Discrete-Time Synthesis of the Sawtooth Waveform With Reduced Aliasing

Vesa Välimäki, *Senior Member, IEEE*

***Abstract*—An efficient signal processing algorithm for generating a sawtooth waveform is proposed. The algorithm improves the trivial waveform sampling method, which suffers from low sound quality due to aliasing. The basic version of the new algorithm differentiates a piecewise parabolic waveform. Another version of the algorithm oversamples and decimates the parabolic wave with a simple filter prior to differentiation. The two algorithm variants improve the signal-to-noise ratio (SNR) over the trivial method by 10 and 15 dB, respectively. A perceptually weighted SNR suggests a larger subjective improvement. The proposed methods are applicable in a digital signal processor (DSP) implementation of subtractive sound synthesis.**

***Index Terms*—Acoustic signal processing, antialiasing, audio oscillators, music, signal synthesis.**

I. Introduction

Subtractive synthesis is one of the early principles used in music synthesizers. Recently, it has become interesting because it is used in digital emulations of analog sound synthesis. Subtractive synthesis is currently used in hardware- and software-based music synthesizers and in ringing tones of mobile phones. The basic principle in subtractive synthesis is first to generate a signal with a rich spectral content and then to filter that signal with a time-varying resonant filter. Often, the input signal is a periodic waveform, such as a triangular, rectangular, or sawtooth wave, or a sum of two or more such waveforms with different fundamental frequencies. In developing a digital implementation of subtractive synthesis, the main challenge is to find efficient algorithms for generating periodic waveforms that do not suffer from excessive aliasing distortion. This letter focuses on the generation of the sawtooth waveform.

An obvious algorithm to generate a discrete-time sawtooth wave with signal values between -1 and 1 samples the continuous-time rising ramp function and shifts it to remove the dc offset

$$s(n) = 2\left(n\frac{f}{f_{\mathrm{s}}} \bmod 1\right) - 1 \tag{1}$$

where $n = 0, 1, 2, \ldots$ is the discrete sample index, f is the fundamental frequency of the tone, and f_s is the sampling rate. This bipolar modulo counter can be realized easily with a computer or a signal processor. In fact, many signal processors have wrapping registers that automatically implement the modulo operation, keeping the result of arithmetic operations always in the desired range of values.

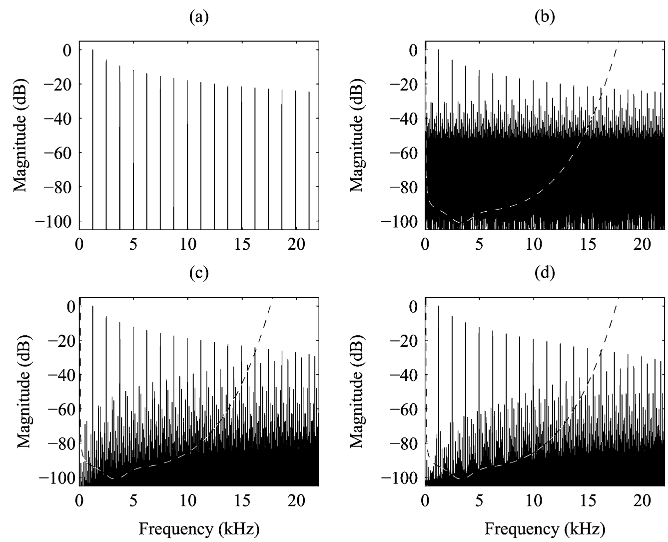

Fig. 1. Magnitude spectra of discrete-time sawtooth signals generated with (a) additive synthesis, (b) the trivial technique, (c) the proposed DPW method, and (d) the proposed DPW2X method. The fundamental frequency of the signals is 1245 Hz, and the sampling rate is 44.1 kHz. In cases (b)–(d), the dashed line represents the threshold of hearing, assuming that the signal is played at 96 dB SPL.

Unfortunately, the discrete-time audio signal obtained using (1) is notorious for its distorted sound quality [1]. The problem is that the continuous-time sawtooth wave is not bandlimited, but it has an infinite number of spectral components, falling off at the rate of about -6 dB per octave. The discretization of the waveform leads to aliasing, where all spectral components above the Nyquist limit $f_{\mathrm{s}}/2$ are reflected down to the audible frequency range. This is heard as disturbing interference.

The magnitude spectrum of the ideal sawtooth wave with fundamental frequency $f = 1245$ Hz is represented in Fig. 1(a). Fig. 1(b) shows an example spectrum of a tone generated using (1) at the sampling rate of 44.1 kHz. It contains many extra components not present in Fig. 1(a). The dashed line is an approximation of the hearing threshold, taken from [2], assuming that the tone is played at the sound pressure level (SPL) of 96 dB. While this curve should not be literally taken as the threshold of hearing for this spectrum—mainly because the auditory masking phenomenon is ignored—it illustrates the frequency-dependent sensitivity of hearing. The figure suggests that many aliased spectral components below 15 kHz may be audible, and listening confirms that some of them are.

Manuscript received June 29, 2004; revised September 28, 2004. This work was supported in part by Academy of Finland under Project 104934. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Darren B. Ward.

The author is with the Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, FI-02015 HUT, Espoo, Finland (e-mail: vesa.valimaki@hut.fi).

Digital Object Identifier 10.1109/LSP.2004.842271

A straightforward method to improve the quality of the trivial sawtooth wave is to oversample the waveform (see [1, pp. 116–117]). The aliasing resulting from using (1) is less severe when the fundamental frequency is small with respect to the sampling rate. In practice, this approach leads to a highly oversampled and, therefore, inefficient system, because the spectrum of the sampled sawtooth wave decays slowly.

An ideal bandlimited sawtooth wave can be computed using additive synthesis, in which each harmonic component is generated separately using a sinusoidal oscillator [3]. The ideal sawtooth waveform can be expressed as

$$s_{\mathrm{id}}(n) = -\sum_{k=1,2,3,\ldots}^{K} \frac{1}{k}\sin(2\pi f k n/f_{\mathrm{s}}) \tag{2}$$

where $K = \mathrm{floor}(f_{\mathrm{s}}/2f)$ is the number of harmonics below the Nyquist limit that will be computed. This method was used for producing Fig. 1(a). The drawback of this approach is that the number of sinusoidal oscillators becomes large when the fundamental frequency is low. Computational savings can be obtained by omitting the highest frequencies or by storing sets of sinusoidal components in several wavetables, as in group additive synthesis [4], but still, this approach is considerably more costly than (1).

Generation of alias-free waveforms for subtractive synthesis has received consideration over the years. Winham and Steiglitz [5] and Moorer [6] have proposed the use of a closed-form summation formula to generate a waveform containing a specified number of sinusoidal components. A low-order digital filter can shape the roll-off rate of the spectrum so that, for example, the sawtooth waveform can be closely approximated. While this method does not require as many operations per sample as additive synthesis at low frequencies, it has its drawbacks: It requires a division per sample, and this leads to numerical errors that must be controlled. One solution is to compute many versions of the bandlimited sawtooth waveform using (2) with a different number of harmonic components and then use them in wavetable synthesis [3]. This is efficient to compute and free of numerical problems, but it consumes much memory. A variation that saves memory consists of a small set of wavetables, such as one per octave, which are played at the various speeds using sample rate conversion techniques [7].

Stilson and Smith [8] have reviewed different approaches to generate alias-free waveforms using bandlimited impulse trains and filters. This method is based on several windowed and sampled sinc functions that are stored in memory. Brandt proposed to linearly combine a trivial waveform with a bandlimited pulse to reduce aliasing [9]. The pulse used is a minimum-phase bandlimited step function (minBLEP) obtained by integrating a minimum-phase windowed sampled sinc function [9]. The minBLEP-based algorithm requires a division each period, and its computational load depends on the fundamental frequency. It consumes some memory, because an oversampled minBLEP sequence must be designed and stored in advance for real-time implementation.

Lane *et al.* [10] introduced algorithms that full-wave rectify a sine wave and modify the resulting signal with two digital filters. In Lane's sawtooth algorithm [10], the first filter has the transfer function

$$H(z) = \frac{\alpha(1 - z^{-1})}{1 - \gamma z^{-1}} \tag{3}$$

where $\gamma = \cos(\omega_{\mathrm{c}})/[1 + \sin(\omega_{\mathrm{c}})]$, $\alpha = (1+\gamma)/2$, and $\omega_{\mathrm{c}} = 2\pi \times 16f/f_{\mathrm{s}}$. (Unfortunately, the formula for γ has a mistake in [10]: It shows a different argument for the cos and the sin functions, although they must be the same.) The second filter is a Butterworth lowpass filter. In the examples of [10], various choices for the filter order (4 or 8) and cutoff frequency (4, 8, or 12 kHz) are given. The sawtooth wave approximation computed with Lane's algorithm contains aliased components, but they are suppressed in comparison to the trivial sawtooth wave.

In this work, we follow the guideline of Lane *et al.* [10]. The ultimate design goal is not to suppress the aliasing completely but rather to reduce it sufficiently at selected frequency areas so that it will be inaudible. Other goals are computational efficiency and a small memory space requirement. These features are desirable in digital audio algorithms used in mobile applications, where the core processor and memory are on the same chip. As a result of these restrictions, a compromise is required to save as much computational expense as possible, and thus, aliasing is reduced considerably, but its audibility is not minimized in a strict sense. The idea behind the algorithm we propose has similarities to the concepts of pre-emphasis [11] and noise shaping [12], which are used to reduce the audible noise caused by recording, transmission, or quantization of audio signals. Pre-emphasis is also used in linear prediction of speech signals to facilitate formant estimation [13].

The basic version of the new algorithm is introduced in Section II, and an enhanced version is described in Section III. Various sawtooth wave algorithms are compared in Section IV. Section V concludes this letter.

II. Differentiated Parabolic Wave

A discrete-time signal that closely resembles the sawtooth wave (but with a smaller aliasing error) can be produced by differentiating a parabolic waveform. The piecewise parabolic wave is obtained as the integral of the sawtooth waveform, which is a linear function of time t within a single period T, that is

$$p(t) = \int t \mathrm{d}t = \frac{t^2}{2}, \quad \text{for} -\frac{T}{2} \le t \le \frac{T}{2}. \tag{4}$$

Differentiating (4) with respect to t restores the linear function

$$s(t) = \frac{\mathrm{d}p(t)}{\mathrm{d}t} = t, \quad \text{for} -\frac{T}{2} \le t \le \frac{T}{2}. \tag{5}$$

The motivation to use a sampled version of the piecewise parabolic signal $p(t)$ is that its spectrum decays about -12 dB per octave, while that of the sawtooth wave falls off at the rate of about -6 dB per octave. The faster roll-off rate leads to suppressed aliasing. Differentiation restores the desired spectral tilt for the harmonic components, but at low and middle frequencies, the aliased part of the spectrum remains at a lower level than in the trivial sawtooth wave. At frequencies close to the Nyquist limit, the level difference of the desired and the aliased spectral components is not affected that much. We call the combination of the piecewise parabolic signal generation and differentiation the differentiated parabolic wave (DPW) algorithm. Fig. 2(a) shows its basic principle. Note that the bipolar modulo counter defined in (1) can be used.

Alternative realizations of the DPW algorithm can be obtained by varying the differentiator design. The simplest choice is a first-order finite impulse response (FIR) differentiator with the transfer function $(1-z^{-1})/2$. This filter attenuates the signal at high frequencies with respect to ideal differentiation. Still, it can be sufficient for audio applications when the sampling rate is high, such as 44.1 kHz. The resulting efficient version of the basic DPW algorithm is shown in Fig. 2(b). The algorithm requires two multiplications and two additions, one of which is used in incrementing the counter. Coefficient $c = f_\mathrm{s}/[4f(1 - f/f_\mathrm{s})]$, which contains factor 0.5 of the FIR differentiator, scales the amplitude of the output signal to be smaller than or equal to 1.0. Coefficient c becomes large when f is small, and thus, the signal values after the differentiator are very small. This can lead to quantization noise problems with finite-accuracy computation at low frequencies.

Fig. 1(c) shows an example spectrum of a signal obtained with the DPW method when the fundamental frequency is 1245 Hz and the sampling rate is 44.1 kHz. The reduction in aliasing on a wide frequency range is apparent when compared against the spectrum of the trivial sawtooth wave [see Fig. 1(b)]. Only at frequencies close to the Nyquist limit is the aliasing heavy and approaches the level of desired spectral components. Note, however, that at such high frequencies, the sensitivity of hearing is low.

III. Oversampled Version of the Algorithm

Oversampling is a common approach to reduce aliasing. In the two times oversampled DPW algorithm (DPW2X), the parabolic waveform is generated at a rate twice that of the output sample rate, the signal is filtered by a decimation filter, then the sampling rate is reduced by factor two by skipping every second sample, and finally, the signal is differentiated, as depicted in Fig. 2(c).

Different choices of the decimation and differentiation filters lead to alternative realizations of the DPW2X method. The decimation filter should be of low order, because the overall complexity of the algorithm must not be considerably increased. For this reason, we consider the simplest possibility, which is the two-point averaging filter with the transfer function $(1+z^{-1})/2$. It provides a high attenuation close to the oversampled Nyquist limit while in the passband, i.e., at frequencies below the original Nyquist limit, its attenuation is less than 3 dB, which is considered acceptable. When the averaging filter is applied to the two sample values generated during the same output sampling interval but not between those of consecutive intervals, the decimation by two is executed automatically. The first-order FIR differentiator is used, like previously. The resulting version of the DPW2X technique is detailed in Fig. 2(d). A scaling coefficient $c_2 = f_\mathrm{s}/[8f(1 - f/f_\mathrm{s})]$ gives a constant amplitude of 1.0. The method uses four additions (one for each modulo counter and for each filter) and three multiplications per output sample.

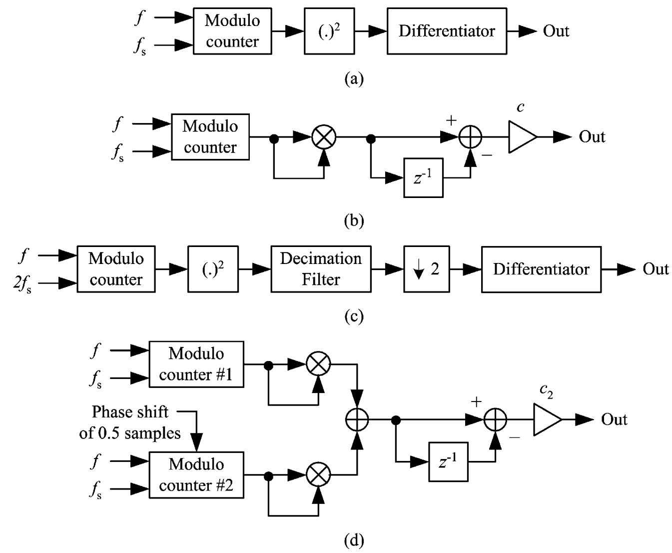

Fig. 2. Block diagrams of sawtooth wave generators based on (a) the general form of the DPW algorithm, (b) one of its particularly efficient realizations, (c) the generic form of the DPW2X, and (d) its efficient realization.

Fig. 1(d) shows an example spectrum of a signal obtained using this version of the algorithm. Comparing this spectrum with that in Fig. 1(c) demonstrates that the aliasing is further suppressed at low and middle frequencies.

IV. Comparison

To evaluate the improvement obtained with the new methods, the SNR was determined for various sawtooth signal approximations over the fundamental frequency range of the piano (27.5–4186 Hz). The algorithm proposed by Lane *et al.* [10] was included in the comparison, together with the trivial, the DPW, and the DPW2X methods. An eighth-order Butterworth lowpass with a cutoff frequency of 12 kHz was included in Lane's algorithm. The SNR was defined as the power ratio of the desired and undesired spectral components, that is, the harmonics and aliased components, respectively.

A long portion of the synthetic signal (1.0 s) was windowed using a Chebyshev window with a 120-dB sidelobe attenuation, and the discrete-time Fourier transform was computed at exact integral multiples of the fundamental frequency up to the Nyquist limit. The corresponding desired signal was synthesized by adding sinusoidal components with the obtained amplitude and phase. The squared sum of the sample values was used as the desired signal power in SNR calculation. To separate the harmonic and aliased components, the level of the reconstructed harmonic signal was adjusted so that its first harmonic had the same amplitude as that of the sawtooth approximation. The harmonic signal was then subtracted from the sawtooth signal. The squared sum of the residual signal was used as the power of the undesired signal in SNR calculation.

Fig. 3(a) shows the SNR figures for the four methods as a function of logarithmic frequency. The average SNR improvement obtained with the DPW and the DPW2X algorithms compared to the trivial algorithm is 10.1 and 14.5 dB, respectively. Informal listening reveals, however, that the subjective improvement is more striking than the SNR curves of Fig. 3(a)

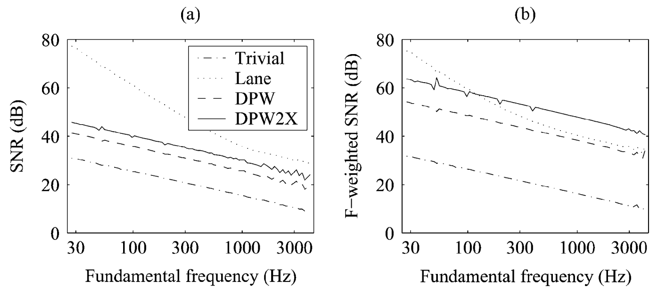

Fig. 3 (a) Conventional and (b) F-weighted SNR of the sawtooth wave approximations obtained using the trivial method, Lane's method [10], and two versions of the proposed algorithm.

suggest. One explanation is that the new methods considerably reduce the aliased spectral energy at low and middle frequencies, where the sensitivity of hearing is at its best. For this reason, a perceptually weighted SNR was computed for the same signals. In the SNR curves displayed in Fig. 3(b), the undesired signal power has been weighted with the F-weighting function, which imitates the hearing sensitivity to low-level signals [12]. The F-weighted SNR of the trivial method is only about 1 dB better than the unweighted one, because the spectrum of the disturbance is roughly flat. Conversely, the DPW and the DPW2X algorithms show an average SNR increase of 12.8 and 18.0 dB, respectively, due to weighting. Although Lane's algorithm has the best unweighted SNR, as illustrated in Fig. 3(a), the F-weighted SNR of the DPW2X algorithm is superior to Lane's algorithm at frequencies above 110 Hz, as can be seen in Fig. 3(b). At frequencies above 400 Hz, the weighted SNR of the DPW2X method is over 6 dB better than that of Lane's algorithm.

The proposed two techniques are computationally more efficient than Lane's algorithm, which consists of a sinusoidal oscillator, a first-order highpass filter, and a fourth- (or higher) order Butterworth lowpass filter [10]. These filtering operations take 11 multiplications and ten additions per sample. This corresponds to approximately five and three times more operations per sample than the DPW and the DPW2X methods, respectively. Furthermore, the modulo counter used in the proposed techniques can be implemented with fewer operations than the sinusoidal oscillator needed in Lane's method.

V. Conclusion

Two versions of the new algorithm for the generation of a discrete-time approximation of the sawtooth wave are proposed. The basic version, called DPW, computes a piecewise parabolic signal, which is differentiated to obtain an approximation of the sawtooth wave. An enhanced version of the algorithm, called DPW2X, is obtained by first generating the parabolic waveform at a sampling rate two times higher, then averaging two consecutive sample values, and finally differentiating the overall output signal. Efficient implementations of the two algorithm versions improve the SNR by approximately 10 and 15 dB, respectively, with respect to the trivial sawtooth wave. When a typical audio sample rate of 44.1 kHz is used, the aliasing is reduced mainly at low and middle frequencies, where human hearing is most sensitive. The psychoacoustically motivated F-weighted SNR shows 22- and 32-dB improvements for the two new methods over the trivial one. Other strengths of the proposed algorithm are the small amount of operations per sample and small memory consumption. These features make it attractive for music synthesis implementations, for example, in mobile applications.

Future work includes an evaluation of the influence of the masking phenomenon to the audibility of the remaining aliased spectral components.

Acknowledgment

The author is grateful to A. Huovilainen and J. Pakarinen of Helsinki University of Technology and to T. Karema and T. Valkonen of VLSI Solution Oy (Tampere, Finland) for helpful discussions.

References

[1] H. Campbell, *Musical Applications of Microprocessors*, 2nd ed. New York: Hayden, 1985.
[2] E. Terhardt, G. Stoll, and M. Seewann, "Algorithm for extraction of pitch and pitch salience from complex tonal signals," *J. Acoust. Soc. Amer.*, vol. 71, no. 3, pp. 679–688, Mar. 1982.
[3] F. R. Moore, *Elements of Computer Music*. Englewood Cliffs, NJ: Prentice-Hall, 1990, pp. 270–271.
[4] P. Kleczkowski, "Group additive synthesis," *Comput. Music J.*, vol. 13, no. 1, pp. 12–20, 1989.
[5] G. Winham and K. Steiglitz, "Input generators for digital sound synthesis," *J. Acoust. Soc. Amer.*, pt. 2, vol. 47, pp. 665–666, Feb. 1970.
[6] J. A. Moorer, "The synthesis of complex audio spectra by means of discrete summation formulae," *J. Audio Eng. Soc.*, vol. 24, no. 9, pp. 717–727, Nov. 1976.
[7] D. C. Massie, "Wavetable sampling synthesis," in *Applications of Digital Signal Processing to Audio and Acoustics*, M. Kahrs and K. Brandenburg, Eds. Norwell, MA: Kluwer, 1998, pp. 311–341.
[8] T. Stilson and J. Smith, "Alias-free digital synthesis of classic analog waveforms," in *Proc. Int. Comput. Music Conf.*, Hong Kong, China, 1996, pp. 332–335.
[9] E. Brandt, "Hard sync without aliasing," in *Proc. Int. Comput. Music Conf.*, Havana, Cuba, 2001, pp. 365–368.
[10] J. Lane, D. Hoory, E. Martinez, and P. Wang, "Modeling analog synthesis with DSPs," *Comput. Music J.*, vol. 21, no. 4, pp. 23–41, 1997.
[11] J. L. Hood, *Audio Electronics*, 2nd ed. New York: Elsevier, 1999.
[12] R. A. Wannamaker, "Psychoacoustically optimal noise shaping," *J. Audio Eng. Soc.*, vol. 40, no. 7/8, pp. 611–620, 1992.
[13] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*. New York: Springer-Verlag, 1976.