The block diagram of the DPW algorithm (Välimäki 2005).



3. Alias-suppressing methods in which it is accepted that some aliasing occurs, but an attempt is made to attenuate it sufficiently.

In this work, we focus on the third category of methods. Three approaches are currently available: oversampling of trivial waveforms (Chamberlin 1985, pp. 423–424), the distortion and filtering of sine waves (Lane et al. 1997), and the differentiated parabolic waveform (DPW) (Välimäki 2005). We discuss and extend the latter method in the following.

DPW Algorithm and the Sawtooth Waveform

The sawtooth waveform is a common source signal in subtractive synthesis. Välimäki (2005) found out that a digital signal that closely resembles the sawtooth wave—but with attenuated aliasing distortion—can be produced by differentiating a piecewise parabolic waveform. The simplest version of the DPW algorithm generates the sawtooth waveform in four stages, as illustrated in Figure 2. First the trivial sawtooth waveform is generated using a bipolar modulo counter, then the waveform is raised to the second power, the signal is differentiated with a first difference filter with transfer function $H_D(z) = 1 - z^{-1}$, and, finally, the obtained waveform is scaled by factor

$$c = f_s/[4f_0(1 - f_0/f_s)], \quad (1)$$

where f_0 is the fundamental frequency of the sawtooth signal and f_s is the sampling rate. This scaling factor may well be replaced with $c = f_s/4f_0$, which gives a slightly too small gain for high fundamental frequencies, but is accurate within 1 dB up to 4 kHz, when the sampling rate of 44.1 kHz is used.

Figure 3 shows how the basic DPW algorithm can be programmed using Matlab. Note that the bipolar modulo counter is slightly different from a phase accumulator used in conventional wavetable synthesis, because counting starts from –1 instead of 0.

Figure 3. Matlab code showing the implementation of the DPW sawtooth algorithm. The variable delta must be initialized to the ratio of the fundamental frequency to the sam-

pling frequency, f_0/f_s. The variable c is the scaling coefficient explained in text. The comment after the percent sign explains the purpose of each line of code.

Figure 4. Waveform and spectrum of (a, b) the trivial sawtooth waveform, (c, d) the squared sawtooth wave, and (e, f) the differentiated parabolic waveform. The fundamental frequency is 2793.8 Hz (MIDI note number 101), and the sampling rate is 44.1 kHz. In (b), (d), and (f), the desired spectral components (2793.8 Hz, 5587.6 Hz, 8381 Hz, . . .) are circled (o), while the rest of the spectral components are caused by aliasing and are heard as disturbance. These and other spectra presented in this paper have been computed from a 1 s signal segment with a 65536-point FFT using a Chebyshev window function that has a 120-dB side-lobe attenuation.

```
phase = mod(phase + delta,1);  % Modulo counter (0 ... +1)
bphase = 2 * phase - 1;        % Bipolar modulo counter (-1...+1)
sq = bphase^2;                 % Piecewise parabolic waveform
dsq = sq - z1;                 % Differentiate the above signal
z1 = sq;                       % Update state variable
out = c * dsq;                 % Scale the improved sawtooth wave
```

Figure 3

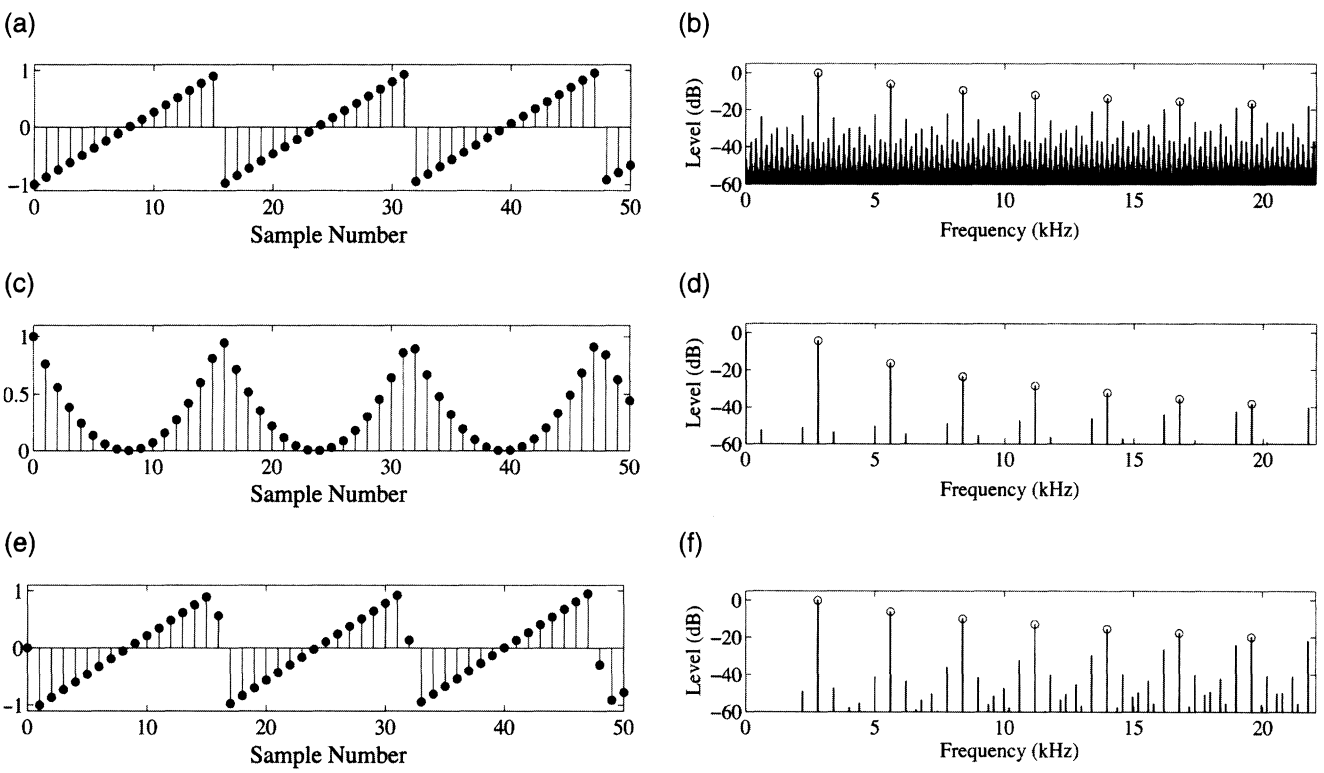


Figure 4

The waveform produced by the bipolar modulo counter resembles the sawtooth waveform, as seen in Figure 4a, but it sounds badly distorted. The reason is that its spectrum decays slowly, about 6 dB per octave. When it is sampled, the spectral components above the Nyquist limit are mirrored down to the audible frequencies. This is clearly seen in Figure 4b, where the desired harmonics are indicated by circles and the rest of the peaks are aliased images. This signal is called the trivial sawtooth wave.

Raising the signal to the second power modifies the waveform so that it now consists of parabola segments, which form the unipolar, non-negative signal shown in Figure 4c. The spectrum of this waveform decays at about 12 dB per octave, and this is why aliasing is reduced in Figure 4d (Välimäki 2005). Finally, when the piecewise parabolic signal is differentiated and scaled, the signal again looks like the sawtooth waveform, see Figure 4e, but the aliased components are suppressed, as seen in Figure 4f.

Figure 5. Magnitude responses of (a) the ideal differentiator (dashed line), the first-order FIR differentiator (solid line), the two-point average filter (dash-dot line), (b) the ideal differentiator (dashed line), and the averaged differentiator (solid line).

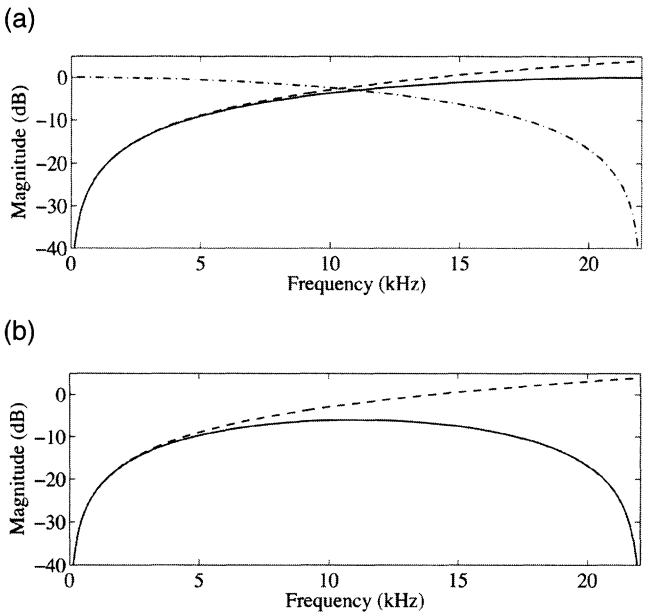
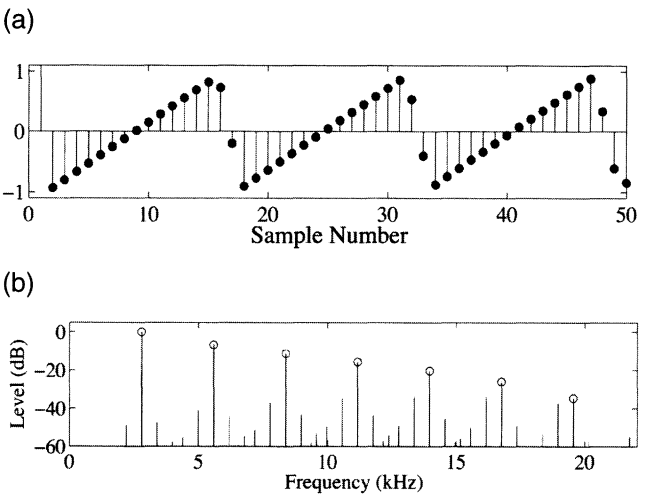


Figure 6. (a) The waveform and (b) the spectrum of the signal obtained with the averaged differentiator.



A remaining problem is that at high frequencies the level of aliased components is close to that of the harmonics. This may lead in some cases to beating. A solution to alleviate this is to replace the differentiator with its averaged version $H_D(z) = (1 - z^{-2})/2 = (1 - z^{-1})(1 + z^{-1})/2$, which does not spoil the simplicity of the algorithm. This filter is a combination of a two-point average filter and the first-difference filter. Figure 5a compares the magnitude responses of the first-difference filter $(1 - z^{-1})/2$ and the ideal differentiator, and also shows the magnitude response of the two-point average filter. It can be seen that the response of the simple differentiator is very close to the ideal at low frequencies. The error exceeds 1 dB above 11 kHz. Figure 5b gives the magnitude responses of the ideal differentiator and the proposed averaged differentiator, which is an FIR comb filter. The proposed filter is very similar to the ideal differentiator at low frequencies with less than a 1-dB deviation below 5 kHz. At high frequencies, much attenuation is achieved, which is necessary to suppress the potential beating artifacts.

The waveform and spectrum resulting from the use of the proposed averaged differentiator are shown in Figure 6. It is seen that in the discrete-time waveform in Figure 6a the transitions from the maximum value (near +1) to the minimum value (near −1) are smoother than in Figure 4e. The corresponding spectrum (see Figure 6b) has decayed faster at high frequencies than that in Figure 4f. Listening confirms that this attenuation of the highest octave suppresses the aliased components and slightly improves the sound quality.

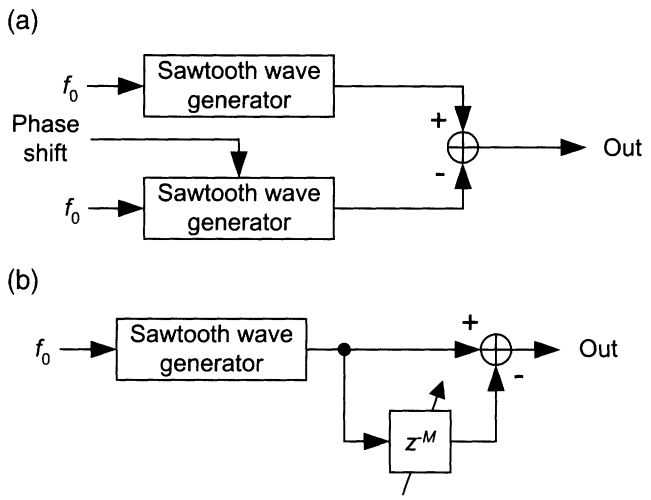
Pulse Waveform

Another common waveform used in subtractive synthesis is the rectangular pulse waveform. A special case is the square wave, which is a symmetric rectangular wave with a duty cycle of 50%. A rectangular waveform can be produced by subtracting two sawtooth waves with a proper phase shift. There are two different principles to implement this, as shown in Figure 7. Any sawtooth waveform generator principle can be used. We consider the use of the DPW algorithm described above.

Figure 7a uses two sawtooth waveform generators with a phase shift to generate a pulse waveform. This two-oscillator method allows smooth pulse-width modulation, since the phase shift can be continuously controlled by offsetting the counter in one of the oscillators. The computational cost of this method is simply twice that of a single sawtooth generator.

Alternatively, as suggested by Lowenfels (2003),

Figure 7. Two alternative ways for implementing a rectangular pulse generator: (a) a difference of two sawtooth wave generator outputs and (b) a combination of a sawtooth wave-generator and an FIR comb filter.

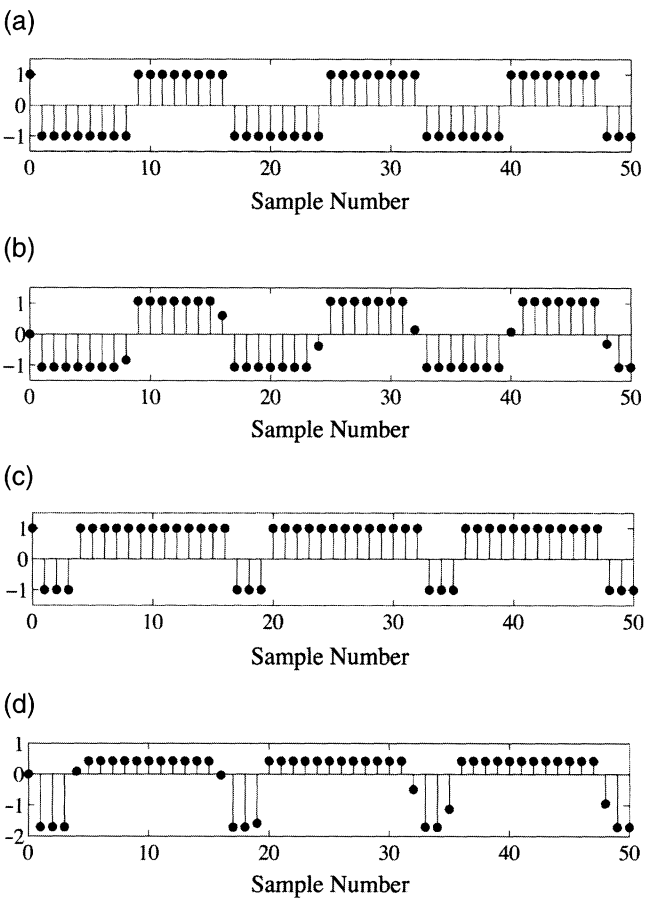


an FIR comb filter can be used to copy and shift a sawtooth wave to produce a pulse wave, as depicted in Figure 7b. This basic implementation requires some additional memory and one addition per sample. However, only phase shifts corresponding to an integral number of samples are available. Consequently, not just any pulse-width can be implemented accurately, and real-time modulation of the pulse-width will not sound smooth but will be contaminated by zipper noise.

An interpolated delay line can be used in the structure of Figure 7b to enable continuous control of the phase shift for pulse-width modulation. In practice, a delay line is then cascaded with a fractional delay filter, which produces the decimal part of the delay-line length. A first-order all-pass filter or a low-order Lagrange interpolation FIR filter (Laakso et al. 1996) is a good alternative for implementing the required fractional delay. However, the computational cost of the FIR comb filter with an interpolated delay line is approximately the same as that of two DPW-type sawtooth waveform generators. For some hardware systems, such as signal processors, the need for additional delay-line memory may be inconvenient because of the very limited on-chip memory. Thus, the two-oscillator method shown in Figure 7a is a more attractive realization scheme for pulse-width modulation.

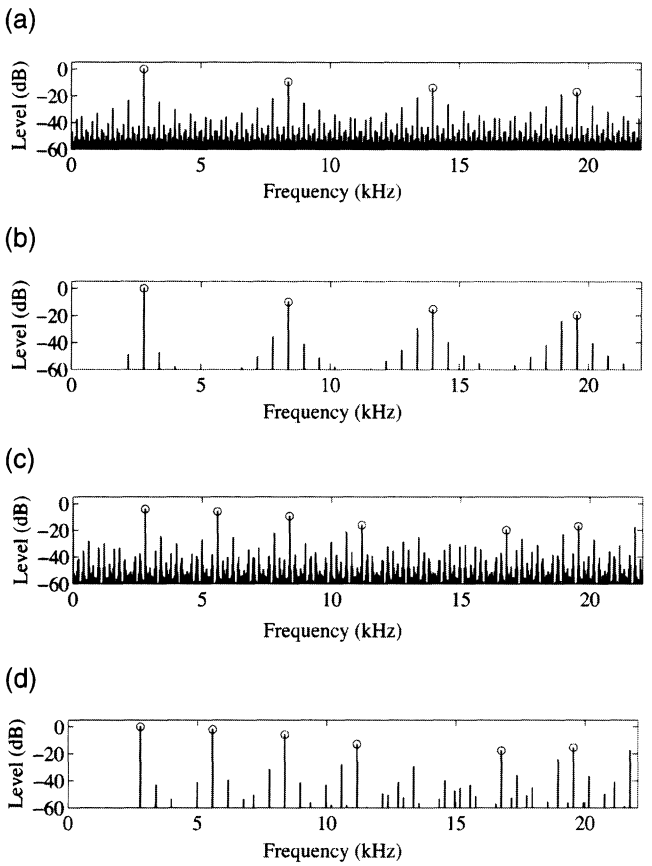
Figure 8 shows examples of trivial pulse waveforms and the same waveforms generated as a difference of two DPW sawtooth generators with an appropriate phase shift. Their spectra are shown in Figure 9. The trivial square wave—or a pulse wave with 50% duty cycle—displayed in Figure 8a suffers from heavy aliasing, as can be seen in Figure 9a, where everything else but the four odd partials of the fundamental are images caused by aliasing. The DPW-based square wave shown in Figure 8b has a much cleaner spectrum, as seen in Figure 9b: just a fraction of aliased components are visible above the –60-dB level, and all aliased components at frequencies below 10 kHz have been suppressed considerably. However, at high frequencies the level of some first-generation images have not been reduced at all; compare, for example, the levels of the spectral peak at 19 kHz in Figures 9a and 9b.

Figure 8. Comparison of rectangular waveforms: (a) The trivial and (b) DPW-based square waveforms; (c) the trivial and (d) DPW-based rectangular pulse wave with 20% duty cycle.



As another example, the trivial and DPW-based rectangular pulse waves with a duty cycle of 20% are

Figure 9. The spectra of the waveforms in Figure 8.



given in Figures 8c and 8d. Again, a similar improvement is obtained: aliasing is reduced much at low and mid frequencies, but not that much at high frequencies around 20 kHz (see Figures 9c and 9d). The DC component (i.e., the average signal level) is also smaller in the DPW-based signal than in the trivial one.

Triangular Waveform

The triangular waveform is also a typical source signal in subtractive synthesis. In discrete-time triangular waveform generation, aliasing is not as serious a problem as it is in the cases of the sawtooth and the pulse waveforms. This stems from the fact that the spectrum of the triangular waveform falls off at the rate of about –12 dB per octave, and not –6 dB per octave. For this reason, a trivial triangular waveform generator may provide a sufficient sound quality for subtractive synthesis. It is realized with a bipolar modulo counter that toggles its direction of counting every time it reaches the value +1 or –1.

It is possible to devise a high-quality algorithm to produce a triangular waveform with reduced aliasing. Lane et al. (1997) proposed one such algorithm, which is based on filtering a nonlinearly distorted sine wave. We show how the triangular waveform can be produced by employing the bipolar piecewise parabolic waveform.

Figure 10. The triangular waveform generation based on a piecewise parabolic signal.

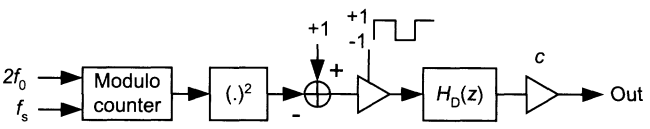


Figure 10

Figure 11. Signals from intermediate phases of the proposed DPW-based triangular waveform algorithm: (a) The bipolar modulo counter, (b) the squared sawtooth (filled circles) and the square (open circles),

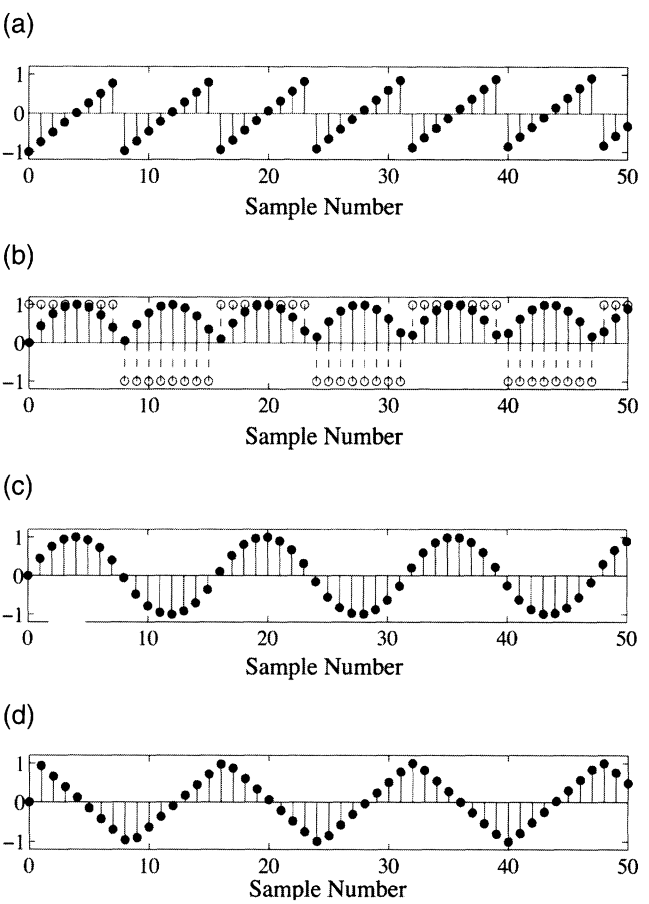


Figure 11

(c) the bipolar piecewise parabolic, and (d) the approximated triangular signals. The fundamental frequency of the obtained triangular signal is 2793.8 Hz and that of the modulo counter is two times higher, 5587.6 Hz.

Figure 12. (a) The trivial triangular, (b) the bipolar piecewise parabolic, and (c) the alias-free triangular waveforms. The fundamental frequency is 2793.8 Hz.

Figure 13. The spectra of the waveforms shown in Figure 12: (a) the trivial triangular, (b) the bipolar piecewise parabolic, and (c) the alias-free triangular waveforms. The desired spectral components, or odd harmonics of the fundamental (2793.8 Hz, 8381.4 Hz, 13969 Hz, . . .), are circled (o).

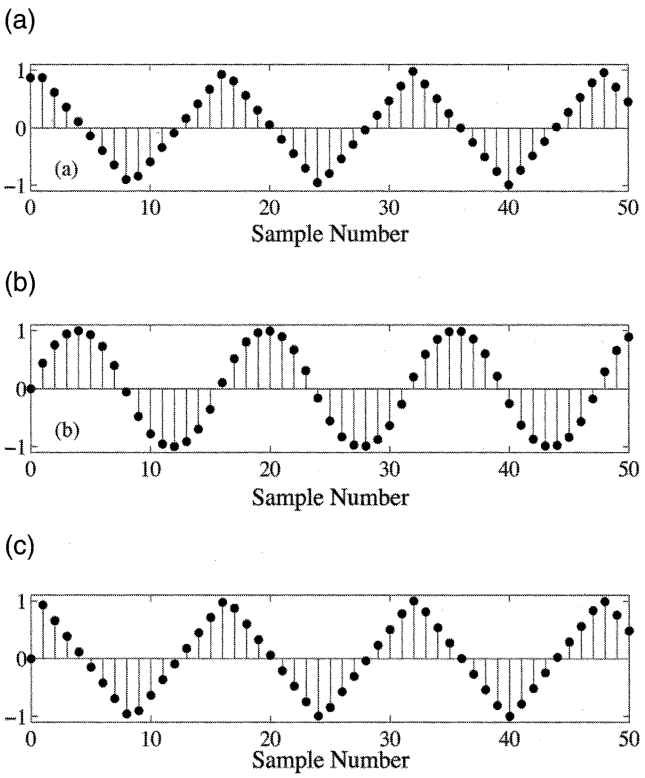


Figure 12

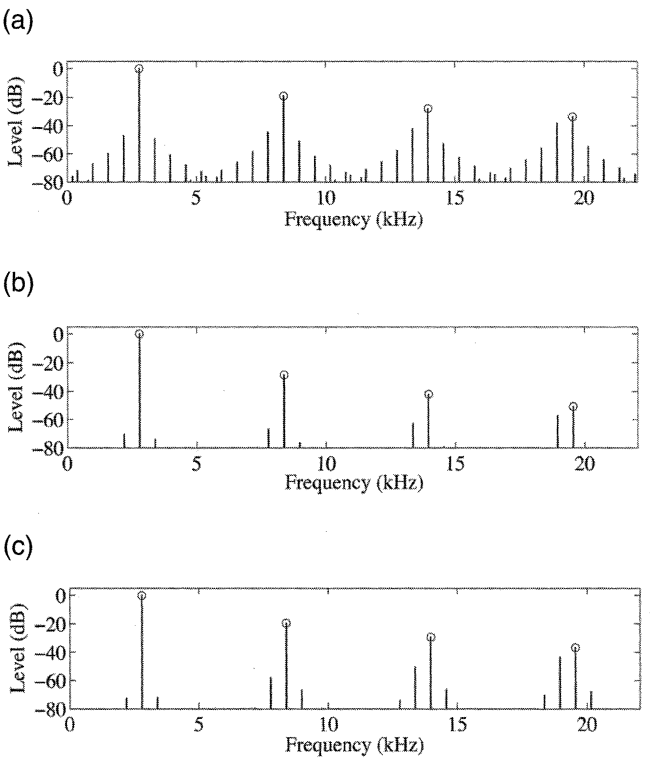


Figure 13

The block diagram of the proposed algorithm is shown in Figure 10. A bipolar modulo counter is used, where the fundamental frequency is twice that of the desired triangular signal. This initial waveform, which is equivalent to the trivial sawtooth signal, is shown in Figure 11a. After squaring the initial waveform, it must be turned upside down by subtracting it from one (see Figure 11b). The signal is modulated by a (trivial) square wave with the fundamental frequency $2f_0$ (see Figure 11b, open circles) to produce a piecewise parabolic waveform of the desired type (see Figure 11c). The square waveform must be phase-locked to the bipolar modulo counter; this is easy by toggling the sign of the square waveform when the counter reaches +1. Finally, the resulting bipolar piecewise parabolic waveform is differentiated and scaled to obtain a triangular waveform with reduced aliasing, which is presented in Figure 11d.

Figures 12 and 13 compare the wave shapes and spectra of the trivial and DPW-based triangular waveforms. It is seen in Figures 12a and 12c that the difference between these two waveforms is microscopic, apart from the initial transient in Figure 12c, which lasts for one sample when the first-order FIR differentiator is used. Nevertheless, the spectra given in Figure 13a and Figure 13c are quite different: the level of the aliased components is much reduced at low frequencies in Figure 13c. Figure 12b shows the bipolar parabolic signal, whose spectrum is given in Figure 13b. It can be noticed that the spectrum of this signal decays fast with frequency, about –18 dB per octave. This is why aliasing is less severe than in the case of the trivial triangular wave. It can also be noted that the triangular wave approximation (see Figure 12c) is perfectly symmetric above and below the zero level. Thus, its spectrum (see Figure 13c) is free of even harmonics, which are not well suppressed in some other triangular waveform approximations.

Digital Resonant Filters

A resonant filter used in computer music differs from traditional IIR filters in mainly three ways: the parameters are changed at a rapid rate, the order is usually predetermined instead of matching to certain stop-band attenuation specification, and a controllable resonant peak is introduced near the cutoff frequency.

A "perfect" digital resonant filter then fulfills the following criteria.

1. Coefficient update should be fast. To avoid clicks, the coefficients should be updated on a per-sample basis (Rossum 1992).
2. The filter cutoff and resonance parameters should be decoupled. Change in one should not affect the other.
3. The filter should stay unconditionally stable as long as parameters are inside the allowed range.
4. The filter should have a response similar to an existing analog resonant filter. Some analog filters have a characteristic sound that should be emulated, if possible.
5. The filter should be capable of self-oscillation.

A number of filters trying to meet the criteria have been developed. We take a closer look at the Moog low-pass filter (Moog 1965) here.

Digital Moog Filter

The Moog ladder filter (Moog 1965) can be considered the first musical filter. It features independent voltage control of both the cutoff frequency f_c and the resonance C_{res} while also having a characteristic sound of its own. The filter consists of four identical one-pole low-pass sections (implemented with an innovative transistor ladder circuit) in series with a global negative feedback to produce a resonant peak near the cutoff frequency.

A digital model of the Moog filter was first presented by Stilson and Smith (1996b). As in the analog prototype, it has four one-pole filters in series, and a global feedback is used to produce the resonance. To realize the filter, a unit delay has to be inserted in the feedback path, but this couples the cutoff and the resonance controls. Various ways of compensation have been examined, with the "compromise" version (Stilson and Smith 1996b) being the most attractive. The compromise version inserts a zero at $z = -0.3$ inside each one-pole filter section, thus mostly decoupling the resonance and the cutoff parameters. The modified one-pole filter structure is shown in Figure 14.

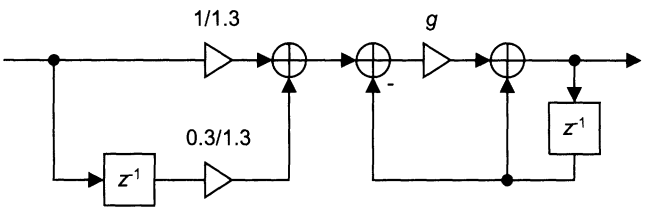


Figure 14. The compromise one-pole filter section.

The coefficient g determines the cutoff frequency. For very low frequencies, it can be trivially derived from the analog prototype filter as $g \approx 2\pi f_c/f_s$, where f_c is the desired cutoff frequency and f_s is the sampling rate. This, however, leads to the actual cutoff frequencies increasingly deviating from the desired. Due to the transfer-function zero at $z = -0.3$ and the unit delay in the feedback path, the resonant frequency cannot be easily predicted. Figure 15a compares three ways of calculating coefficient g: the impulse invariant, the bilinear, and the direct linear method.

All of the methods deviate from the desired tuning at high cutoff frequencies. Some form of compensating this deviation must therefore be used. A least-squares fit of a fourth-order polynomial to compensate the directly tuned section gives the following equation for parameter g:

$$g = 0.9892\omega_c - 0.4342\omega_c^2 + 0.1381\omega_c^3 - 0.0202\omega_c^4 \quad (2)$$

where $\omega_c = 2\pi f_c/f_s$ is the cutoff frequency in radians per sample. The tuning error, see Figure 15a, is then below 15 cents within the 0 to 14-kHz range of the cutoff frequency at a 44.1-kHz sampling rate. This is accurate enough so that the filter can be tuned to a multiple of a tone's fundamental frequency to emphasize an individual harmonic or, when self-oscillating, to produce an additional harmonic.

Figure 15. (a) Deviation in tuning of the resonance frequency in the compromise Moog filter using the scaled impulse invariant transform, the bilinear z-transform, the direct linear scaling, and the polynomial method for calculating the filter cutoff coefficient. (b) Deviation in feedback amount from ideal.

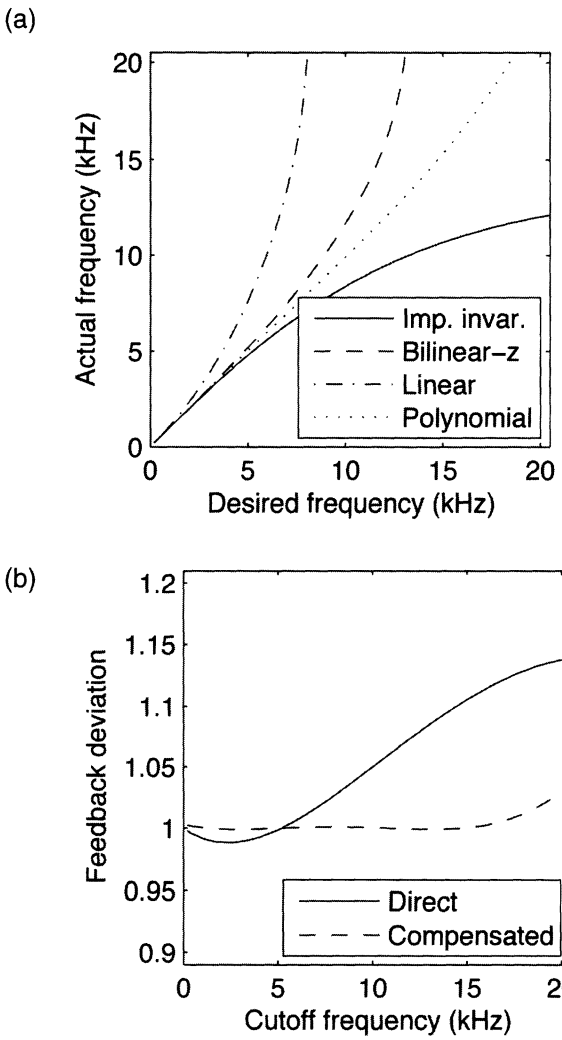
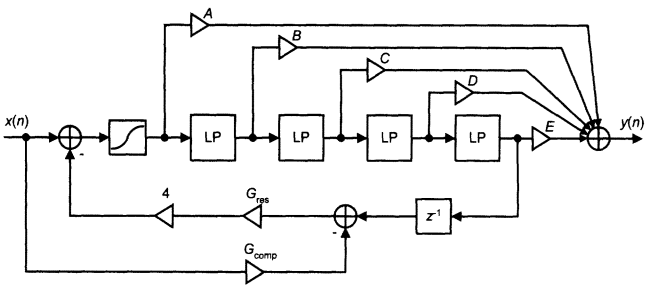


Figure 15b shows the error in the amount of feedback, too. This error can similarly be compensated by a suitable polynomial. Using

$$G_{res} = C_{res}\,(1.0029 + 0.0526\omega_c - 0.0926\omega_c^2 + 0.0218\omega_c^3), \quad (3)$$

where $0 \le C_{res} \le 1$, the error is less than 1% at frequencies below 17 kHz. This polynomial has been obtained by fitting in the least-square sense a third-order polynomial to the inverse function of the feedback deviation curve shown in Figure 15b. The corrected behavior is also displayed in Figure 15b.

Figure 16. Improved Moog-style filter. Each block labeled as 'LP' contains the one-pole filter structure shown in Figure 14. Coefficients A, B, C, D, and E can be used to select the type of output (low-pass, high-pass, band-pass, or notch filter or one of their combinations).



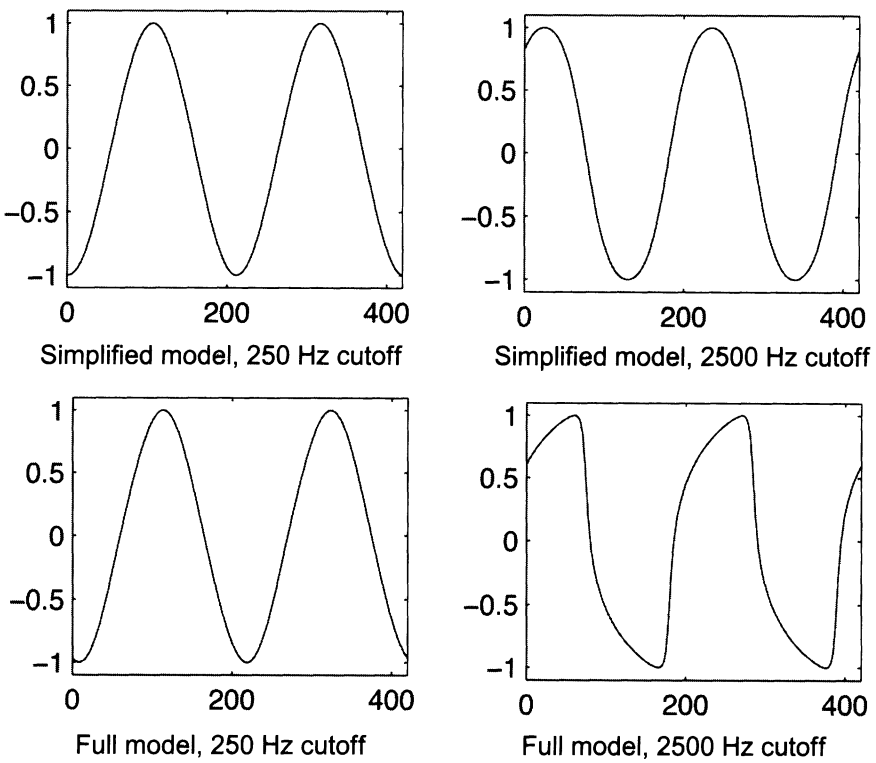
Improved Moog Filter

While the Stilson-Smith Moog filter model (Stilson and Smith 1996b) is certainly useful and solves the problem of fast coefficient update, it becomes unstable at very large resonance values and it cannot self-oscillate. Furthermore, it does not emulate the characteristic distortion produced by the original transistor ladder circuit. Huovilainen has developed an improved model that simulates the ladder circuit by inserting nonlinearities inside the one-pole filter sections (Huovilainen 2004). This improved model emulates more closely the characteristic sound of the original Moog filter and is also capable of self-oscillation. A disadvantage is the need for five hyperbolic tangent ('tanh') function evaluations per sample and for oversampling by a factor of at least two.

We propose an alternative extended model presented in Figure 16. The embedded nonlinearities within sections are replaced by a single nonlinearity, thus greatly reducing the computational cost of the filter. We have used the 'tanh' function for the nonlinearity, but any smoothly saturating function may be used. There is a difference in the sound compared to the full nonlinear Moog filter model (Huovilainen 2004), but this new model can emulate most of the behavior, such as self-oscillation. Its output is also always bounded.

The traditional Moog filter and similar cascaded one-pole filters suffer from a decreasing pass-band gain as the resonance is increased, because the resonance is produced with a global negative feedback. If a fraction of the input signal is subtracted from the feedback signal before multiplying by the resonance parameter, the pass-band gain change can be

Figure 17. Comparison of output waveforms of the simplified (see Figure 16) and the full (Huovilainen 2004) nonlinear Moog filter models using a sinusoidal input for two different choices of cutoff frequency.



controlled (Curtis Electromusic Specialties 1984). A value of 1.0 for the G_{comp} parameter in Figure 16 keeps the pass-band gain constant. This, however, results in a large increase of the output amplitude as the resonance is increased. To maintain the overall level approximately constant, the value of the G_{comp} parameter should be set to 0.5, resulting in a 6-dB pass-band gain decrease at the maximum resonance (compared to a 12-dB decrease in the original Moog model).

Figures 17 and 18 compare the two models for 210-Hz sinusoidal and sawtooth inputs using moderate overdrive. The input peak-to-peak amplitude is 2.0 for the full model and 1.0 for the simplified model in order to visually match their behavior. The output signals have been normalized to ease the comparison. The difference is clearly visible in the sine wave when the cutoff is much above the input frequency. For the sawtooth wave the difference is smaller. An interesting effect is the asymmetric output shape in the full model even though both the input waveform and the distortion function are symmetric. The difference almost disappears when the resonance parameter is increased. This can be attributed to the negative feedback employed to produce the resonance peak. As soon as the pass-band gain compensation is used, the difference is restored. The simplified Moog model of Figure 16 therefore seems best suited for uses where a large overdrive is not demanded. The simple nonlinearity still limits the amplitude when a very high resonance amount or the self-oscillation mode is used.

Another improvement to the original Moog model is the addition of various frequency response modes besides the original 24-dB/oct low-pass filter mode. This can be easily achieved by mixing the outputs of the individual sections with different weights. The concept was pioneered in the Oberheim Xpander and Matrix-12 synthesizers (Oberheim 1984), but it was not widely used due to the large number of required components and the need for precision resis-

Figure 18. Comparison of output waveforms of the simplified (see Figure 16) and the full (Huovilainen 2004) nonlinear Moog filter models using a sawtooth wave input with (left) no resonance, (center) with 50% resonance, and (right) with 50% resonance combined with a 6-dB passband gain compensation.

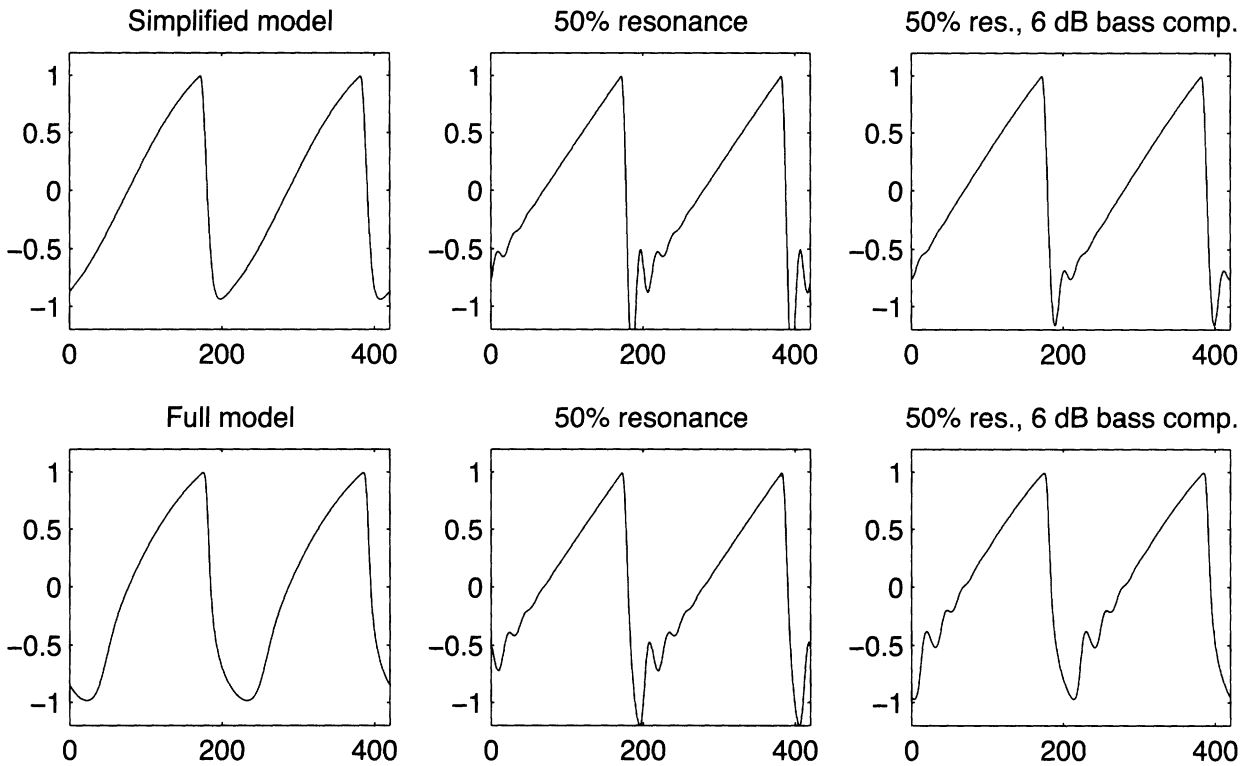


Table 1. Examples of Weighting Coefficient Values for Typical Magnitude Response Types That Can Be Obtained with the Filter Structure of Figure 16.

Filter Type	*A*	*B*	*C*	*D*	*E*
Two-pole low-pass	0	0	1	0	0
Four-pole low-pass	0	0	0	0	1
Two-pole band-pass	0	2	−2	0	0
Four-pole band-pass	0	0	4	−8	4
Two-pole high-pass	1	−2	1	0	0
Four-pole high-pass	1	−4	6	−4	1

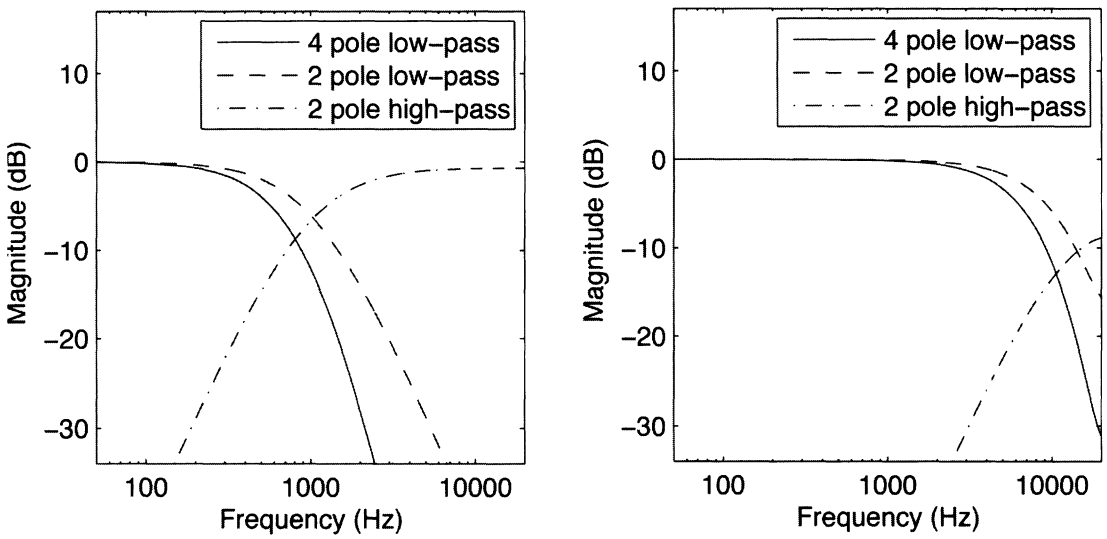
tors. Accurate mixing is trivial in the digital domain, and a large number of different low-pass, band-pass, high-pass, and notch filter responses and their combinations can be easily realized. Table 1 shows weighting coefficient values for common filter types. Morphing between these modes is possible by changing the coefficients at runtime, thus allowing interesting modulation possibilities.

Figure 19 shows examples of four-pole low-pass, two-pole low-pass, and two-pole high-pass filter responses produced by the filter structure of Figure 16. More examples of responses as well as the equations for deriving the coefficients are available elsewhere (Oberheim 1984). The filter response shapes remain relatively constant independent of the cutoff frequency, but the amplitude falls as the cutoff frequency is increased. The drop depends on the response used but can be readily compensated using a polynomial gain correction. For most responses, a simple linear correction will be sufficient.

Conclusions

We discussed new oscillator and resonant filter algorithms. The DPW oscillator algorithm generates an approximate sawtooth waveform that has reduced aliasing with respect to the trivial sawtooth waveform (i.e., the modulo counter output). This re-

Figure 19. Example filter responses with a 1-kHz and a 10-kHz cutoff frequencies.



cently proposed method is probably the simplest useful technique for this purpose, because only the trivial sawtooth is simpler, which is practically useless due to its heavy aliasing. In this paper we proposed an alternative differentiator for the DPW algorithm that suppresses high frequencies to avoid audible artifacts. We introduced methods to produce pulse and triangular waveforms based on the parabolic waveform.

The new nonlinear model of the Moog ladder filter is based on a cascade of four first-order IIR filters and a memoryless nonlinearity within a feedback loop. The proposed new Moog filter structure has nice advantages, such as a smaller computational cost than that of a recently proposed nonlinear filter structure, the adequate decoupling of the cutoff frequency and the resonance parameters, and the possibility to obtain various types of filter responses by selecting a weighted sum of different output points. The proposed methods allow the synthesis of retro sounds with a modern computer.

Acknowledgments

This work has been financed partly by the Academy of Finland (project no. 104934, "Control, Analysis, and Parametric Synthesis of Audio Signals"). The authors are grateful to Teppo Karema and Tomi Valkonen of VLSI Solution Oy (Tampere, Finland) for helpful discussions.

References

Brandt, E. 2001. "Hard Sync without Aliasing." *Proceedings of the 2001 International Computer Music Conference.* San Fransisco: International Computer Music Association. http://www-2.cs.cmu.edu/~eli/L/icmc01/hardsync.html

Burk, P. 2004. "Band Limited Oscillators Using Wave Table Synthesis." In *Audio Anecdotes II—Tools, Tips, and Techniques for Digital Audio,* pp. 37–53. Eds. K. Greenebaum and R. Barzel. Wellesley, MA: A. K. Peters, Ltd.

Chamberlin, H. 1985. *Musical Applications of Microprocessors,* 2nd Ed., Hayden Book Company.

Chaudhary, A. 1998. "Bandlimited Simulation of Analog Synthesizer Modules by Additive Synthesis." *Proceedings of the Audio Engineering Society 105th Convention,* paper no. 4779, San Francisco, CA: Audio Engineering Society.

Clavia DMI AB. 2002. "The Virtual Analog Concept." http://www.clavia.se/nordlead2/concept.htm

Curtis Electromusic Specialties. 1984. "CEM 3328 four pole low-pass VCF." http://www.synthtech.com/cem/c3328pdf.pdf

Huovilainen, A. 2004. "Nonlinear Digital Implementation of the Moog Ladder Filter." *Proceedings of the International Conference on Digital Audio Effects.* Naples, Italy, pp. 61–64. http://dafx04.na.infn.it/

Kaplan, S. J. 1981. "Developing a Commercial Digital Sound Synthesizer." *Computer Music Journal* 5(3): 62–73.
Laakso, T. I., V. Välimäki, M. Karjalainen, and U. K. Laine. 1996. "Splitting the Unit Delay—Tools for Fractional Delay Filter Design." *IEEE Signal Processing Magazine* 13(1):30–60.
Lane, J., D. Hoory, E. Martinez, and P. Wang. 1997. "Modeling Analog Synthesis with DSPs." *Computer Music Journal* 21(4):23–41.
Lazzaro, J., and J. Wawrzynek. 2004. "Subtractive Synthesis without Filters." In *Audio Anecdotes II—Tools, Tips, and Techniques for Digital Audio,* pp. 55–64. Eds. K. Greenebaum and R. Barzel. Wellesley, MA: A. K. Peters, Ltd.
Lowenfels, D. 2003. "Virtual Analog Synthesis with a Time-Varying Comb Filter." Proceedings of the Audio Engineering Society 115th Convention, paper no. 5960. New York: Audio Engineering Society.
Moog, R. A. 1965. "Voltage-Controlled Electronic Music Modules." *Journal of the Audio Engineering Society* 13(3):200–206.
Moore, F. R. 1990. *Elements of Computer Music.* Englewood Cliffs, NJ: Prentice-Hall.
Moorer, J. A. 1976. "The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulae." *Journal of the Audio Engineering Society* 24(9): 717–727.
Oberheim Electronics, Inc. 1984. "Oberheim Xpander Service Manual." Los Angeles, CA.
Rossum, D. 1992. "Making Digital Filters Sound Analog." *Proceedings of the 1992 International Computer Music Conference.* San Fransisco: International Computer Music Association, pp. 30–33.
Stilson, T., and J. O. Smith. 1996a. "Alias-Free Digital Synthesis of Classic Analog Waveforms." *Proceedings of the 1996 International Computer Music Conference.* San Fransisco: International Computer Music Association, pp. 332–335. http://ccrma.stanford.edu/~stilti/papers/
Stilson, T., and J. O. Smith. 1996b. "Analyzing the Moog VCF with Considerations for Digital Implementation." *Proceedings of the 1996 International Computer Music Conference.* San Fransisco: International Computer Music Association, pp. 398–401. http://ccrma.stanford.edu/~stilti/papers/
Välimäki, V. 2005. "Discrete-Time Synthesis of the Sawtooth Waveform with Reduced Aliasing." *IEEE Signal Processing Letters* 12(3):214–217.
Winham, G., and K. Steiglitz. 1970. "Input Generators for Digital Sound Synthesis." *Journal of the Acoustical Society of America* 47(part 2): 665–666.