HIGHER-ORDER INTEGRATED WAVETABLE SYNTHESIS

Andreas Franck,
Acoustics Department
Fraunhofer Institute for Digital Media Technology
Ilmenau, Germany
andreas.franck@idmt.fraunhofer.de

Vesa Välimäki,
Department of Signal Processing and Acoustics,
Aalto University
Espoo, Finland
vesa.valimaki@aalto.fi

ABSTRACT

Wavetable synthesis is a popular sound synthesis method enabling the efficient creation of musical sounds. Using sample rate conversion techniques, arbitrary musical pitches can be generated from one wavetable or from a small set of wavetables: downsampling is used for raising the pitch and upsampling for lowering it. A challenge when changing the pitch of a sampled waveform is to avoid disturbing aliasing artifacts. Besides bandlimited resampling algorithms, the use of an integrated wavetable and a differentiation of the output signal has been proposed previously by Geiger. This paper extends Geiger's method by using several integrator and differentiator stages to improve alias-reduction. The waveform is integrated multiple times before it is stored in a wavetable. During playback, a sample rate conversion method is first applied and the output signal is then differentiated as many times as the wavetable has been integrated. The computational cost of the proposed technique is independent of the pitch-shift ratio. It is shown that the higher-order integrated wavetable technique reduces aliasing more than the first-order technique with a minor increase in computational cost. Quantization effects are analyzed and are shown to become notable at high frequencies, when several integration and differentiation stages are used.

1. INTRODUCTION

Wavetable synthesis [1, 2, 3, 4] is a widely used technique for producing musical tones. Its basic form stores one period of recorded or synthetic sound in a look-up table and retrieves it for playback. Wavetable synthesis enables the playback of arbitrary sounds, including those with a rich harmonic structure, without increasing the computational complexity. It is consequently often more efficient for such sounds than other synthesis techniques, for instance additive synthesis [2, 3].

There are two different strategies for implementing wavetable synthesis with time-varying timbre: 1) wavetable cross-fading [1, 2] in which two wavetables are playing at the same time and 2) multiple wavetable synthesis [5] in which many wavetables are mixed together with their own temporal envelope function. Sampling synthesis is a related method, which uses longer recordings than one cycle, such as an entire sound event, e.g., a dog bark.

Interpolation can be used during the reading of a wavetable for changing the pitch of the stored sound, and thus, a large range of pitches can be generated from just one or from a small set of wavetables [2]. In this context, interpolation is often called pitch shifting, but in this paper we call it sample rate conversion to highlight the change in the spectrum of the sound. Downsampling is used for raising the pitch and upsampling for lowering the pitch of a stored waveform, and since the sample rate of the system is kept unchanged, such alterations of the sampling period lead to a change in the fundamental frequency of the tone.

The sound quality of wavetable synthesis is directly related to the quality of the sample rate conversion algorithm. As in general sample rate conversion problems, two classes of errors occur: imaging and aliasing [6, 7]. Imaging errors occur independent of the conversion ratio. In the case of upsampling (pitch-shifting down), images of the signal spectrum at high frequencies become audible. These errors can be attenuated below a given limit using well-established interpolation techniques or lowpass filtering.

Aliasing artifacts arise when a wavetable is downsampled (pitch-shifting up) and some high-frequency components get shifted above the Nyquist limit, or half of the sampling frequency. Then the image copy of each such high-frequency component is reflected down to the audible band. Aliasing ruins sound quality, because it leads to particularly disturbing audible effects, such as inharmonicity, beating, and heterodyning [8].

A few different approaches have been proposed to reduce aliasing errors in wavetable synthesis. In general, the suppression of aliasing artifacts requires a variable lowpass filter whose cutoff frequency depends on the conversion ratio. Smith and Gossett have proposed a bandlimited interpolation algorithm, which explicitly implements an anti-aliasing filter enabling both the increase and the decrease of the pitch [9]. Massie has described its application to wavetable synthesis [3].

More recently, Geiger suggested a different approach to anti-aliasing in wavetable synthesis [10]. It is based on integrating the signal stored in the wavetable and differentiating the resampled output signal. This technique was inspired by an alias-reduction algorithm for bandlimited synthesis of classical analog waveforms using a differentiated parabolic waveform [11].

This paper generalizes Geiger's scheme by using several integrator and differentiator stages for improved alias-reduction. This work is also motivated by the respective extension of the alias-suppression techniques by high-order differentiated polynomial waveforms [12]. In this paper we consider the new method in the context of basic wavetable synthesis, but it is straightforward to apply it to wavetable cross-fading, multiple wavetable synthesis, and sampling synthesis.

The remainder of this paper is outlined as follows. In Section 2, the interpolation or pitch-shifting needed in wavetable synthesis is interpreted as a sample rate conversion problem. Section 3 introduces the new higher-order integrated wavetable synthesis method and considers FIR differentiator designs. In Section 4, the effects of the number of integration and differentiation stages, the order of the interpolator used, and quantization effects are evaluated. Section 5 concludes this paper.

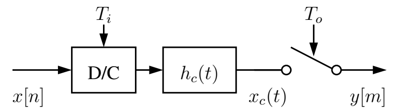


Figure 1: *Hybrid analog/digital model for resampling.*

2. WAVETABLE SYNTHESIS AS A RESAMPLING PROBLEM

2.1. Hybrid Analog/Digital Model for Resampling

Altering the pitch of a waveform by arbitrary ratios corresponds to accessing, i.e., sampling, the wavetable with different sampling frequencies. Therefore, artifacts of wavetable synthesis systems are best characterized in the framework of arbitrary sample rate conversion.

The hybrid analog/digital model [6, 13] is the most convenient model to describe arbitrary resampling methods. It is schematically depicted in Figure 1. A discrete-time input signal $x[n]$, with input sampling period T_i or sampling frequency $f_i = 1/T_i$ is converted into a continuous-time signal by an ideal discrete-to-continuous converter D/C. In the frequency domain, this corresponds to a replication of the original spectrum, shifted by multiples of the input sampling frequency f_i. These replications are referred to as images.

This signal is filtered by a continuous-time combined anti-aliasing/ anti-imaging filter $h_c(t)$, yielding the bandlimited signal $x_c(t)$. Sampling with the output period T_o, corresponding to the output sampling frequency $f_o = 1/T_o$, results in the discrete-time output sequence $y[m]$. In the frequency domain, sampling corresponds to the superposition of spectral replications of $x_c(t)$ that are shifted by multiples of f_o. Overlap between these spectra results in aliasing.

Notwithstanding this continuous-time modeling, resampling filters are generally implemented entirely in the digital domain. The main advantage of the hybrid analog/digital model is that the complete behavior of a resampling algorithm is described by the impulse response of the continuous-time filter $h_c(t)$ or, equivalently, its continuous-time frequency response $H_c(f)$.The purpose of this filter is to attenuate both imaging and aliasing components. The ideal anti-imaging/anti-aliasing filter is a lowpass filter

$$\widehat{H}_c(f) = \begin{cases} T_i, & |f| < f_c \\ 0, & |f| \geq f_c \end{cases} \quad \text{with } f_c = \min\left(f_i/2, f_o/2\right) . \quad (1)$$

Thus, the cutoff frequency f_c depends on the relation between f_i and f_o, which is conveniently described by the conversion ratio $R = f_o/f_i = T_i/T_o$. If $R \geq 1$, corresponding to increasing the sample rate, the cutoff frequency is constant: $f_c = f_i/2$. In case of downsampling, i.e., $R < 1$, f_c varies as a function of the output rate: $f_c = f_o/2$.

Figure 2 depicts the continuous frequency response of a typical resampling system for the downsampling case $R < 1$. The frequency components between $f_o/2$ and $f_i/2$ are part of the baseband of the input signal. Incomplete attenuation of these components results in aliasing. The components above $f_i/2$ cause an incomplete attenuation of signal images.

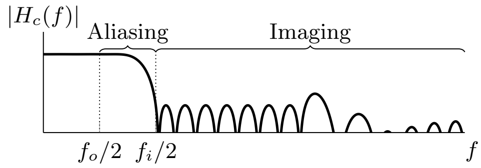


Figure 2: *Continuous frequency response of an ASRC system.*

2.2. Application to Wavetable Synthesis

The application of the hybrid analog/digital model to wavetable synthesis is straightforward, yet it requires some consideration as there is no explicit sample rate conversion visible from the outside. The output sampling frequency f_o of a wavetable synthesizer is the fixed sampling rate of the discrete-time signal processing system, e.g. 44.1 kHz for consumer electronics. Likewise, the discrete-time signal stored in the wavetable has been sampled with a fixed sampling frequency f_i. The conversion ratio used in the resampling process is determined by the ratio between the pitch of the output signal, termed f in the following, and the original pitch of the sampled waveform f_{wt}.

To achieve this output pitch, the read pointer within the wavetable is incremented by f/f_{wt} sample values between successive output samples. This pitch ratio corresponds to a resampling ratio $R = f_{wt}/f$. That is, a decrease of the output pitch with respect to the original pitch corresponds to an increasing conversion ratio, while an increase of the pitch requires a decreasing conversion ratio.

The number of wavetables and the respective length of these tables to represent a single tone are important choices in designing a wavetable synthesis system. They directly affect the fundamental pitch of the signals stored in the wavetables and consequently the used conversion ratios. A particular important question is whether increasing or decreasing conversion ratios are used. This choice is also influenced by the spectral content, such as the contained harmonics, in the stored wavetable signal as well as in the output signal.

According to the sampling theorem, each wavetable must contain spectral components, for instance harmonics, only up to half the sampling frequency, thus having a cutoff frequency $f_c < f_i/2$. If the conversion ratio is increasing ($R > 1$), corresponding to a decreased pitch, the number of harmonics in the output signal is not increased. Instead, the output signal is bandlimited with a cutoff frequency $f_c' = f_c/R$. Thus, higher-order harmonics are missing in the output. To generate sounds with rich spectral contents, it is therefore necessary to use a relatively large number of wavetables to avoid such large resampling ratios [14, 15].

In contrast, increasing the pitch of a sound with respect to the wavetable corresponds to a decreasing conversion ratio. In this case, the wavetable may contain rich spectral contents. Consequently, the resampling algorithm must provide effective lowpass filtering to prevent aliasing, i.e., folding, of signal components above $f_o/2$ into the output signal. In this case, the output signal contains all harmonic components of an ideally bandlimited representation of this sound. Thus, resampling using decreasing conversion ratios is an attractive choice for wavetable synthesis, as it does not impose constraints either on the harmonic contents of the output signal or on the number of required wavetables.

It is noted that the properties of the resampling algorithm is not the only factor that determines the number of required wavetables. Depending on the instrument, the characteristics of the sound event may change too much over the range of desired pitches, thus preventing the use of a single wavetable. As an example, the undesirable rescaling of formants in signals as speech may mandate the use of multiple wavetables [3]. Nonetheless, we consider the availability of efficient anti-aliasing algorithms for wavetable synthesis as an attractive means to reduce the computational resources of sound synthesis applications.

2.3. Conventional Resampling Algorithms

For downsampling, the resampling algorithm requires a lowpass filter with a variable cutoff frequency that is controlled by the conversion ratio. While conventional lowpass filters, either based on a stored set of coefficients or using online designs, are inappropriate in most cases, a few resampling algorithms enable arbitrary downconversion. Smith and Gossett [9] proposed an algorithm that enables arbitrary conversion ratios. It is based on an oversampled, linearly interpolated impulse response of a continuous-time lowpass filter. In case of a sample rate decrease, the impulse response is sampled with a step size that depends on the output period. Hentschel and Fettweis [16] introduced the so-called *transposed Farrow structure*. This algorithm is also based on sampling a continuous impulse response with a step size depending on the conversion ratio, but enables impulse responses that are piecewise polynomials of arbitrary order.

However, application of these algorithms to wavetable synthesis results in an unfavorable computational complexity. As the conversion ratio decreases, corresponding to a significant increase of the pitch, the sampling of the stored impulse responses becomes increasingly dense. Thus, more filter coefficients have to be calculated and applied for a single output sample. As a wavetable synthesizer operates on a fixed external output sampling frequency, this implies a varying computational complexity which is approximately proportional to the pitch change. Such a behavior is unfavorable for most real-time applications.

2.4. Integrated Wavetables for Alias Reduction

Geiger [10] proposed an entirely different approach to anti-alias filtering for table lookup oscillators. It stores an integrated representation of the waveform in the wavetable. The signal obtained from the table lookup is filtered by a discrete-time differentiator. This yields a significant reduction of the aliasing components compared to a simple table lookup. This method is motivated by the differentiated parabolic wave (DPW) algorithm [11] for synthesizing the sawtooth wave, a classical analog waveform. The DPW method evaluates a parabolic waveform, which is obtained from analytic integration of the sawtooth wave, and applies a discrete-time differentiator to gain an output signal with reduced aliasing.

There are two basic choices in the implementation of the integrated wavetable method. First, the interpolation method used in the table lookup determines the imaging error. In [10], the requested output time is rounded to the nearest wavetable location, corresponding to an interpolation filter of order zero. Second, there are different designs for discrete-time differentiator. In the same way as in the DPW method, the integrated wavetable algorithm uses first-order differentiators, which are the simplest differentiator design, but show a considerable magnitude roll-off towards high frequencies.

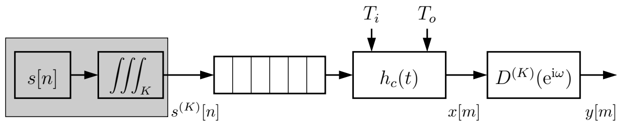


Figure 3: *Signal flow of the higher-order integrated wavetable algorithm.*

3. HIGHER-ORDER INTEGRATED WAVETABLES

This paper proposes the extension of the integrated wavetable algorithms to higher orders of integration and differentiation. Thus, it follows the same idea as the extension of the DPW algorithm proposed in [12] that synthesizes discrete-time representations of analog waveforms using higher-order polynomials. However, the application of this algorithms to discrete-time signals stored in wavetables requires a set of specific decisions and solutions which are described in the following.

The general structure of the higher-order integrated wavetable synthesis algorithm is depicted in figure 3. The discrete-time waveform $s[n]$ is K times integrated to form the integrated signal $s^{(k)}[n]$ which is stored in the wavetable of length L. This process is performed offline and does not contribute to the runtime computational complexity. Without loss of generality, the table lookup process is modeled as an arbitrary resampling process. Thus, it is represented by the continuous-time interpolation filter $h_c(t)$. The resulting resampled signal $x[m]$ is filtered by the discrete-time differentiator component, which differentiates the signal K times, to yield the synthesized output signal $y[m]$.

3.1. Integration of Wavetable Signals

Without loss of generality, we assume that the discrete-time waveform $s[n]$ is obtained from sampling a continuous-time signal $s(t)$ with sampling period T_i. To fulfill the requirements of the sampling theorem, $s(t)$ is assumed to be bandlimited to the Nyquist limit $f = f_i/2 = 2/T_i$. Under these conditions, a discrete-time integration of $s[n]$ is identical to an integration of the continuous-time signal $s(t)$. Let $s^{(k)}[n]$ denote the sequence obtained by a k-fold integration of $s[n]$. This implies $s^{(0)}[n] = s[n]$.

The transfer function of a discrete-time integrator is given by

$$H_i(z) = \frac{1}{1 - z^{-1}} = \sum_{k=0}^{\infty} h[k] z^{-1} \,, \tag{2}$$

with the associated causal impulse response

$$h[k] = 1 \quad 0 \leq k < \infty \,. \tag{3}$$

Thus, a discrete-time integrator can be stated as a convolution

$$s^{(k)}[n] = T_i \sum_{l=0}^{\infty} s^{(k-1)}[n] + c_k \quad k \leq 1 \,, \tag{4}$$

where c_k denotes an integration constant, a free parameter as known from integrating continuous functions. Note that the scaling factor T_i in (4) is required since the differentiator corresponding to this integration operates at a different sampling period. Thus, the wavetable synthesizer is a multirate system, which requires explicit handling of the scaling factors associated with discrete representations of continuous-time signals (e.g. [6, 7]).

By a suitable choice of the integration constants, the infinite summation (4) can be transformed into a finite expression. In [12], the constants used in the integration of continuous waveforms are obtained by solving a linear system. In this paper, we use a different approach that determines the free parameter c_k for each of the successive integrations independently.

As wavetables are typically played back repeatedly, the waveform $s[n]$ can be interpreted as a single cycle of a periodic signal with period L, which can be stated as a condition

$$s[n+L] = s[n] \quad \text{for } n \in \mathbb{Z}. \tag{5}$$

This periodicity should also hold for integrated wavetables, i.e.,

$$s^{(k)}[n+L] = s^{(k)}[n] \quad \text{for } n \in \mathbb{Z}. \tag{6}$$

This condition can be enforced if the waveform $s^{(k)}[n]$ contains no DC component, that is,

$$\sum_{n=0}^{L-1} s^{(k)}[n] = 0. \tag{7}$$

The removal of the DC component is a necessary precondition for the integration, since the integrator has an infinite DC gain, e.g. [17]. As the waveform is known beforehand, it can be performed offline using the integration constant

$$c_k = -\frac{1}{L}\sum_{n=0}^{L-1} s^{(k)}[n]. \tag{8}$$

If the sequence $s^{(k-1)}[n]$ is DC-free, then the infinite summation (4) is reduced to a finite number of terms

$$s^{(k)}[n] = T_i\left(\sum_{l=0}^{n} s^{(k-1)}[n] + c_k\right) \text{ for } k \leq 1,\, 0 \leq n < L. \tag{9}$$

For $s^{(0)}[n]$, removal of the DC component must be performed, if necessary, by adding the negative mean (8) prior to the first integration.

3.2. Design of the Discrete-Time Differentiators

The ideal discrete-time differentiator of order K has the frequency response

$$H_{id}\left(e^{j\omega}\right) = (j\omega)^K \quad \text{for } -\pi \leq \omega \leq \pi. \tag{10}$$

Because this ideal differentiator is not realizable, approximations of this filter are used in practical implementations (see, e.g., [18]). These approximations generally deviate from the ideal frequency response and also induce an implementation delay for causal implementation. In the following, only linear-phase FIR differentiators are considered, which introduce a delay of $N/2$ samples, where N is the filter order.

First-order FIR differentiators are widely used in alias reduction algorithms [11, 10]. Its transfer function is given by $1 - z^{-1}$, possibly subject to a scaling. In case of higher-order differentiators, cascades of first-order differentiators are used, for instance,

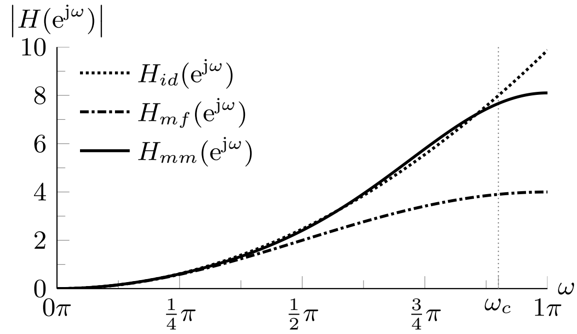


Figure 4: *Magnitude response of discrete-time FIR differentiators of order $K = 2$. Ideal frequency response: $H_{id}(e^{j\omega})$, Maximally flat design: $H_{mf}(e^{j\omega})$, Minimax design: $H_{mm}(e^{j\omega})$. Parameters for minimax design: filter length $5K$, cutoff frequency $\omega_c = 0.9\pi$.*

in [12]. It can be shown that a cascade of K first-order differentiator forms a Kth-order differentiator with filter order K that is maximally flat at $\omega = 0$. Its impulse response is determined by

$$h[k] = \binom{K}{k}(-1)^k \quad \text{for } 0 \leq k \leq K. \tag{11}$$

Due to the property of maximum flatness, the frequency response is very good for low frequencies, but shows a significant roll-off towards higher frequencies, especially for higher differentiator orders. In wavetable synthesis, this leads to amplitude errors, but also an inherent reduction of high-frequency aliasing components [11, 12].

In this paper, we focus on K-th order FIR differentiators that are specifically designed to minimize the approximation error over a wide frequency range. Specifically, we use design methods that minimize the maximum error, which is referred to as the minimax, Chebyshev, or L_∞ norm. Such filters are obtained using convex optimization techniques or variants of the Parks-McClellan algorithm [7]. Here, first-order minimax differentiators are designed by the Matlab **firpm** method with the **'differentiator'** option, and higher-order differentiators are obtained by convolving these basic differentiators. Minimax differentiators are an attractive choice for higher-order integrated wavetables. On the one hand, they virtually avoid errors of passband components. On the other hand, minimax differentiators are more sensitive to aliasing errors than maximally flat designs, as they do not attenuate high-frequency components.

Figure 4 shows the magnitude frequency response of the considered differentiators for order $K = 2$. While the maximally flat differentiator shows a large magnitude error for higher frequencies, the minimax design approximates the ideal frequency response very well up to the cutoff frequency $\omega_c = 0.9\pi$ used in this example.

To recover the correct amplitude, the output of the differentiator must be scaled by a factor T_o^{-K}, where T_o is the sampling period used in the wavetable lookup. This corresponds to the magnitude scaling used in the integration of the wavetable (4).

3.3. The Interpolation Filter

Interpolation filters $h_c(t)$ are used to determine the value of the waveform stored in the wavetable at arbitrary output times. Like in conventional wavetable synthesis algorithms, the main purpose of this filter is the suppress imaging artifacts that otherwise would be aliased into the output signal. The most simple way to perform interpolation is to truncate the requested output time, which is referred to as zero-order hold or drop zero tuning [3]. Rounding the output time to the nearest wavetable index is a similar choice that is used, for instance, in [10]. More sophisticated methods determine the output value by interpolation between two or more wavetable samples adjacent to the requested output time. Without loss of generality, all interpolation methods used for wavetable synthesis can be considered as FIR filters with variable coefficients.

Lagrange interpolation is an attractive choice for the integrated wavetable algorithm, because it has very good quality for lower frequencies, and allows very efficient implementation [19, 20]. Lagrange interpolation of order 1 corresponds to linear interpolation. For resampling applications, only Lagrange interpolators of odd orders are of interest, since even-order interpolators exhibit inferior image suppression.

3.4. Computational Complexity

Two operations determine the computational complexity of the integrated wavetable algorithm: the interpolation filter and the discrete-time differentiator. Integration of the wavetables is performed offline and therefore does not contribute to the computational effort. The number of interpolation operations is determined by the external output sampling frequency, which is constant. The discrete-time differentiator also operates at the external output rate. Thus, the higher-order integrated wavetable algorithms offers a constant computational effort independent of the pitch change. For real-time implementations with fixed resource limits, this is a significant improvement compared to wavetable synthesis using conventional arbitrary downsampling methods, such as those described in Section 2.3.

4. PERFORMANCE EVALUATION

This section provides an evaluation of the performance of the integrated wavetable synthesis algorithm and investigates the influence of several parameters, such as the integration or interpolation orders, on the quality of the output signal.

As test signal, we use the sawtooth wave, a classical analog waveform ubiquitously used in sound synthesis, e.g. [21, 14, 15]. The sawtooth wave is also a common example for alias-suppression algorithms [11, 10, 12]. The wavetable contains a single period of the integrated sawtooth wave, thus the wavetable synthesizer is used in the style of a table lookup oscillator. The analog sawtooth wave has a rich, non-bandlimited spectrum and its harmonics fall off with a relatively low rate of about 6 dB per octave. Consequently, algorithms for synthesizing sawtooth waves require effective anti-aliasing capabilities, thus making it a sensible, critical test signal for the methods considered here.

Throughout this evaluation, a sawtooth wave with a fundamental frequency of $f = 1245\,\text{Hz}$ is used with a sampling rate of $f_i = 44.1\,\text{kHz}$ and a wavetable length of $L = 1024\,\text{samples}$, corresponding to a conversion ratio of $R = f_s/(f \times L) \approx 0.0346$. The signals are synthesized in the time domain. As in [11, 12], the output spectra are obtained by windowing the time-domain signal with a Chebyshev window having a sidelobe attenuation of 120 dB and a discrete Fourier transform. The spectra are compensated for the magnitude loss due to windowing, i.e., the coherent gain [22], which is about 9.4 dB for this Chebyshev window.

4.1. Integration and Interpolation Order

Figure 5 shows the output spectra of the integrated wavetable synthesis algorithm for different integration and interpolation orders.

The ideal bandlimited sawtooth wave is illustrated in Figure 5a. Its harmonics are truncated to half the sampling frequency. Thus, this signal is conveniently represented by the Fourier series

$$s[n] = -\frac{2}{\pi}\sum_{k=1}^{K}\frac{(-1)^{k+1}}{k}\sin\left(\frac{2\pi f n}{f_s}\right) \quad \text{with } K = \left\lfloor \frac{f_s}{2f} \right\rfloor . \tag{12}$$

Figure 5b shows the spectrum generated by a trivial table lookup algorithm where the samples are obtained by rounding the output instants to the nearest wavetable index. This round-to-nearest scheme can be also considered as interpolation of order zero. The resulting spectrum contains are large number of spectral components, caused by aliased harmonics contained in the wavetable.

The spectrum generated from an integrated wavetable of order $K = 1$ combined with round-to-nearest wavetable lookup is shown in Figure 5c. This is basically the algorithm proposed by Geiger [10], although a minimax differentiator is used here instead of a first-order differentiator. Due to the integration/differentiation scheme, the amount of aliasing is significantly reduced compared to the simple table lookup. However, as observed in Figure 5d, an increase of the integration order to $K = 2$ does not increase the quality of the output signal, but increases the amount of aliased components. An analysis reveals that these components are due to the lack of an effective anti-imaging (or interpolation) filter in the resampling process. Replacing the round-to-nearest table lookup with a linear interpolator, which is equivalent to a Lagrange interpolator of order $N = 1$, yields a considerable reduction of aliasing components. This is shown in Figure 5e.

As depicted in Figure 5f, an increase of the integration order to $K = 3$ yields a further reduction of aliasing components. Most notably, the rate of decay of the dominant errors, which result from the first reflection of alias components, increases as a function of the integration order K. At the same time, the minimax differentiator design preserves the amplitudes of the harmonics of the bandlimited sawtooth wave irrespective of the integration order. This is an advantage over existing algorithms based on maximally flat differentiators, e.g. [12] that exhibit a roll-off towards higher frequencies which increases as a function of the integration order.

However, for $K = 4$, new aliasing components are introduced into the output spectrum as depicted in Figure 5g. Again, increasing the quality of the anti-imaging interpolation filter eliminates these artifacts. This is shown in Figure 5h, which shows the output of an integrated wavetable synthesis algorithm of integration order $K = 4$ combined with a Lagrange interpolator of order $N = 3$.

Figures 5i and 5j show the spectra for integrated wavetables of order $K = 5$ and $K = 6$, respectively. It is observed that the increase of the order results in an increased rate of decay of the dominant aliasing components.

These examples show that integrated wavetable synthesis with higher integration orders may yield significant performance improvements over trivial table lookup algorithms or first-order integrated wavetables. However, as the integration order increases, effective interpolation filters become mandatory in order to exploit

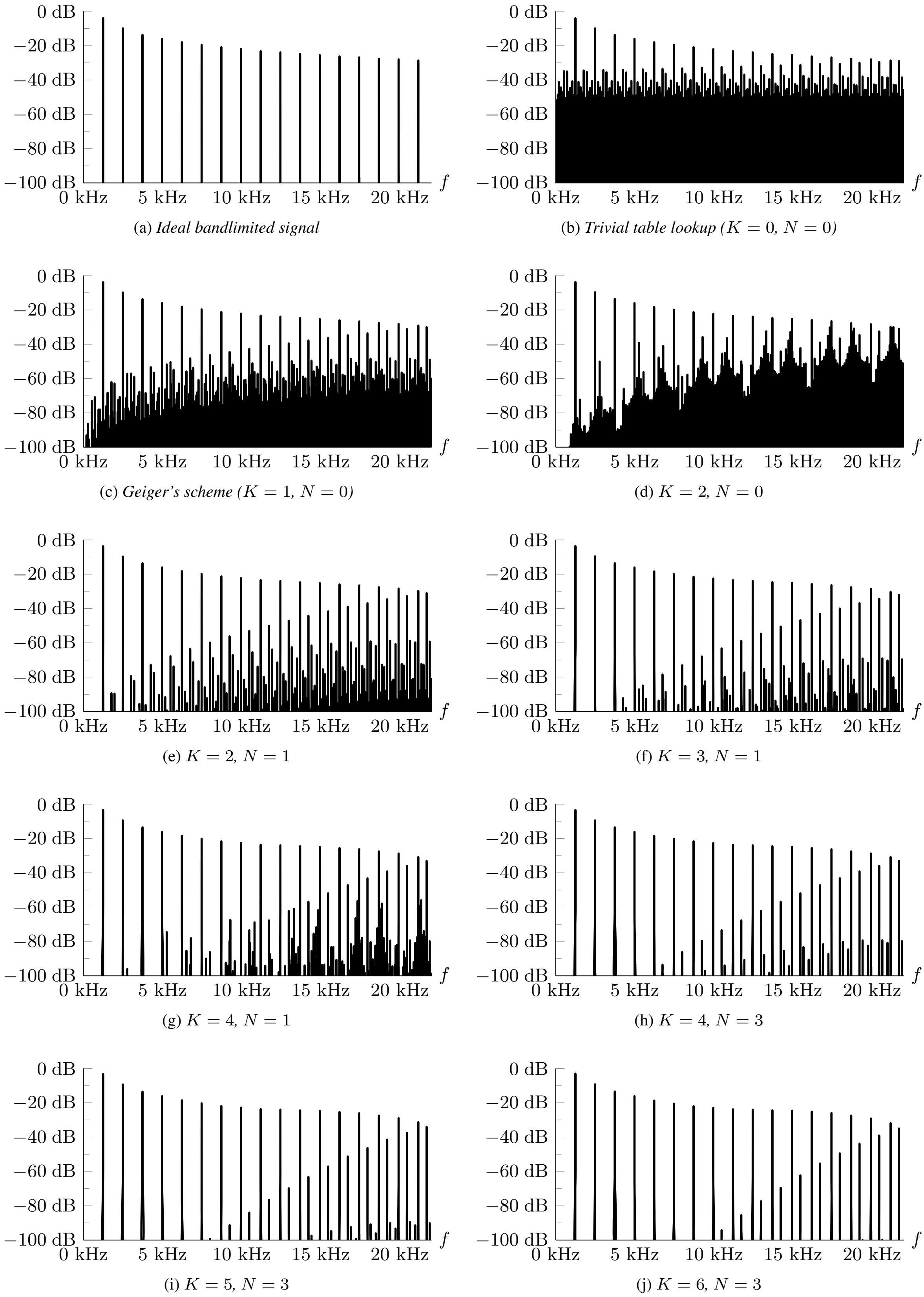


Figure 5: *Discrete-time spectra of integrated wavetable synthesis. Sawtooth wave as test signal, fundamental frequency $f = 1245\,\mathrm{Hz}$, sampling frequency $f_i = 44.1\,\mathrm{kHz}$, discrete-time differentiators designed according to the minimax (L_∞) norm. K denotes the order of integration/differention, and N is the order of the Lagrange interpolator.*

this potential performance gain. The quality (or order) of the interpolation filter should be matched to the integration order. While a higher-order integrated wavetable requires a good interpolation method, increasing the quality of interpolation does not improve the image-suppression properties of a low-order integrated wavetable.

4.2. Quantization and Roundoff Errors

The structure of the integrated wavetable synthesizer makes it susceptible to quantization and arithmetic roundoff errors. From the frequency-domain viewpoint, the basic working principle is an attenuation of high-frequency components by the integrator followed by the restoration of these components by the discrete-time differentiator. Thus, depending on the order of integration, the differentiator provides large gains of high frequency components. Consequently, errors in these frequency regions, which are introduced by quantization of the wavetables or arithmetic errors in the interpolation process, are amplified in the output signal.

This section analyzes the influence of a quantized wavetable on the quality of the output signal. To avoid the effect of magnitude scalings of the wavetable, this analysis operates on a floating-point algorithm. As in the classical quantization model for a uniform quantizer, e.g. [23, 7], the quantization noise is modeled as a uniformly distributed noise signal. The quantization step size Q of the wavetable signal $x[n]$ is given by

$$Q = \frac{\max(x[n]) - \min(x[n])}{2^b}, \tag{13}$$

where b denotes the quantization wordlength in bits. Then, the quantization is modeled as additive noise $e[n]$ that is uniformly distributed in the interval $[-Q/2, Q/2]$ [23]

$$x_q[n] = x[n] + e[n] \quad \text{with } e[n] = \frac{1}{Q}\text{rect}(e/Q). \tag{14}$$

The effects of quantizing the integrated wavetables on the spectrum of the output signals are demonstrated in Figure 6. For this analysis, the sawtooth wave introduced in the preceding section with a fundamental frequency of $f = 1245$ Hz and a FIR differentiator designed with respect to a L_∞ (minimax) error norm is used. Figure 6a depicts the spectrum obtained from a third-order integrated wavetable with 3rd-order Lagrange interpolation and quantization corresponding to a wordlength of 24 bit. It is observed that this quantization introduces some additional high-frequency noise to the output signal compared to the unquantized version (Figure 5h). Figures 6b shows the spectrum for a wordlength of 20 bit. It is apparent that the power of the deteriorating noise increases relatively rapidly as the wordlength decreases.

To observe the influence of the integration order, Figure 6c shows the spectrum of a fifth-order integrated wavetable algorithm quantized to 24 bit. While the quality of the unquantized algorithm (Figure 5i) is superior to the corresponding fourth-order wavetable (Figure 5h), this relation reverses even for very small levels of quantization noise. This is observed by comparing Figures 6a and 6c. Decreasing the wordlength to 20 bit results in a disproportionate increase of the noise, as shown in Figure 6d.

Thus, the sensitivity to quantization errors increases with the integration order. As conjectured above, this effect is due to the large high-frequency amplification of higher-order discrete-time differentiators. This implies that errors due to finite-accuracy arithmetic or imperfections of the interpolation algorithms are amplified in the same way.

While the magnitude of the quantization effects depends on the specific differentiator design, the underlying problem is inherent to the integrated wavetable algorithm. Additional tests with maximally flat differentiators have shown a reduced level of quantization noise at the expense of an increased roll-off of higher-order harmonics, but the qualitative effect remains.

5. CONCLUSIONS

In this paper, we considered the application of higher-order integrated wavetables to provide efficient anti-aliasing for wavetable synthesis, including table lookup oscillators. Wavetable synthesis algorithms with decreasing conversion ratios are an attractive choice, as they enable the synthesis of spectrally rich sounds from a small set of wavetables. However, the use of such conversion ratios requires effective anti-aliasing techniques.

We show that integrated wavetables with higher orders of integration can achieve significantly improved anti-aliasing compared to first-order integration. This technique is particularly interesting because, unlike existing anti-aliasing solutions known from sample rate conversion, its computational effort is independent of the conversion ratio. Moreover, the use of discrete-time differentiators designed according to the L_∞ norm avoids amplitude errors of the harmonics of the bandlimited signal, avoiding the magnitude roll-off often associated with algorithms based on cascades of first-order differentiators.

However, two important issues must be considered when implementing higher-order integrated wavetables. First, unlike the first-order case, an effective interpolation filter must be used, and the order of the interpolation filter must be adapted to the order of integration. Second, the combination of higher-order integration followed by a higher-order differentiator may show an increased sensitivity to quantization effects in the stored wavetables as well as the arithmetic accuracy. Thus, the use of higher-order integrated wavetables requires careful trade-offs between the order of integration, the quality of the interpolation filter, and the wordlength and the numerical accuracy used in the implementation.

Audio examples and supplementary material related to this work are available online at `http://www.idmt.fraunhofer.de/andreasfranck/dafx2012hoiws`.

6. REFERENCES

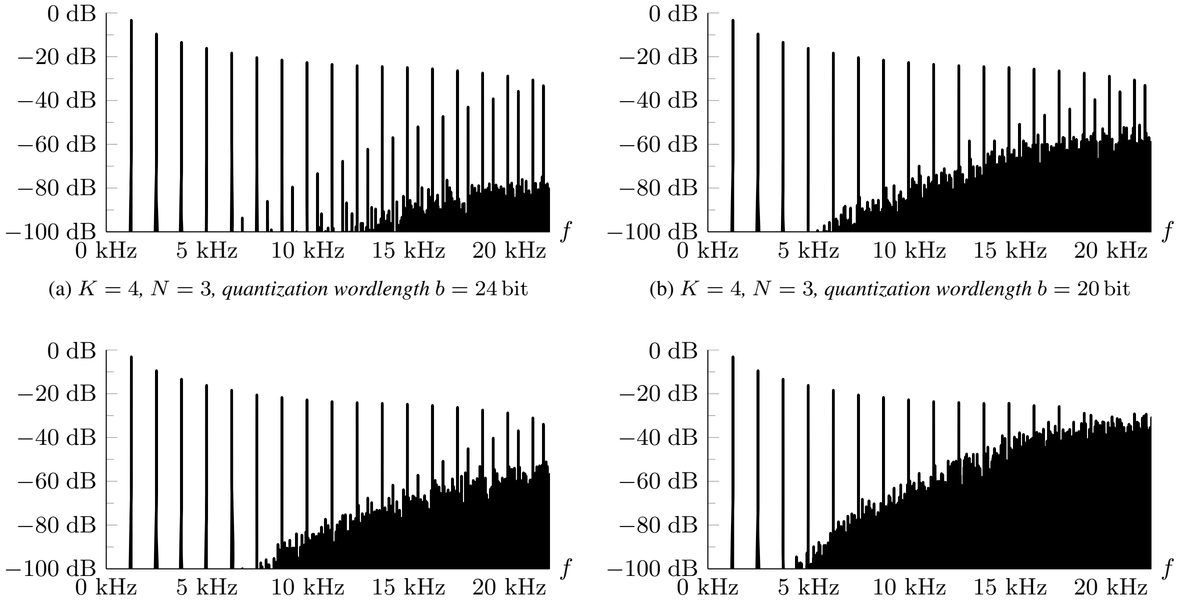
[1] C. Roads, *The Computer Music Tutorial*, MIT Press, Cambridge, MA, 1996.

[2] R. Bristow-Johnson, "Wavetable synthesis 101, a fundamental perspective," in *Proc. 101th AES Convention*, Los Angeles, CA, Nov. 1996.

[3] D. C. Massie, "Wavetable sampling synthesis," in *Applications of digital signal processing to audio and acoustics*, M. Kahrs and K. Brandenburg, Eds. Kluwer, Boston, MA, 1998.

[4] R. C. Maher, "Wavetable synthesis strategies for mobile devices," *J. Audio Eng. Soc.*, vol. 53, no. 3, pp. 205–212, Mar. 2005.

[5] A. Horner, J. Beauchamp, and L. Haken, "Methods for multiple wavetable synthesis of musical instrument tones," *J. Audio Eng. Soc.*, vol. 41, no. 5, pp. 336–356, May 1993.



(a) $K = 4$, $N = 3$, *quantization wordlength* $b = 24\,\mathrm{bit}$

(b) $K = 4$, $N = 3$, *quantization wordlength* $b = 20\,\mathrm{bit}$

(c) $K = 5$, $N = 3$, *quantization wordlength* $b = 24\,\mathrm{bit}$

(d) $K = 5$, $N = 3$, *quantization wordlength* $b = 20\,\mathrm{bit}$

Figure 6: *Effects of quantization errors in integrated wavetables on the output spectrum. Sawtooth wave, fundamental frequency* $f = 1245\,\mathrm{Hz}$*, wavetable length 1024 samples , sampling frequency* $f_i = 44.1\,\mathrm{kHz}$*, minimax differentiator design.*

[6] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1983.

[7] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Pearson, Upper Saddle River, NJ, 3rd edition, 2010.

[8] V. Välimäki and A. Huovilainen, “Antialiasing oscillators in subtractive synthesis,” *IEEE Signal Processing Mag.*, vol. 24, no. 2, pp. 116–125, Mar. 2007.

[9] J. O. Smith and P. Gossett, “A flexible sampling-rate conversion method,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP’84)*, San Diego, CA, Mar. 1984, vol. 9, pp. 19.4.1–19.4.4.

[10] G. Geiger, “Table lookup oscillators using generic integrated wavetables,” in *Proc. 9th Int. Conf. Digital Audio Effects (DAFx-06)*, Montreal, Canada, Sept. 2006, pp. 169–172.

[11] V. Välimäki, “Discrete-time synthesis of the sawtooth waveform with reduced aliasing,” *IEEE Signal Processing Lett.*, vol. 12, no. 3, pp. 214–217, Mar. 2005.

[12] V. Välimäki, J. Nam, J. O. Smith, and J. S. Abel, “Alias-suppressed oscillators based on differentiated polynomial waveforms,” *IEEE Trans. Audio, Speech, Language Processing*, vol. 18, no. 4, pp. 786–798, May 2010.

[13] J. Vesma and T. Saramäki, “Polynomial-based interpolation filters—Part I: Filter synthesis,” *Circuits Systems Signal Process.*, vol. 26, no. 2, pp. 115–146, Apr. 2007.

[14] J. Wright, “Synthesising band limited waveforms using wavetables,” [Online] `http://www.musicdsp.org/files/bandlimited.pdf`, 2000.

[15] P. Burk, “Band limited oscillators using wave table synthesis,” in *Audio Anecdotes II: Tools, Tips, and Techniques for Digital Audio*, K. Greenebaum and R. Barzel, Eds., pp. 37–53. A.K. Peters, Wellesley, MA, 2004.

[16] T. Hentschel and G. Fettweis, “Continuous-time digital filtering for sample-rate conversion in reconfigurable radio terminals,” in *Proc. European Wireless*, Dresden, Germany, Sept. 2000, pp. 55–59.

[17] E. Brandt, “Hard sync without aliasing,” in *Proc. Int. Computer Music Conf.*, Havana, Cuba, 2001, pp. 332–335.

[18] C. S. Dutta Roy and B. Kumar, “Digital differentiators,” in *Handbook of Statistics: Signal Processing and its Applications*, vol. 10, chapter 6, pp. 159–205. Elsevier, Amsterdam, Netherlands, 1993.

[19] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, “Splitting the unit delay: Tools for fractional delay filter design,” *IEEE Signal Processing Mag.*, vol. 13, pp. 30–60, Jan. 1996.

[20] A. Franck, “Efficient algorithms and structures for fractional delay filtering based on Lagrange interpolation,” *J. Audio Eng. Soc.*, vol. 56, no. 12, pp. 1036–1056, Dec. 2008.

[21] T. Stilson and J. O. Smith, “Alias-free digital synthesis of classic analog waveforms,” in *Proc. Int. Computer Music Conf.*, Hong Kong, 1996, pp. 332–335.

[22] F. J. Harris, “On the use of windows for harmonic analysis with the discrete Fourier transform,” *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, Jan. 1978.

[23] U. Zölzer, *Digital Audio Signal Processing*, Wiley, Chichester, UK, 2008.