Rounding corners with blamp

ROUNDING CORNERS WITH BLAMP

Fabián Esqueda[*] , *Vesa Välimäki*

Dept. of Signal Processing and Acoustics
Aalto University
Espoo, Finland
fabian.esqueda@aalto.fi

Stefan Bilbao [†]

Acoustics and Audio Group
University of Edinburgh
Edinburgh, UK
s.bilbao@ed.ac.uk

ABSTRACT

The use of the bandlimited ramp (BLAMP) function as an antialiasing tool for audio signals with sharp corners is presented. Discontinuities in the waveform of a signal or its derivatives require infinite bandwidth and are major sources of aliasing in the digital domain. A polynomial correction function is modeled after the ideal BLAMP function. This correction function can be used to treat aliasing caused by sharp edges or corners which translate into discontinuities in the first derivative of a signal. Four examples of cases where these discontinuities appear are discussed: synthesis of triangular waveforms, hard clipping, and half-wave and full-wave rectification. Results obtained show that the BLAMP function is a more efficient tool for alias reduction than oversampling. The polynomial BLAMP can reduce the level of aliasing components by up to 50 dB and improve the overall signal-to-noise ratio by about 20 dB. The proposed method can be incorporated into virtual analog models of musical systems.

1. INTRODUCTION

Nonlinear audio processing introduces frequency components that are not present in the original input signal. When the frequencies of these components exceed half the sampling rate, or Nyquist limit, they are reflected into the baseband through aliasing [1, 2]. Aliasing distortion can cause audible disturbances, such as beating and inharmonicity, and affect the overall performance of an audio system [3, 1]. In fields such as virtual analog modeling of musical systems, the aim is to emulate the harmonic distortion introduced by analog systems while avoiding aliasing distortion [4, 2]. Therefore, it is of great importance to find efficient algorithms that minimize its effect.

A well known previous approach to avoiding aliasing in nonlinear audio processing is oversampling [4, 5, 6, 7]. In oversampling, the input signal is upsampled prior to processing (typically by a low factor) and downsampled back to the original rate after processing. This approach requires access to the original unprocessed signal and, ideally, some knowledge on the order of the nonlinear processing stage. In oversampling, the added computational costs will depend on the oversampling factor and order of the filters used for its implementation. Other techniques available to avoid aliasing in nonlinear processing include the harmonic mixer [8] and reducing the order of the nonlinearity [5]. The latter approach can also be used in distortion synthesis of classical oscillator waveforms [9].

Signal processing operations that introduce discontinuities in the waveform of a signal or its derivatives are major sources of aliasing. These discontinuities require infinite bandwidth to be represented in the digital domain. Attempting to sample them trivially will inevitably introduce aliasing distortion [10, 11]. When a discontinuity is introduced in the first derivative of a signal, a sharp edge or corner is introduced in the actual waveform.

Previous work on alias-reduced synthesis of oscillator waveforms has introduced the concept of quasi-bandlimiting discontinuities found in the waveform [12, 11, 13]. This work further explores this idea by presenting the use of the bandlimited ramp (BLAMP) function to treat any discontinuities found in the first derivative of a signal. This is achieved by quasi-bandlimiting the corners found in the waveform of a signal. The BLAMP function was originally proposed for synthesis of alias-free triangular waveforms [14, 11]. We derive a polynomial approximation of the BLAMP function, or polyBLAMP, which leads to an efficient implementation. Four examples of audio-specific scenarios where corners appear in the waveform of a signal are discussed: synthesis of triangular oscillator waveforms, hard clipping, half-wave and full-wave rectification. Results obtained demonstrate that the polyBLAMP method can effectively reduce the aliasing caused by these corners and the discontinuities they introduce.

This paper is organized as follows. Section 2 derives the analytical form of the BLAMP correction function. Section 3 discusses the computational costs of the BLAMP function and presents the derivation of its polynomial approximation. In Section 4, the performance of the method is evaluated by considering four applications. Finally, concluding remarks appear in Section 5.

2. INTEGRATED BANDLIMITED FUNCTIONS

Analog signals with discontinuities in their waveforms have infinite frequency content and must be bandlimited to less than half the Nyquist limit prior to sampling to avoid aliasing. The full-band nature of discontinuous signals can be observed, for instance, by considering the Fourier series (FS) expansion of a rectangular pulse. The FS for this signal consists of an infinite sum of odd sinusoidal components, with the amplitude of the k^{th} harmonic defined as $1/k$ of the first harmonic or fundamental.

We can model a single discontinuity in the continuous-time domain using the Heaviside unit step function, which is defined as

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0, \end{cases} \tag{1}$$

where t is time. This function jumps from 0 to 1 at $t = 0$ and is used in system analysis to measure the step response of a system.

In this work we are concerned with aliasing caused by discontinuities occurring not in the waveform of a signal, but in its first

[*] The work of F. Esqueda is funded by the Aalto ELEC Doctoral School.

[†] The work of S. Bilbao is supported by the European Research Council, under grant number ERC-StG-2011-279068-NESS.

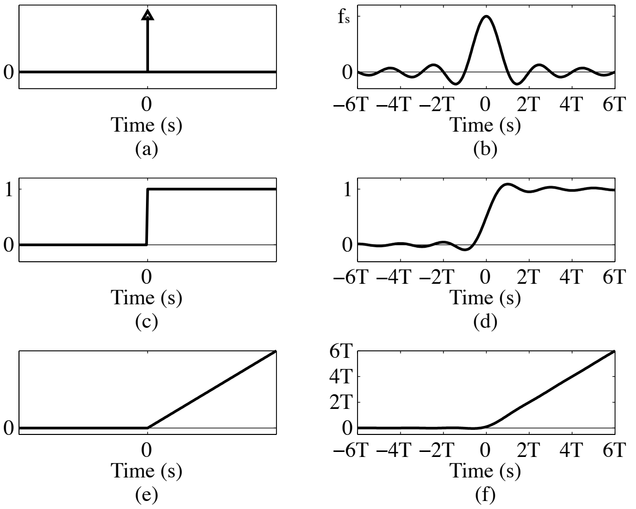

Figure 1: *Time-domain waveform of the (a) impulse, (b) bandlimited impulse, (c) Heaviside unit step, (d) BLEP, (e) ramp, and (f) BLAMP functions. Parameter T is the sampling period.*

derivative. Therefore, we model a discontinuity in the first derivative by evaluating the integral of the Heaviside function (1) as

$$\int_{-\infty}^{t} u(\tau)\mathrm{d}\tau = tu(t) = r(t). \tag{2}$$

Equation (2) is known in the literature as the ramp function [15]. It is characterized by the sharp corner that occurs at $t = 0$ when the function starts to linearly increase.

Signals with discontinuities in their first derivative also have infinite frequency content. One example of this is the triangular waveform, which can be represented using the FS as an infinite sum of odd sinusoidal components. In this series, harmonics decay at a steeper rate than in the case of the rectangular pulse, with the amplitude of the k^{th} harmonic given by $1/k^2$ with respect to the fundamental. In the digital domain, this steeper decay means that the level of aliasing introduced during trivial, non-bandlimited sampling of a sharp corner will be lower than that introduced in discontinuous signals. Nevertheless, when working at audio rates (e.g. 44.1 kHz) the effects of this aliasing may still be perceived, particularly at high fundamental frequencies. Section 4.1 further expands on the issue of aliasing in triangular waveforms.

In order to derive a correction function that can reduce the aliasing introduced by trivial sampling of sharp corners we need to take one step back and evaluate the derivative of the Heaviside unit step function with respect to time. This derivative is defined as the Dirac delta function [15], so that

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = \delta(t). \tag{3}$$

Figures 1(a), 1(c) and 1(e) show the continuous-time domain waveforms for the Dirac delta (represented by a single impulse), Heaviside unit step, and ramp functions, respectively. From these waveforms it should become evident that the size of the discontinuity introduced in the first derivative by a sharp corner will depend on the slope of the signal at this corner.

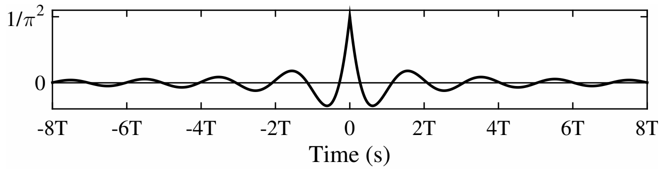

Figure 2: *BLAMP residual function is the difference of the BLAMP and trivial ramp functions. Cf. Figs. 1(f) and 1(e).*

The delta function has a flat unity spectrum, so its bandlimited form can then be obtained by evaluating the inverse Fourier Transform (FT) of an ideal brickwall lowpass filter [16], which yields

$$h^{(0)}(t) = f_\mathrm{s}\mathrm{sinc}(f_\mathrm{s}t), \tag{4}$$

where f_s represents the sampling rate and $\mathrm{sinc}(x) = \sin(\pi x)/\pi x$. Figure 1(b) shows the waveform for this expression.

Following our previous logic, we can derive a bandlimited expression for the ramp function (2) by integrating (4) twice. The integral of the bandlimited unit impulse yields the closed-form equation for the bandlimited step (BLEP) function [11], expressed as

$$h^{(1)}(t) = \frac{1}{2} + \frac{1}{\pi}\mathrm{Si}(\pi f_\mathrm{s}t), \tag{5}$$

where $\mathrm{Si}(x)$ is the sine integral, defined as $\mathrm{Si}(x) = \int_0^x \frac{\sin(t)}{t}\mathrm{d}t$. Previous work in the field of alias-free synthesis of rectangular and sawtooth oscillators has focused on using this expression to bandlimit the inherent discontinuities of these waveforms [17, 11, 18]. Figure 1(d) shows the shape for this function.

Moving on, (5) can be integrated once more using integration by parts, yielding

$$h^{(2)}(t) = t\left[\frac{1}{2} + \frac{1}{\pi}\mathrm{Si}(\pi f_\mathrm{s}t)\right] + \frac{\cos(\pi f_\mathrm{s}t)}{\pi^2 f_\mathrm{s}} \tag{6}$$

$$= th^{(1)}(t) + \frac{\cos(\pi f_\mathrm{s}t)}{\pi^2 f_\mathrm{s}}. \tag{7}$$

This equation gives the closed form expression for the BLAMP function with unit slope, and its shape is shown in Fig. 1(f). At first glance, Figs. 1(e) and 1(f) may appear indistinguishable. However, computing the difference between (7) and (2) quickly proves otherwise, as shown in Fig. 2. This function is referred to as the BLAMP residual function in this study.

In the discrete-time domain, the BLAMP residual can be used to reduce the aliasing caused by a discontinuity in the first derivative by adding it to every sharp edge in the waveform. The first step of this process involves centering the residual function at the exact points in time where the edges occur and sampling it at the nearest integer sample points. These sampled values must then be scaled by the magnitude and direction (i.e. rising or falling edge) of the discontinuity introduced in the first derivative of the signal. The magnitude parameter, as previously stated, can be computed from the slope of the signal at the edge.

3. POLYNOMIAL BLAMP APPROXIMATION

The analytic expression for the BLAMP residual function has two limitations. First, its implementation is computationally expensive due to the presence of the sine integral function. Secondly, the

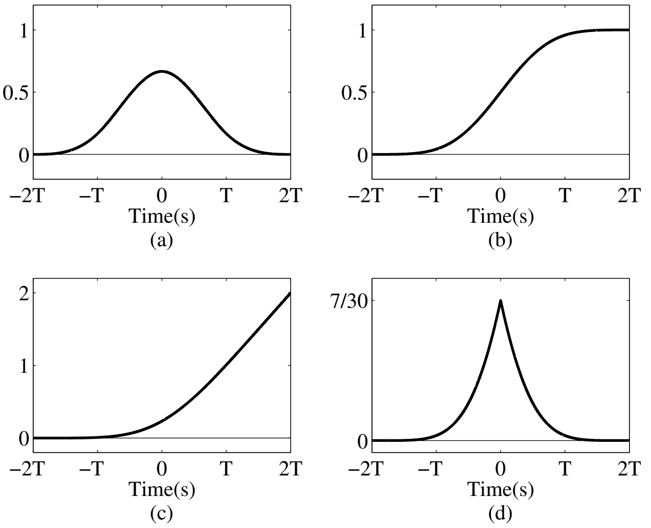

Figure 3: *(a) Cubic B-spline basis function, (b) its first integral, (c) its second integral or polyBLAMP approximation and (d) the polyBLAMP residual, i.e., the difference between the polyBLAMP and the trivial ramp functions.*

function does not have finite support, it does not vanish. Therefore, its truncation to a finite interval introduces small discontinuities which will produce further aliasing. Both issues can be addressed by storing a windowed precomputed portion of the function in a lookup table. This approach is sometimes used in practical implementations of the BLEP method [17, 12, 18]. In general, the effectiveness and efficiency of a table-based implementation will depend on several variables, including table size, interpolation method used (if any) and type of window.

In this work, we instead propose the use of a B-spline polynomial approximation of the BLAMP function (the polyBLAMP) which can be implemented with minimal computational costs. This polyBLAMP function can correct four samples, two on each side of every sharp edge in the waveform. The four-point polyBLAMP function is derived by first approximating the bandlimited impulse (3) as a piecewise polynomial using the coefficients for the third-order B-spline basis function and following the same steps detailed in the previous section [i.e. integrate twice and subtract (2)]. B-spline interpolating polynomials have been used in this study due to their steep spectral decay which makes them suitable for antialiasing applications [19, 11].

Before moving on to the derivation of the polyBLAMP, we first consider that, in practice, the exact sample points at which the sharp edges occur in a signal (i.e. the points where the derivative of the signal is discontinuous) will most likely not coincide with the sampling intervals of the system and must be estimated. In the four-point case, the process of centering the correction function around a set of four samples can be seen as equivalent to delaying it by $D = D_{\text{int}} + d$ samples, where $D_{\text{int}} = 1$, and $d \in [0, 1)$ is the fractional delay.

The coefficients for the B-spline basis function can be expressed in terms of delay D using the four polynomials shown at the top of Table 1. These polynomial coefficients are derived via the iterative convolution of a rectangular pulse, and the resulting waveform can be seen in Fig. 3(a). From this figure, that loosely resembles the central lobe of the bandlimited impulse [see Fig. 1(b)], we can observe the characteristic bell-shaped curve of B-spline interpolators. Integrating this basis function once yields the B-spline polynomial form of the BLEP function (known as the polyBLEP [12, 11]), and integrating once more results in the four-point polyBLAMP function [20]. The polynomials for these two functions and their corresponding waveforms are shown in Table 1, and Figs. 3(b) and 3(c), respectively. Finally, the bottom four rows of Table 1 show the piecewise polynomial coefficients for the polyBLAMP residual evaluated by substituting $D = d + 1$ and computing difference between the polyBLAMP and the ramp function. A two-point version of the polyBLAMP function can be found in [21]. However, due to its superior performance, this work focuses solely on the four-point method.

Table 1: Third-order B-spline basis functions, its first integral (polyBLEP), its second integral (polyBLAMP), and polyBLAMP residual ($1 \leq D < 2$ and $0 \leq d < 1$) [20].

Span	Third-order B-spline basis function
$[-2T, -T]$	$D^3/6 - D^2/2 + D/2 - 1/6$
$[-T, 0]$	$-D^3/2 + 2D^2 - 2D + 2/3$
$[0, T]$	$D^3/2 - 5D^2/2 + 7D/2 - 5/6$
$[T, 2T]$	$-D^3/6 + D^2 - 2D + 4/3$
Span	First integral: Four-point polyBLEP
$[-2T, -T]$	$D^4/24 - D^3/6 + D^2/4 - D/6 + 1/24$
$[-T, 0]$	$-D^4/8 + 2D^3/3 - D^2 + 2D/3 - 1/6$
$[0, T]$	$D^4/8 - 5D^3/6 + 7D^2/4 - 5D/6 + 7/24$
$[T, 2T]$	$-D^4/24 + D^3/3 - D^2 + 4D/3 + 1/3$
Span	Second integral: Four-point polyBLAMP
$[-2T, -T]$	$D^5/120 - D^4/24 + D^3/12 - D^2/12 + D/24 - 1/120$
$[-T, 0]$	$-D^5/40 + D^4/6 - D^3/3 + 2D^2/3 - D/6 + 1/30$
$[0, T]$	$D^5/40 - 5D^4/24 + 7D^3/12 - 5D^2/12 + 7D/24 - 1/24$
$[T, 2T]$	$-D^5/120 + D^4/12 - D^3/3 + 2D^2/3 + D/3 + 4/15$
Span	Four-point polyBLAMP residual
$[-2T, T]$	$d^5/120$
$[-T, 0]$	$-d^5/40 + d^4/24 + d^3/12 + d^2/12 + d/24 + 1/120$
$[0, T]$	$d^5/40 - d^4/12 + d^2/3 - d/2 + 7/30$
$[T, 2T]$	$-d^5/120 + d^4/24 - d^3/12 + d^2/12 - d/24 + 1/120$

Expressing the four-point polyBLAMP residual function in terms of the fractional delay d required to center it around a sharp edge simplifies the procedure of sampling it at the four neighboring sample points. Therefore, parameter d must be estimated to a certain degree of accuracy. First, we consider $s[n]$ to be the discrete-time signal to be antialiased, where $n \in \mathbb{Z}_{\geq 0}$ is the sample index. Next, we define n_a and n_b as the sample indices of the signal before and after an edge, i.e. the *corner boundaries*. For every edge in the waveform, sample points $n_\text{a} - 1$, n_a, n_b and $n_\text{b} + 1$ will be processed by the algorithm. The aim is to fit a polynomial of the form $f(D) = aD^3 + bD^2 + cD + e$ to the signal $s[n]$ at these four points. Lagrange interpolation can be used to find the closed form expressions for coefficients a, b, c, and e. Since the data points are evenly spaced, these coefficients can be written as

$$\begin{aligned}
a &= -\tfrac{1}{6}s[n_\text{a} - 1] + \tfrac{1}{2}s[n_\text{a}] - \tfrac{1}{2}s[n_\text{b}] + \tfrac{1}{6}s[n_\text{b} + 1] \\
b &= s[n_\text{a} - 1] - \tfrac{5}{2}s[n_\text{a}] + 2s[n_\text{b}] - \tfrac{1}{2}s[n_\text{b} + 1] \\
c &= -\tfrac{11}{6}s[n_\text{a} - 1] + 3s[n_\text{a}] - \tfrac{3}{2}s[n_\text{b}] + \tfrac{1}{3}s[n_\text{b} + 1] \\
e &= s[n_\text{a} - 1]. \qquad (8)
\end{aligned}$$

After fitting the polynomial, the next step is to obtain the intersection of this curve with ρ, the corner parameter. The value of ρ will depend on the particular application. For instance, for corners caused by rectification we need to find the zero-crossings of the polynomial, thus $\rho = 0$. Further details on how this parameter is adjusted for each application are given in Sec. 4. This inverse interpolation problem is equivalent to solving the following equation for D:

$$aD^3 + bD^2 + cD + e - \rho = 0. \tag{9}$$

A solution can be estimated using Newton-Raphson's (NR) iterative method [20], defined as

$$D_{q+1} = D_q - \frac{f(D_q)}{f'(D_q)}, \tag{10}$$

where $q = 0, 1, 2, ..., Q - 1$, and Q is the number of iterations required for the ratio $f(D_q)/f'(D_q)$ to become small enough to be neglected, and D_0 is an initial guess [22]. Since the solution to (9) will range between [1,2) due to the restriction on D, an appropriate initial guess would be $D_0 = 1.5$.

We can then estimate the point where the discontinuity in the first derivative occurs as

$$D_{q+1} = D_q - \frac{aD_q^3 + bD_q^2 + cD_q + e - \rho}{3aD_q^2 + 2bD_q + c}. \tag{11}$$

The resulting value D_Q represents the fractional delay associated with a sharp edge or corner. The slope at this point is obtained as a byproduct of the NR method, which is given as

$$\mu(D_Q) = 3aD_Q^2 + 2bD_Q + c. \tag{12}$$

Finally, the value of d can be computed as $d = D_Q - 1$. This represents an estimated sharp edge at $n_a + d$, i.e. $s[n_a + d] = \rho$.

4. POLYBLAMP APPLICATIONS

This section shows how the polyBLAMP correction method can be applied for antialiasing in four audio applications where discontinuities appear in the first derivative of the signal waveform.

4.1. Alias-Free Triangular Oscillator

The first application considered in this study is the synthesis of antialiased triangular oscillator waveforms. This type of geometric waveform is commonly used as a source signal in subtractive synthesis due to its rich harmonic content. As mentioned in Sec. 2, the triangular waveform is composed of odd harmonics only and has the perceptual attribute of being *smoother* to the ears than sawtooth and rectangular waveforms. This characteristic can be attributed to the steep spectral decay of its harmonics, which decay at a rate of about −12 dB per octave (the spectrum of sawtooth and rectangular waveforms decays at a rate of about −6 dB per octave) [3]. This steep spectral decay rate is associated with the discontinuity in its first derivative [10].

Fig. 4(a) shows the continuous-time domain waveform for four periods of a triangular oscillator with fundamental frequency f_0 and period $T_0 = 1/f_0$. Computing the first derivative of this signal results in the square signal shown in Fig. 4(b). The peak-to-peak amplitude of this resulting waveform is determined by 2μ, where μ is the absolute value of the slope of the rising and falling portions of the signal. Since the slope of the falling section is the negative of the slope of the rising section, the signal in Fig. 4(a) is formally known as the symmetrical triangular waveform. Finally, evaluating the derivative of Fig. 4(b) yields the alternating impulse train shown in Fig. 4(c).

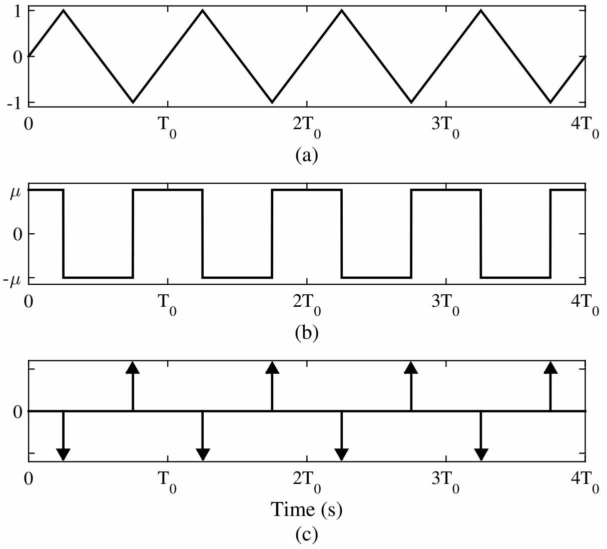

Figure 4: *(a) Continuous-time triangular waveform with arbitrary fundamental frequency f_0, (b) its first and (c) second derivatives.*

In theory, an alias-free discrete-time implementation of the triangular waveform can be achieved by replacing the impulses seen in Fig. 4(c) with (4) (note that the polarity of every second pulse has to be inverted) and integrating the function twice [16]. Due to the infinite nature of the bandlimited impulse (4) and the difficulties associated with performing the double integration, this approach is impractical. Instead, we propose adding the four-point polyBLAMP residual function to the actual waveform at the exact points where the impulses would appear in the second derivative, i.e. at the corners. The residual function has to be scaled by 2μ and inverted for positive edges of the waveform [see Fig. 4(c)].

Since this is a synthesis application of the polyBLAMP method, there is no need to estimate the fractional points at which the edges occur or the slope of the signal at those points; these two parameters are readily available. To implement the proposed method we first synthesize a trivial triangular waveform using a bipolar modulo counter $\phi[n]$ that switches its direction every time it reaches $+1$ or -1 [23]. The fractional delay d associated with each corner can be computed every time the polarity of the counter is inverted as

$$d = (T_\phi - \phi[n])/T_\phi, \tag{13}$$

where $T_\phi = 2f_0/f_s$ is the phase step size. The slope parameter is given by $|\mu| = 2T_\phi$.

Fig. 5 shows the waveform and spectrum for a 1661-Hz (MIDI note A6) trivial triangular waveform sampled at 44.1 kHz without and with four-point polyBLAMP correction. These results show that the corrected signal is virtually alias-free below approx. 12 kHz. Due to the inherent steep spectral decay of B-spline polynomials, the polyBLAMP method introduces a frequency droop of approx. −12 dB. This droop begins after the 10 kHz mark and, if necessary, can be compensated using a shelving EQ filter [11]. However, due to it only affecting high frequencies, it can be neglected in most applications. One convenient property of the B-

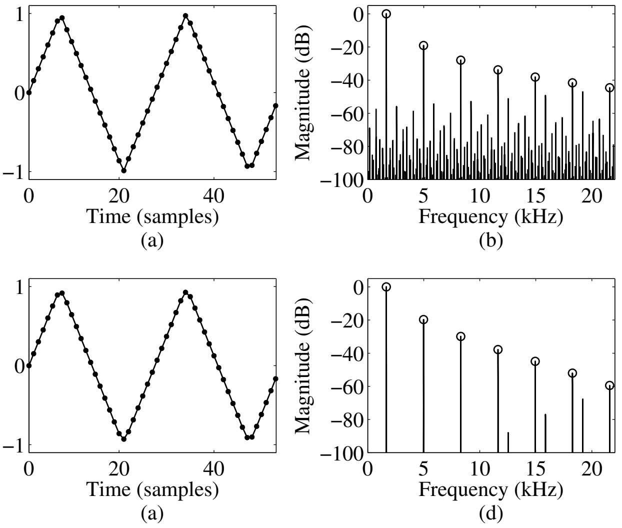

Figure 5: *Waveform and magnitude spectra of a (a)-(b) 1661.2-Hz trivial triangle wave, and (c)-(d) the same signal after four-point polyBLAMP correction. Circles indicate non-aliased components.*

spline polyBLAMP method is that it preserves the original range of signal values, as the correction is performed “inwards”, so to speak.

Several methods to synthesize triangular waveforms with reduced aliasing have been proposed. Stilson et al. initially suggested double integration of a bipolar bandlimited impulse train (BLIT) [16, 19]. Välimäki et al. developed a more efficient approach using a differentiated parabolic waveform (DPW) [24]. This approach was later optimized for synthesis of triangle waveforms by Ambrits and Bank [13] using efficient polynomial transition regions (EPTR). The EPTR method was used as a reference to evaluate the performance of the proposed polyBLAMP method.

The signal-to-noise ratio (SNR) of the A6 triangular waveform was measured with and without polyBLAMP correction. In this context, SNR was defined as the power ratio between harmonics and aliasing components. To show the limits of the proposed method, a second measurement was performed on a 4168-Hz (MIDI note C8) signal. This frequency represents the highest fundamental frequency on a piano. For further evaluation, the SNRs obtained using oversampling by factors 2 and 4 were also computed. The top two rows of Table 2 show the results obtained from these measurements. The polyBLAMP method exhibits results comparable to oversampling by 4 at a fraction of the computational costs. Additionally, the resulting SNRs for the EPTR algorithm were 54 dB and 43 dB for the A6 and C8 signals, respectively.

In terms of computational costs, the top row of Table 3 shows the average synthesis times for a 1-second C8 triangular signal using oversampling by 2 and 4, the EPTR method and the four-point polyBLAMP. These results were obtained by porting the algorithm into Python and using the `time` function. The polyBLAMP method yielded the fastest processing times.

Table 2: SNR measurements in dB for test signals of 1661 Hz (A6) and 4186 Hz (A8). The best SNR on each row is bolded.

Signal	Triv.	OS		polyBLAMP
		by 2	by 4	
Triangular A6	42 dB	52 dB	56 dB	**54 dB**
Triangular C8	30 dB	42 dB	46 dB	**45 dB**
Clipping A6	34 dB	42 dB	43 dB	**57 dB**
Clipping C8	24 dB	34 dB	38 dB	**42 dB**
Half-W. Rec. A6	40 dB	43 dB	44 dB	**61 dB**
Half-W. Rec. C8	28 dB	36 dB	38 dB	**48 dB**
Full-W. Rec. A6	32 dB	40 dB	41 dB	**53 dB**
Full-W. Rec. C8	20 dB	28 dB	30 dB	**39 dB**

4.2. Alias-Free Hard Clipping

Hard clipping is another example of an audio application where discontinuities in the first derivative of a signal are introduced [20]. Signal clipping is a form of distortion that limits the values of a signal that lie above or below a predetermined threshold. Symmetric hard clipping can be expressed as

$$f_c(x[n]) = \text{sgn}(x[n])\text{min}(|x[n]|\,, L), \tag{14}$$

where $x[n]$ is the input signal, $\text{sgn}(\cdot)$ is the sign function, and $L \in (0, 1]$ is the normalized clipping threshold. In practice, signal clipping may be necessary due to system limitations, e.g. to avoid overmodulating an audio transmitter. In discrete systems, it can be caused unintentionally due to data resolution constraints, or intentionally as when simulating an analog system in which signal values are saturated [25].

Fig. 6(a) shows the continuous-time clipped f_0-Hz sinusoid with clipping threshold $L = 0.7$ (solid line) together with the original sine wave (dashed line). Following the same approach as in the previous subsection, we evaluate the first derivative of this signal and observe that this derivative presents discontinuities at the exact points in time where it enters or leaves a saturation [see Fig. 6(b)]. Further derivation of this signal yields the waveform shown in Fig. 6(c), which contains impulses whose polarities depend on the direction of the observed discontinuities.

Implementation of the four-point polyBLAMP correction on an arbitrary input signal requires a polynomial to be fit to the four corner boundaries as described in Sec. 3. Then, the NR method can be used to estimate parameters d and μ by substituting $\rho = \pm L$ in (11). The polarity of ρ will depend on the polarity of the clipping point being corrected, as shown in Fig. 6(c). Fig. 7 shows the

Table 3: *Averaged computation time (in ms) for oversampling by factors of 2 and 4, the EPTR, and 4-point polyBLAMP methods.*

	Oversampling		EPTR	polyBLAMP
Signal	by 2	by 4		
Triangular C8	36 ms	72 ms	45 ms	**18 ms**
Clipping C8	46 ms	102 ms	-	**43 ms**
Half-W. Rect. C8	40 ms	89 ms	-	**18 ms**
Full-W. Rect. C8	41 ms	90 ms	-	**28 ms**

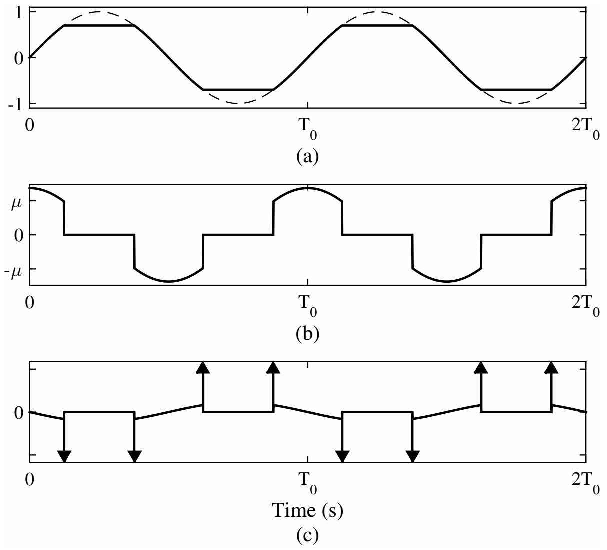

Figure 6: *(a) Continuous-time f_0-Hz sinusoid hard-clipped with clipping threshold $L = 0.8$, (b) its first and (c) second derivatives.*

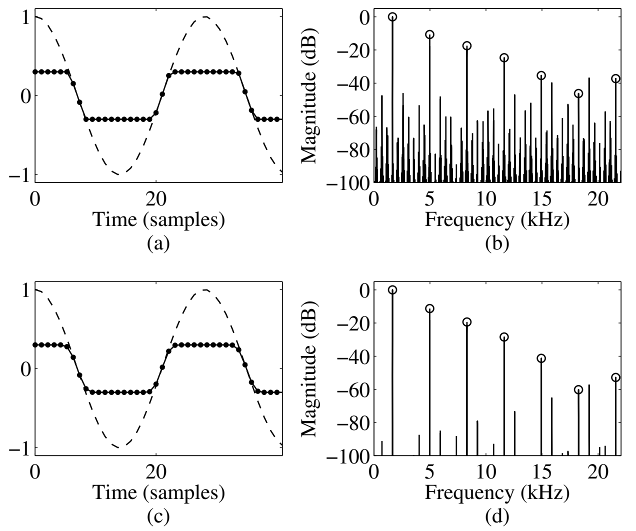

Figure 7: *Waveform and magnitude spectra of a (a)-(b) 1660-Hz trivial hard-clipped sine wave ($L = 0.3$), and (c)-(d) the same signal after four-point polyBLAMP correction.*

waveform and magnitude spectra for a 1661-Hz sinusoid clipped at $L = 0.3$ before and after four-point polyBLAMP correction. Once again, the corrected signal exhibits improved performance in terms of aliasing. For instance, the level of the most prominent aliasing component below the fundamental, at 720 Hz has been attenuated by 43 dB. As before, the clipping threshold is preserved after the correction, since the polyBLAMP method does not introduce any overshoot in the time domain.

Rows 3 and 4 of Table 2 show the SNRs measured for two sinusoidal signals (MIDI notes A6 and C8) trivially-clipped, using oversampling by factors 2 and 4, and after four-point polyBLAMP correction. All these measurements were performed using a clipping threshold $L = 0.3$. In this application, the four-point polyBLAMP method also exhibits better performance than oversampling by low factors, with SNR improvements of 22.6 and 17.4 dB for each respective signal. In terms of computational costs, the polyBLAMP method shows similar costs to those of oversampling by factor 2 but with improved SNR (see Tables 2 and 3).

As a final note on hard clipping, Fig. 6(c) shows that the second derivative of the signal has discontinuities around each impulse. These discontinuities, while small, will contribute to the overall aliasing seen at the output of the clipper. Integrating the BLAMP or polyBLAMP function should, in theory, yield a correction function that further reduces aliasing. This idea is not explored any further in this study and is left as future work.

4.3. Alias-Free Half-Wave Rectification

Signal rectification is a type of memoryless nonlinear processing that can be used to introduce harmonic distortion. In a half-wave rectifier, only positive portions of the waveform are kept, while negative portions are set to zero

$$f_r(x[n]) = \max(x[n], 0). \tag{15}$$

In analog applications, this can be achieved using a diode, which only allows current to flow in one direction. A particular feature of half-wave rectification is that it introduces even harmonics only.

Figs. 8(a) and 8(b) show the continuous-time domain waveform for a half-wave rectified sine wave and its first derivative, respectively. As expected, the corners introduced by the rectifier translate into discontinuities in the derivative. The magnitude of each discontinuity is determined by the slope μ of the original signal at the zero-crossings. Derivating this signal once more yields the positive impulse train depicted in Fig. 8(c).

The polyBLAMP method can be used to round the corners seen in Fig. 8(a) by centering it at the zero crossings. Parameters d and μ can be estimated by replacing $\rho = 0$ in (9) and (11). This is equivalent to finding the zero-crossings of the input signal. Fig. 9 shows the waveforms and magnitude spectra for a 1660-Hz sinusoid without and with polyBLAMP correction. In this example, the removal of the spurious frequency components is evident, with aliases below the fundamental are attenuated by more than 40 dB. Rows 5 and 6 of Table 2 show the proposed method outperforms oversampling by 2 and 4, and increases SNR by more than 20 dB with respect to a trivial implementation of half-wave rectification. In terms of computational costs, this implementation is cheaper than oversampling by a factor 2, as shown in Table 3.

4.4. Alias-Free Full-Wave Rectification

In full-wave rectification, negative portions of the waveform are not zeroed, but inverted, for example by taking the absolute value:

$$f_R(x[n]) = |x[n]|\,. \tag{16}$$

This process introduces both even and odd harmonics. Several analog audio effects incorporate a full-wave rectifier as part of a larger signal processing chain, such as the Octavio Fuzz pedal [26]. It can also be found as a stand-alone effect in modular synthesizer units, e.g. the Malekko 8NU8R [27].

Fig. 10(a) shows the continuous-time waveform for a rectified sine wave. Figs. 10(b) and 10(c) show the first and second derivatives of this waveform, respectively. In this case, the magnitude of the discontinuities that appear in the first derivative are defined

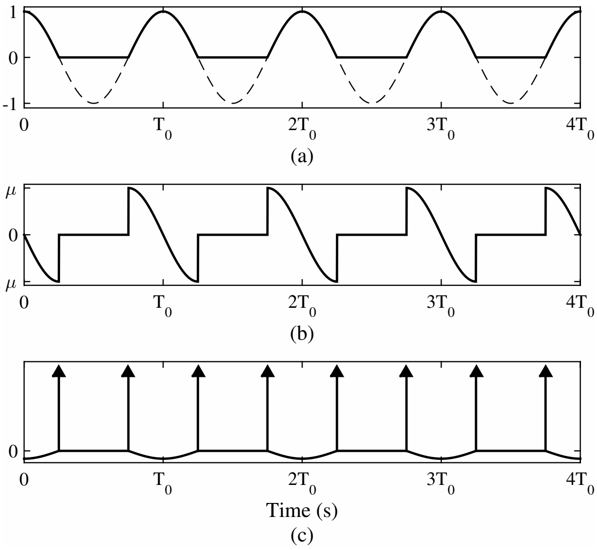

Figure 8: *(a) Half-wave rectified continuous-time sine wave, (b) its first and (c) second derivatives.*

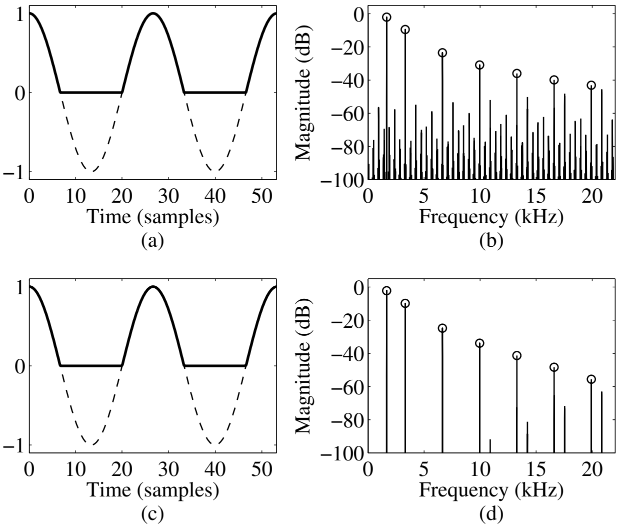

Figure 9: *Waveform and magnitude spectrum of a (a)-(b) 1660-Hz trivial half-wave rectified sine wave, and (c)-(d) the same signal after four-point polyBLAMP correction.*

as twice the slope of the original signal at the zero-crossings. As with hard clipping, both types of rectification also introduce discontinuities in subsequent derivatives of the signal, hinting at the possibility that further correction could be achieved using higher-order bandlimited integral functions.

The polyBLAMP method can be used in the same fashion as with half-wave rectification by scaling the slope parameter by factor 2. The bottom two rows of Table 2 show the measured SNRs for the two sinusoidal test signals discussed in the previous subsections. Once again, the polyBLAMP method outperforms oversampling by factors 2 and 4, offering a nearly 20-dB improvement in SNR with reduced computational costs (cf. Table 3).

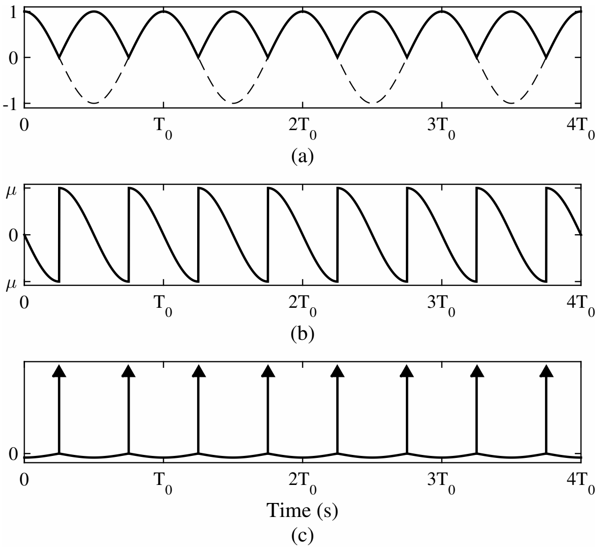

Figure 10: *(a) Full-wave rectified continuous-time sine wave, (b) its first and (c) second derivatives.*

Lane et al. [28] have proposed to use a full-wave rectified sine wave (after further linear filtering) to approximate the sawtooth waveform. Välimäki and Huovilainen analyzed this approximation showing that, while it contains considerably less aliasing than the trivial sawtooth, the aliasing can be still be audible at high fundamental frequencies [12]. The polyBLAMP method could now be used to further enhance this sawtooth generation method.

Additionally, to demonstrate that the BLAMP method is applicable to nonlinearly processed arbitrary signals and not just sine waves, Fig. 11 shows the waveform and magnitude spectrum for a synthetic string sound recording before and after full-wave rectification without and with polyBLAMP correction. Overall, aliasing components have been reduced by nearly 20 dB on average.

5. CONCLUSIONS

The corner-rounding capabilities of the polynomial approximation of the BLAMP function, or polyBLAMP, were studied. In addition to alias-free synthesis of triangular waveforms, it can enhance certain nonlinear waveshaping methods, which introduce discontinuities in the first derivative of the signal waveform. The fractional delay and slope at each corner need to be estimated, and then this method can correct a few samples in the neighborhood of each corner. The polyBLAMP method helps implementing alias-free versions of hard-clipping and rectification for arbitrary signals without oversampling, and thus enables enhanced nonlinear audio effects processing.

Supplementary material, including MATLAB code and sound examples, can be found in `http://research.spa.aalto.fi/publications/papers/dafx16-blamp/`.

6. REFERENCES

[1] T. S. Stilson, *Efficiently-Variable Non-Oversampled Algorithms in Virtual-Analog Music Synthesis*, PhD thesis, Stanford University, Stanford, CA, USA, June 2006.

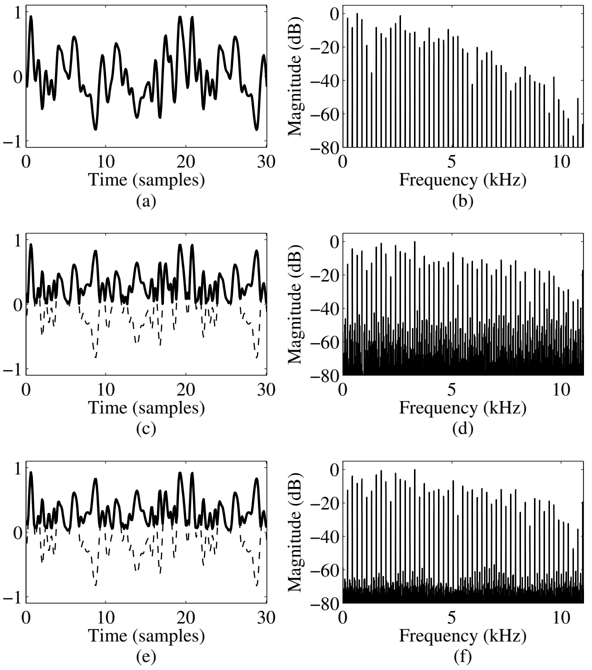

Figure 11: *Waveform and spectrum for a (a)-(b) synthetic string recording and the same signal after full-wave rectification (c)-(d) before and (e)-(f) after four-point polyBLAMP correction.*

[2] D. T. Yeh, *Digital Implementation of Musical Distortion Circuits by Analysis and Simulation*, PhD thesis, Stanford University, Stanford, CA, USA, June 2009.

[3] V. Välimäki and A. Huovilainen, "Oscillator and filter algorithms for virtual analog synthesis," *Computer Music J.*, vol. 30, no. 2, pp. 19–31, 2006.

[4] J. Pakarinen and D. T. Yeh, "A review of digital techniques for modeling vacuum-tube guitar amplifiers," *Computer Music J.*, vol. 33, no. 2, pp. 85–100, 2009.

[5] H. Thornburg, "Antialiasing for nonlinearities: Acoustic modeling and synthesis applications," in *Proc. Int. Comput. Music Conf.*, Beijing, China, Oct. 1999, pp. 66–69.

[6] D. Mapes-Riordan, "A worst-case analysis for analog-quality (alias-free) digital dynamics processing," *J. Audio Eng. Soc.*, vol. 47, no. 11, pp. 948–952, Nov. 1999.

[7] J. Schimmel, "Audible aliasing distortion in digital audio synthesis," *Radioengineering*, vol. 21, no. 1, pp. 57, 2012.

[8] J. Schattschneider and U. Zölzer, "Discrete-time models for nonlinear audio systems," in *Proc. Digital Audio Effects Workshop*, Trondheim, Norway, Dec. 1999, pp. 45–48.

[9] V. Lazzarini and J. Timoney, "New perspectives on distortion synthesis for virtual analog oscillators," *Computer Music J.*, vol. 34, no. 1, pp. 28–40, 2010.

[10] P. Kraght, "Aliasing in digital clippers and compressors," *J. Audio Eng. Soc.*, vol. 48, no. 11, pp. 1060–1064, Nov. 2000.

[11] V. Välimäki, J. Pekonen, and J. Nam, "Perceptually informed synthesis of bandlimited classical waveforms using integrated polynomial interpolation," *J. Acoust. Soc. Am.*, vol. 131, no. 1, pp. 974–986, Jan. 2012.

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

[13] D. Ambrits and B. Bank, "Improved polynomial transition regions algorithm for alias-suppressed signal synthesis," in *Proc. 10th Sound and Music Computing Conf. (SMC2013)*, Stockholm, Sweden, Aug. 2013, pp. 561–568.

[14] A. Huovilainen, "Design of a scalable polyphony-MIDI synthesizer for a low cost DSP," M.S. thesis, Aalto University, Espoo, Finland, 2010.

[15] R. N. Bracewell, *The Fourier Transform and its Applications*, McGraw-Hill, 2nd edition, 1986.

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

[17] E. Brandt, "Hard sync without aliasing," in *Proc. Int. Computer Music Conf.*, Havana, Cuba, Sep. 2001, pp. 365–368.

[18] W. Pirkle, *Designing Audio Effect Plug-Ins in C++*, Focal Press, 1st edition, Oct. 2013.

[19] J. Nam, V. Välimäki, J. S. Abel, and J. O. Smith, "Efficient antialiasing oscillator algorithms using low-order fractional delay filters," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 4, pp. 773–785, May 2010.

[20] F. Esqueda, S. Bilbao, and V. Välimäki, "Aliasing reduction in clipped signals," *IEEE Trans. Signal Process.*, 2016, accepted for publication.

[21] F. Esqueda, V. Välimäki, and S. Bilbao, "Aliasing reduction in soft-clipping algorithms," in *Proc. European Signal Process. Conf.*, Nice, France, Aug. 2015, pp. 2059–2063.

[22] F. B. Hildebrand, *Introduction to Numerical Analysis*, Dover Publications, 2nd edition, Jun. 1987.

[23] J. Kleimola and V. Välimäki, "Reducing aliasing from synthetic audio signals using polynomial transition regions," *IEEE Signal Process. Lett.*, vol. 19, no. 2, pp. 67–70, Feb. 2012.

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

[25] D. Rossum, "Making digital filters sound analog," in *Proc. Int. Computer Music Conf.*, San Jose, CA, Oct. 1992, pp. 30–33.

[26] Dunlop Manufacturing, Inc., "Jimi Hendrix Octavio Pedal," Product information available at `http://www.jimdunlop.com/product/jhoc1-jimi-hendrix-octavio-effect`, accessed March 19, 2015.

[27] Malekko Heavy Industry Corporation, "8NU8R Dual Analog Attenuator Module," Product information available at `https://malekkoheavyindustry.com/product/8nu8r/`, accessed March 19, 2015.

[28] J. Lane, D. Hoory, E. Martinez, and P. Wang, "Modeling analog synthesis with DSPs," *Computer Music J.*, vol. 21, no. 4, pp. 23–41, 1997.