Rounding corners with blamp

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Proceedings of the 19th International Conference on Digital Audio Effects

Publisher Rights Statement:
All copyrights of the individual papers remain with their respective authors.

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
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 [3,4]. 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 [3]. 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

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

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 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 \[18\], which yields

\[ h^{(0)}(t) = f_s \text{sinc}(f_s t), \]

where \( f_s \) represents the sampling rate and \( \text{sinc}(x) = \sin(\pi x) / \pi x \).

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

\[ h^{(1)}(t) = \frac{1}{2} + \frac{1}{\pi} \text{Si}(\pi f_s t), \]

where \( \text{Si}(x) \) is the sine integral, defined as \( \text{Si}(x) = \int_0^x \frac{\sin t}{t} \, dt \).

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 \[17, 11\]. Figure 1(d) shows the shape for this function.

Moving on, \[9\] can be integrated once more using integration by parts, yielding

\[ h^{(2)}(t) = t \left[ \frac{1}{2} + \frac{1}{\pi} \text{Si}(\pi f_s t) \right] + \frac{\cos(\pi f_s t)}{\pi^2 f_s} \]

where \( \text{Si}(x) \) is the sine integral, defined as \( \text{Si}(x) = \int_0^x \frac{\sin t}{t} \, dt \).

This equation gives the closed form expression for the BLAMP function with unit slope, and its shape is shown in Fig. 1(i). At first glance, Figs. 1(e) and 1(f) may appear indistinguishable. However, computing the difference between \[7\] and \[9\] 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 edge.

\[ h(t) = h^{(1)}(t) + \cos(\pi f_s t) / \pi^2 f_s. \]

This equation gives the closed form expression for the BLAMP function with unit slope, and its shape is shown in Fig. 1(i). At first glance, Figs. 1(e) and 1(f) may appear indistinguishable. However, computing the difference between \[7\] and \[9\] 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 edge.

\[ h(t) = h^{(1)}(t) + \cos(\pi f_s t) / \pi^2 f_s. \]

This equation gives the closed form expression for the BLAMP function with unit slope, and its shape is shown in Fig. 1(i). At first glance, Figs. 1(e) and 1(f) may appear indistinguishable. However, computing the difference between \[7\] and \[9\] 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 edge.

\[ h(t) = h^{(1)}(t) + \cos(\pi f_s t) / \pi^2 f_s. \]

This equation gives the closed form expression for the BLAMP function with unit slope, and its shape is shown in Fig. 1(i). At first glance, Figs. 1(e) and 1(f) may appear indistinguishable. However, computing the difference between \[7\] and \[9\] 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 edge.

\[ h(t) = h^{(1)}(t) + \cos(\pi f_s t) / \pi^2 f_s. \]

This equation gives the closed form expression for the BLAMP function with unit slope, and its shape is shown in Fig. 1(i). At first glance, Figs. 1(e) and 1(f) may appear indistinguishable. However, computing the difference between \[7\] and \[9\] 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 edge.

\[ h(t) = h^{(1)}(t) + \cos(\pi f_s t) / \pi^2 f_s. \]

This equation gives the closed form expression for the BLAMP function with unit slope, and its shape is shown in Fig. 1(i). At first glance, Figs. 1(e) and 1(f) may appear indistinguishable. However, computing the difference between \[7\] and \[9\] 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 edge.

\[ h(t) = h^{(1)}(t) + \cos(\pi f_s t) / \pi^2 f_s. \]
function does not have finite support, it does not vanish. Therefore, its truncation to a finite interval introduces small discontinuities which will 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, its truncation to a finite interval introduces small discontinuities which closely resemble the central lobe of the bandlimited impulse [see Fig. 4(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. A two-point version of the polyBLAMP function can be written as a piecewise polynomial using the coefficients for the third-order B-spline basis function and the trivial ramp functions.

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 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 (5)]. B-spline interpolating polynomials have been used in this study due to their steep spectral decay which makes them suitable for anti-aliasing applications [10, 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 $D = \Delta t + d$ samples, where $\Delta t = 1$, 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. 4(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.

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 $n \in \mathbb{Z}_{>0}$ 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 $n_a - 1$, $n_a$, $n_b$, and $n_b + 1$ will be processed by the algorithm. The aim is to fit a polynomial of the form $f(D) = aD^n + bD^{n-1} + cD + d + e$ 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

\[
a = -\frac{1}{6} s[n_a - 1] + \frac{1}{2} s[n_a] - \frac{1}{6} s[n_b] + \frac{1}{3} s[n_b + 1] \\
b = \frac{1}{6} s[n_a - 1] - \frac{1}{2} s[n_a] + \frac{1}{2} s[n_b] - \frac{1}{2} s[n_b + 1] \\
c = -\frac{5}{6} s[n_a - 1] + \frac{5}{6} s[n_a] - \frac{5}{3} s[n_b] + \frac{1}{3} s[n_b + 1] \\
e = \frac{1}{2} s[n_b + 1].
\]

(8)

After fitting the polynomial, the next step is to obtain the inter-

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].

<table>
<thead>
<tr>
<th>Span</th>
<th>Third-order B-spline basis function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[-2T, -T]$</td>
<td>$D^3/6 - D^2/2 + D/2 - 1/6$</td>
</tr>
<tr>
<td>$[-T, 0]$</td>
<td>$-D^3/2 + 2D^2 - 2D + 3/6$</td>
</tr>
<tr>
<td>$[0, T]$</td>
<td>$D^3/2 - 5D^2/6 + 7D/2 - 5/6$</td>
</tr>
<tr>
<td>$[T, 2T]$</td>
<td>$-D^3/6 + D^2/2 + D + 1/3$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Span</th>
<th>First integral: Four-point polyBLEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[-2T, -T]$</td>
<td>$D^3/24 - D^3/4 + D/4 + 1/24$</td>
</tr>
<tr>
<td>$[-T, 0]$</td>
<td>$-D^3/8 + 2D^3/3 - D^3 + 2D/3 - 1/6$</td>
</tr>
<tr>
<td>$[0, T]$</td>
<td>$D^3/8 - 5D^3/6 + 7D^3/4 - 5D^3/6 + 7/24$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Span</th>
<th>Second integral: Four-point polyBLAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[-T, 0]$</td>
<td>$-D^3/40 + D^3/6 - D^3/3 + 2D/3 - 1/120$</td>
</tr>
<tr>
<td>$[0, T]$</td>
<td>$D^3/40 - 5D^3/6 + 7D^3/4 - 5D^3/6 + 7/24 - 1/24$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Span</th>
<th>Four-point polyBLAMP residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[-2T, T]$</td>
<td>$d^3/120$</td>
</tr>
<tr>
<td>$[-T, 0]$</td>
<td>$-d^3/40 + d^3/24 + d^3/12 + d^3/12 + d/24 + 1/120$</td>
</tr>
<tr>
<td>$[0, T]$</td>
<td>$d^3/40 - d^3/12 + d^3/3 - d/2 + 7/30$</td>
</tr>
<tr>
<td>$[T, 2T]$</td>
<td>$-d^3/120 + d^3/24 - d^3/12 + d^3/12 - d/24 + 1/120$</td>
</tr>
</tbody>
</table>
section of this curve with \( \rho \), the corner parameter. The value of \( \rho \) 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 \( \rho \) 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

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

The resulting value \( D_\rho \) 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

\[
\phi(D_\rho) = 3aD_\rho^2 + 2bD_\rho + c. \tag{12}
\]

Finally, the value of \( d \) can be computed as \( d = D_\rho - 1 \). This represents an estimated sharp edge at \( n_\alpha + d \), i.e. \( \phi(n_\alpha + d) = \rho \).

### 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 antialiasing 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. 4, 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 \( 2f_0 \), where \( \mu \) 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).

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 \([10]\). 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 \( 2\mu \) 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 \( T_\phi = 2f_\mu/f_1 \) 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-
The 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.

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.

<table>
<thead>
<tr>
<th>Signal by 2</th>
<th>OS by 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trivial A6</td>
<td>42 dB</td>
</tr>
<tr>
<td>Triangular C8</td>
<td>30 dB</td>
</tr>
<tr>
<td>Clipping A6</td>
<td>34 dB</td>
</tr>
<tr>
<td>Clipping C8</td>
<td>24 dB</td>
</tr>
<tr>
<td>Half-W. Rect. A6</td>
<td>40 dB</td>
</tr>
<tr>
<td>Half-W. Rect. C8</td>
<td>28 dB</td>
</tr>
<tr>
<td>Full-W. Rect. A6</td>
<td>32 dB</td>
</tr>
<tr>
<td>Full-W. Rect. C8</td>
<td>20 dB</td>
</tr>
</tbody>
</table>

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

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

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 \( f_c \)-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 \( \rho \) by substituting \( \rho \rightarrow \pm L \) in (11). The value of \( \rho \) will depend on the polarity of the clipping point being corrected, as shown in Fig. 6(c). Fig. 7 shows the computational time (in ms) for oversampling by factors 2 and 4, the EPTR, and 4-point polyBLAMP methods.

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

<table>
<thead>
<tr>
<th>Signal by 2</th>
<th>Oversampling by 2</th>
<th>EPTR by 4</th>
<th>polyBLAMP by 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangular C8</td>
<td>36 ms</td>
<td>72 ms</td>
<td>45 ms</td>
</tr>
<tr>
<td>Clipping C8</td>
<td>46 ms</td>
<td>102 ms</td>
<td>-</td>
</tr>
<tr>
<td>Half-W. Rect. C8</td>
<td>40 ms</td>
<td>89 ms</td>
<td>-</td>
</tr>
<tr>
<td>Full-W. Rect. C8</td>
<td>41 ms</td>
<td>90 ms</td>
<td>-</td>
</tr>
</tbody>
</table>
signal rectification is a type of memoryless nonlinear processing that can be used to introduce harmonics into a signal. In half-wave rectification, negative portions of the waveform are not zeroed, but inverted, for example by taking the absolute value:

\[ f_R(x[n]) = |x[n]|. \] (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.

4.3. Alias-Free Half-Wave Rectification

Signal rectification is a type of memoryless nonlinear processing that can be used to introduce harmonics into a signal. In half-wave rectification, negative portions of the waveform are not zeroed, but inverted, for example by taking the absolute value:

\[ f_R(x[n]) = |x[n]|. \] (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.

\[ f_R(x[n]) = \max(x[n], 0). \] (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 larger signal processing chains, such as the Malekko 8NU8R [27].

Figs. 8(a) and 8(b) show the continuous-time domain waveform for a half-wave rectified sine wave and its first derivative, respectively.

The polyBLAMP method can be used to round the corners of the waveform as it translates into discontinuities in the derivative. The magnitude of each discontinuity is determined by the slope of the original signal at the zero-crossings. Derivating this signal once more yields the positive impulse train depicted in Fig. 8(c).

As a final note on hard clipping, Fig. 9(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.

4.4. 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:

\[ f_R(x[n]) = |x[n]|. \] (16)
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/](http://research.spa.aalto.fi/publications/papers/dafx16-blamp/)

6. REFERENCES

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


