Aliasing reduction in clipped signals

Citation for published version:
Esqueda, F, Bilbao, S & Valimaki, V 2016, 'Aliasing reduction in clipped signals', *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5255-5267. https://doi.org/10.1109/TSP.2016.2585091

Digital Object Identifier (DOI):
10.1109/TSP.2016.2585091

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
IEEE Transactions on Signal Processing

Aliasing Reduction in Clipped Signals

Fabián Esqueda, Stefan Bilbao, *Senior Member, IEEE* and Vesa Välimäki, *Fellow, IEEE*

***Abstract*—An aliasing reduction method for hard-clipped sampled signals is proposed. Clipping in the digital domain causes a large amount of harmonic distortion, which is not bandlimited, so spectral components generated above the Nyquist limit are reflected to the baseband and mixed with the signal. A model for an ideal bandlimited ramp function is derived, which leads to a post-processing method to reduce aliasing. A number of samples in the neighborhood of a clipping point in the waveform are modified to simulate the Gibbs phenomenon. This novel method requires estimation of the fractional delay of the clipping point between samples and the first derivative of the original signal at that point. Two polynomial approximations of the bandlimited ramp function are suggested for practical implementation. Validation tests using sinusoidal, triangular, and harmonic signals show that the proposed method achieves high accuracy in aliasing reduction. The proposed 2-point and 4-point polynomial correction methods can improve the signal-to-noise ratio by 12 dB and 20 dB in average, respectively, and are more computationally efficient and cause less latency than oversampling, which is the standard approach to aliasing reduction. An additional advantage of the polynomial correction methods over oversampling is that they do not introduce overshoot beyond the clipping level in the waveform. The proposed techniques are useful in audio and other fields of signal processing where digital signal values must be clipped but aliasing cannot be tolerated.**

***Index Terms*—Antialiasing, interpolation, nonlinear distortion, signal denoising, signal sampling.**

I. Introduction

CLIPPING is a form of distortion that limits the values of a signal that lie above or below certain 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, such as when a sample exceeds the maximum value that can be represented, or intentionally as when simulating a process in which signal values are constrained. Clipping is a nonlinear operation and introduces frequency components not present in the original signal. In the digital domain, when the frequencies of these new components exceed the Nyquist limit, the components are reflected back into the baseband, causing *aliasing*. This paper proposes a novel method to reduce aliasing in clipped signals.

Aliasing can severely affect the quality of a digital signal by corrupting the data it represents. For example, in audio applications, aliasing can cause severe audible effects such as beating, inharmonicity and heterodyning [1]. Nevertheless, if the aliased components are sufficiently attenuated, their effects become inaudible and can therefore be neglected [1], [2].

A large share of earlier research on clipped signals has focused on *declipping* or the reconstruction of the underlying original unclipped signal. Abel and Smith [3] introduced optimization methods to reconstruct the clipped samples based on constraints. Recent work on declipping has considered methods based on matching pursuits [4], compressed sensing [5], social sparsity [6], sparse and co-sparse regularization [7], and non-negative matrix factorization [8]. Declipping can improve the audio quality of nonlinearly-compressed sound files [9] and help speaker recognition [10], for example.

The purpose of this work is not signal reconstruction but enhancement, by allowing clipping to occur and by suppressing the aliasing introduced. Practical applications for the proposed approach are found in systems in which clipping is implemented digitally. One application in radio broadcasting and music production is the limiting of the maximum values of the signal, such as in dynamic range compressors and limiters, which are known to introduce distortion and aliasing [11]–[13]. In such applications, declipping is out of question, because it would cancel the limiting effect, which is necessary for maximizing the signal level. However, the antialiasing method proposed in this work can be useful, since it cleans the clipped signal by suppressing the aliasing, and restores the limited distribution of sample values, as required.

Other practical applications involving digital clipping are simulations of analog and physical systems in which signal values are naturally limited. In digital simulation of analog filters, hard clipping is used as a simple model for the saturation of large signals inside analog filters [14], [15]. In the digital modeling of vacuum-tube amplifiers and guitar effects, the saturating characteristics must also be implemented digitally [16]–[18]. Often hard clipping is used in connection with soft clipping so that the latter works at small signal values while the former saturates (clips) the large signal values to a maximum or minimum value. Antialiasing for the combination of a soft and a hard clipper in this context was the first application for the method discussed in this paper [19]. Similar saturating modules appear in physical models of musical wind and string instruments in which a saturating waveshaper simulates the nonlinear interaction between the excitation and state of an acoustic resonator, such as a tube or a string [20], [21].

Full-wave and half-wave rectification are special cases of hard clipping. Thus, the proposed method can enhance such digitally rectified signals, which have applications in various fields. For example in music processing, full-wave rectification offers an easy implementation of an octaver, a device which doubles the fundamental frequency of a signal [13].

The only previous general method for reducing aliasing in digitally-clipped signals is oversampling [11], [16], [18], [22]–

This work was supported by the CIMO Center for International Mobility and the Aalto ELEC Doctoral School.

F. Esqueda and V. Välimäki are with the Department of Signal Processing and Acoustics, Aalto University School of Electrical Engineering, FI-00076 AALTO, Espoo, Finland (e-mail: fabian.esqueda@aalto.fi; vesa.valimaki@aalto.fi).

S. Bilbao is with the Acoustics and Audio Group, University of Edinburgh, Edinburgh EH9 3JZ, UK (e-mail: sbilbao@staffmail.ed.ac.uk).

[24]. Its use is limited to cases where the input signal of a nonlinear device can be accessed. In oversampling, the input signal of the nonlinear device is upsampled (e.g. by factor of eight) so that most of the spectral components generated by the clipper can be suppressed using a digital lowpass filter. Then, the output signal is downsampled back to the original rate. The level of aliasing reduction will then depend on the oversampling factor and order of the antialiasing lowpass filter, which will together also determine the computational cost of the process as a whole.

The concept of aliasing suppression has been previously studied in the design of digital oscillators such as those used in music synthesis [25]. Here, a correction function based on a bandlimited step (BLEP) can reduce aliasing in sawtooth and rectangular oscillators [1], [26], [27]. In the BLEP method, a trivially sampled waveform is modified around each discontinuity by adding to its sample values a correction term determined by the BLEP function. This suppresses aliasing whereas the desired signal components are largely retained. The fractional delay value is needed at each correction point, but in sound synthesis it is typically known from the phase counter value [27]. Another method based on the assumption of smooth polynomial transition regions around each signal discontinuity leads to similar results as the polynomial approximation of the BLEP method [28], [29]. This approach has been derived as an extension of the differential polynomial waveform synthesis method [30], [31]. However, the anti-aliasing signal synthesis techniques are not directly applicable to enhancing hard-clipped arbitrary signals.

The approach proposed here is based on the idea of quasi-bandlimiting a hard-clipped signal. We first consider the case of an ideal bandlimited hard-clipped sinewave and design a correction function based on the ideal bandlimited ramp (BLAMP) function. This function is used to adjust an arbitrary number of sample values before and after each clipping point of a signal. Additionally, this paper proposes a computationally efficient implementation of the BLAMP correction technique based on a polynomial approximation using B-spline interpolation (polyBLAMP). The polyBLAMP is shown to effectively reduce aliasing distortion in hard-clipped signals with less computational costs than when oversampling. The first case of the BLAMP method (2-point correction) was published recently elsewhere [19], and the present paper generalizes and expands these results.

This paper is organized as follows. Section II provides an overview on the effect of hard clipping in the digital domain by considering the case of an ideal bandlimited clipped sinewave. Section III derives the BLAMP correction function. Details on the implementation and performance of the polynomial approximation of the BLAMP function are presented in Sections IV and V, respectively. Finally, concluding remarks and recommendations for further work appear in Section VI.

II. Ideal Bandlimited Clipped Sinewave

The effects of hard clipping in the digital domain can be appreciated by comparing its trivial implementation with an ideal bandlimited case. To do so, we begin by considering the simple case of a continuous-time sinewave

$$s(t) = \cos\left(2\pi f_0 t\right), \tag{1}$$

where t is time and f_0 is fundamental frequency in Hz. Next, we define $c(t)$, a time-domain expression for this signal after bipolar (positive and negative) clipping:

$$c(t) = \operatorname{sgn}\left[s(t)\right] \min\left(|s(t)|, L\right). \tag{2}$$

Here, $0 < L < 1$ is the clipping threshold relative to a normalized peak amplitude of 1 and $\operatorname{sgn}(\cdot)$ is the sign function.

Now, in order to generate a digital bandlimited clipped sinewave, we follow the work of [11] and derive the Fourier series expansion of (2) and synthesize all harmonics up to the Nyquist limit $f_s/2$, where f_s is the sampling rate of the system. Due to even symmetry, the expansion of this signal will contain odd harmonics only and can be defined as

$$c_{\text{bl}}[n] = \sum_{k=1}^{K} C(2k-1)\cos(2\pi(2k-1)f_0 nT), \tag{3}$$

where $n \in \mathbb{Z}_{\geq 0}$ is the sample index, $T = 1/f_s$ is the sampling interval, and

$$K = \left\lceil \left\lfloor \frac{f_s}{4f_0} - \frac{1}{2} \right\rfloor + \frac{1}{2} \right\rceil \tag{4}$$

determines the frequency of the highest harmonic to synthesize (Kf_0). The amplitude of the m^{th} coefficient, $C(m)$, is given by [11]

$$\begin{aligned} C(m) = \delta(m-1) + \tau(L)(2L\text{sinc}[m\tau(L)] \\ - \text{sinc}[(m-1)\tau(L)] + \text{sinc}[(m+1)\tau(L)]), \end{aligned} \tag{5}$$

where m is restricted to positive odd integers, $\delta(m)$ is the unit impulse function, $\text{sinc}(x)$ is the cardinal sine function, defined as $\text{sinc}(x) = \sin(\pi x)/(\pi x)$, and $\tau(L)$ is defined as

$$\tau(L) = \frac{1}{2} - \frac{\sin^{-1}(L)}{\pi}. \tag{6}$$

This parameter determines the duration of the clipped portions of the signal relative to one period ($1/f_0$).

Figs. 1(a) and 1(c) show two periods of a 1245-Hz clipped digital sinewave synthesized without and with bandlimiting, respectively. All the examples in this paper were implemented using 64-bit MATLAB version 8.5 (unless otherwise specified) and a fixed sampling rate $f_s = 44.1\,\text{kHz}$, a typical value in audio applications. To generate the signal in Fig. 1(a), the unclipped signal (1) was converted to the digital domain as

$$s[n] = \cos(2\pi f_0 nT) \tag{7}$$

and processed by the bipolar clipping function, yielding a discrete-time expression for the trivial clipped sinusoid:

$$c[n] = \text{sgn}(s[n])\text{min}(|s[n]|, L). \tag{8}$$

The signal shown in Fig. 1(c) was computed using (3) to (6). In Figs. 1(b) and 1(d), circles have been used to indicate non-aliased components. As expected, the spectrum of the non-bandlimited signal in Fig. 1(b) exhibits high levels of aliasing distortion.

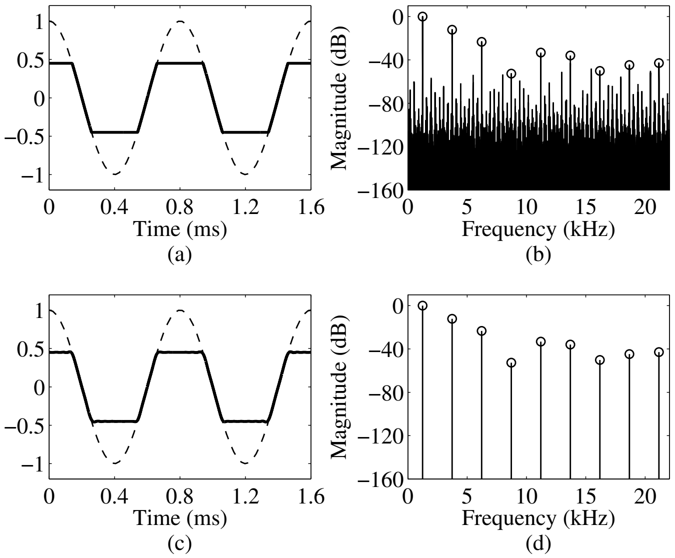

Fig. 1: Waveform and magnitude spectrum of (a)-(b) trivial clipped sinewave and (c)-(d) bandlimited clipped sinewave. In each case, the dashed line shows the original undistorted waveform. Circles indicate non-aliased components. The clipping parameter is $L = 0.45$ in this and all other examples.

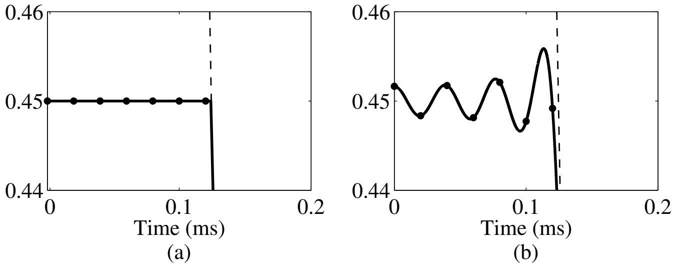

Fig. 2: Detail around the first clipping point of the waveforms in (a) Fig. 1(a) and (b) Fig. 1(c). The Gibbs phenomenon can be seen in (b).

At simple glance, the waveforms in Fig. 1(a) and Fig. 1(c) seem identical. However, a closer inspection around any clipping point reveals otherwise. Figs. 2(a) and 2(b) show a close-up view of the first clipping point of each signal. Fig. 2(b) has been produced by computing 100 samples per sampling interval to interpolate the actual signal shape between the samples. It can be seen in Fig. 2(b) that the hard edge that results from a trivial implementation is replaced with an oscillatory ripple in the bandlimited case. This ripple is the well-known Gibbs phenomenon that occurs when an infinite Fourier series is truncated [32], as is the case with this example. The maximum value, or overshoot, achieved by this overshoot will depend on f_0 and L. Therefore, in order to reduce the level of aliasing distortion in hard-clipped signals, we must derive a correction function that simulates the ripple seen in the bandlimited case.

III. Bandlimited Ramp Function

As mentioned in the previous example, the amount of overshoot was found to depend on both frequency and clipping threshold. Together, these two parameters also determine the slope of the unclipped signal at the clipping points, i.e. the exact points in time when clipping either begins or stops. Therefore, the region around any given clipping point can be visualized as a function that consists of a flat region, a hard edge or corner, and a slope. In its simplest case, this function would resemble a ramp that starts rising when time equals zero, as shown in Fig. 3(a). This trivial ramp can be defined as

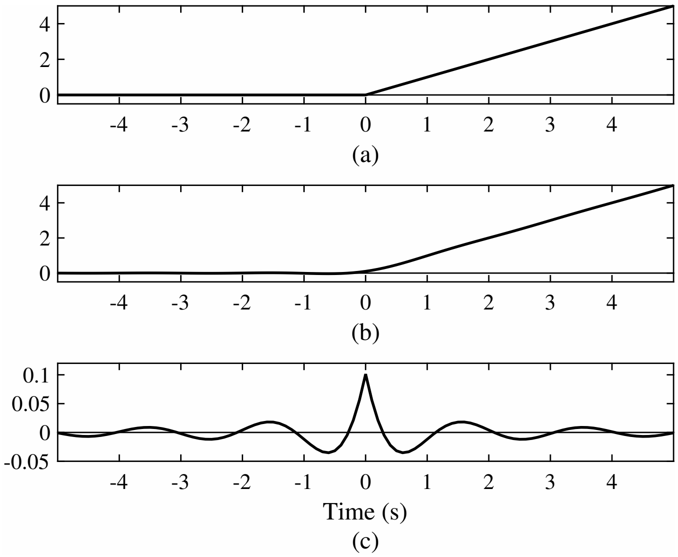

Fig. 3: Continuous-time (a) trivial ramp, (b) BLAMP, and (c) BLAMP residual functions.

$$r(t) = \begin{cases} 0 & \text{when } t < 0 \\ t & \text{when } t \geq 0. \end{cases} \tag{9}$$

The first derivative of this function is the unit step function. Since the unit step function consists of two flat regions separated by a discontinuity, its derivative is then defined as an impulse. Therefore, an expression for the BLAMP function can be derived by double integration of the bandlimited impulse [27].

The bandlimited impulse is defined by the well known expression which is equivalent to the impulse response of an ideal brickwall lowpass filter [25], defined as

$$h^{(0)}(t) = f_{\text{s}}\text{sinc}(f_{\text{s}}t). \tag{10}$$

Integrating (10) results in the closed form equation for the BLEP function [27], expressed as

$$h^{(1)}(t) = \frac{1}{2} + \frac{1}{\pi}\text{Si}(\pi f_{\text{s}} t), \tag{11}$$

where $\text{Si}(x)$ is the sine integral, defined as $\text{Si}(x) = \int_0^x \frac{\sin(t)}{t}\text{d}t$. Equation (11) can then be integrated using integration by parts, yielding

$$h^{(2)}(t) = th^{(1)}(t) + \frac{\cos(\pi f_{\text{s}} t)}{\pi^2 f_{\text{s}}}. \tag{12}$$

This is the closed form expression for the BLAMP function with unit slope, shown in Fig. 3(b). Fig. 3(c) presents the BLAMP residual, which is computed as the difference between the BLAMP and the trivial ramp (9).

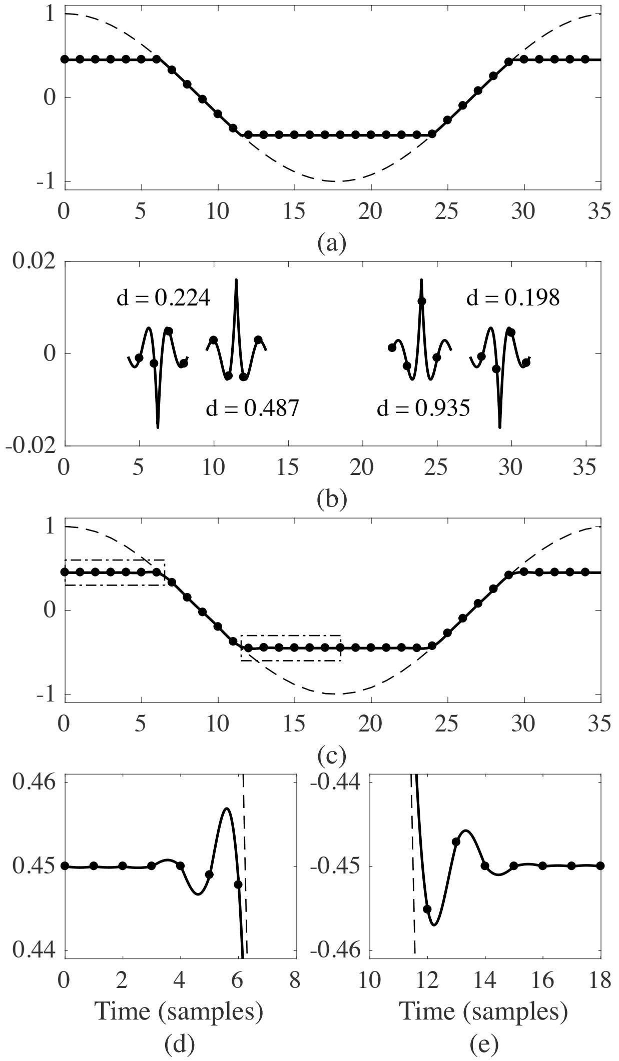

Fig. 4: One cycle of a (a) continuous-time (solid line) and discrete (dots) clipped sinusoid, (b) continuous-time (solid) and discrete (dots) truncated BLAMP residual functions scaled by the slope, inverted for positive portions of the signal and centered at each clipping point using fractional delay d (see Sec. IV), and (c) approximated bandlimited signal that results from adding the discrete signals in (a) and (b). Plots (d) and (e) show close-up views of the sections delimited by the boxes in (c).

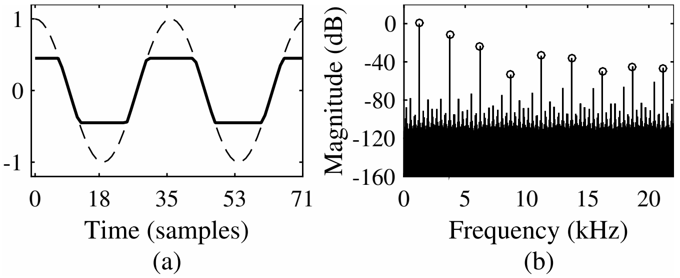

Fig. 5: (a) Waveform and (b) magnitude spectrum of a clipped sinewave after eight-point BLAMP correction, showing an approximately 15-dB SNR improvement over its trivial implementation [cf. Figs. 1(a) and (b)].

To appreciate the steps necessary to use the BLAMP residual to reduce aliasing we revisit the example of the clipped sinusoid with the same parameters as in Fig. 1. Fig. 4(a) shows the first period of this signal after hard clipping. The first step of the process is to identify the exact clipping points. These points will most likely not coincide with the system sampling intervals and must be estimated. This issue is further discussed in Sec. IV. In this example, the clipping points were derived analytically since the input signal (7) is known.

Once the necessary parameters are available, the correction function (the BLAMP residual) is centered around every clipping point and scaled by the absolute value of its corresponding slope. This step is illustrated in Fig. 4(b). Due to our original definition of the ramp function, the polarity of the correction function must be inverted for positive portions of the waveform. Then, the correction function is sampled at the nearest sample points (two on each side of each clipping point for this example) and the resulting values are added to the clipped signal. This process introduces ripple similar to that seen in the bandlimited clipped sinewave [see Fig. 2(b)], as detailed in Figs. 4(d) and 4(e).

Figs. 5(a) and 5(b) show the waveform and magnitude spectrum of a 1-second clipped sinewave processed using the aforementioned method but with 8-point correction, i.e. four points on each side of every clipping point. As before, all clipping points and their respective slopes were computed analytically in order to minimize the effects of estimation errors in the proposed method. The level of spurious frequency components has been significantly reduced in comparison with its trivial implementation [see Fig. 1(b)]. For instance, in Fig. 1(b) the level of the most prominent alias component below the fundamental, with a frequency of 525 Hz, is -60.3 dB. After correction, the level of this component was reduced to approximately -78.0 dB, as seen in Fig. 5(b).

To further evaluate the performance of the BLAMP method, the signal-to-noise ratio (SNR) of the test signal was measured. For the purpose of this study, we considered the SNR of a clipped signal $c[n]$ as the power ratio between harmonics and the aliasing components [30], computed using the following formula:

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\sum_{n=0}^{P-1} c_{\mathrm{id}}[n]^2}{\sum_{n=0}^{P-1} \varepsilon[n]^2}\right), \tag{13}$$

where $c_{\mathrm{id}}[n]$ represents the ideal alias-free version of $c[n]$, ε is the error signal, computed as $\varepsilon[n] = c[n] - c_{\mathrm{id}}[n]$, and P is the length of both signals in samples. For the case of the clipped sinewave, $c_{\mathrm{id}}[n]$ was generated by computing the discrete-time Fourier transform of the signal at odd multiples of the fundamental frequency up to the Nyquist limit (bipolar hard clipping introducing odd harmonics only). The computed magnitude and phase information was used to simulate a bandlimited version of the signal via additive synthesis. This same technique was used to compute the SNR of all the examples given in this study that involve sinusoidal and triangular

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSP.2016.2585091, IEEE Transactions on Signal Processing

waveforms. Following the aforementioned method, the SNR of the 1245-Hz clipped sinewave after BLAMP correction was measured to be 57.9 dB, which represents an improvement of 14.7 dB over the 43.2-dB SNR of trivial clipping (see Fig. 1).

The trivial triangular oscillator is a special case of an input signal that fits the assumption of the non-clipping samples having a constant slope. In this case, the overall level of aliasing reduction after BLAMP correction should be, in theory, superior to the case of the sinewave. Fig. 6 shows the waveform and magnitude spectrum of a 1-second trivially-clipped triangular signal before and after eight-point BLAMP correction. In this case, the 525-Hz alias has been attenuated by 23.3 dB, from -66.4 dB to 89.7 dB. The SNR of this signal has been increased by 17.2 dB, from 44.6 dB to 61.8 dB. These results are indeed better than in the case of the sinusoidal input. These numbers are provided as a quick reference to show the effectiveness of the proposed method. Please refer to Sec. V for a more detailed performance evaluation.

While effective, the analytic implementation of the BLAMP method is computationally costly due to the presence of the sine integral function in (12). This issue can be tackled, for instance, by precomputing the residual function and storing it as a lookup table [26]. The efficiency and accuracy of this approach will then depend on the resolution of the table and interpolation method used. In addition to this, truncation of the residual function may introduce small artifacts to the signal. While negligible, these artifacts compromise the overall quality of the signal and should be suppressed, as they may cause problems further down the processing chain. A straightforward solution to this issue would be to apply a window function to a table-based BLAMP implementation [33]. The performance of this method will then vary depending on the window of choice. This paper proposes a completely different approach based on polynomial approximations of the BLAMP, and as such, these other alternatives and their implications will not be discussed any further.

IV. Polynomial Approximations of the Bandlimited Ramp Function

In this study, the efficiency issue of the BLAMP method is addressed by following the work of Välimäki et al. [27] and proposing two polynomial approximations of the BLAMP, namely the polyBLAMP. These approximations are derived by first expressing the basis function of the BLAMP, the bandlimited impulse, as a piecewise polynomial and then repeating the steps detailed in the previous section (i.e. integrate twice and subtract the trivial ramp). The following subsections detail the derivation of these polynomials along with order-specific recommended techniques to estimate the clipping points and their respective slopes.

A. Two-Point Polynomial Correction

B-splines are a family of polynomial functions for interpolating discrete signals commonly used in signal processing [34]. Previous work on the topic of aliasing reduction in geometric periodic waveforms has proved the suitability of this family of interpolators due to their steep spectral decay [27], [35]. B-splines are symmetrical, bell-shaped functions constructed from the iterative convolution of a rectangular pulse

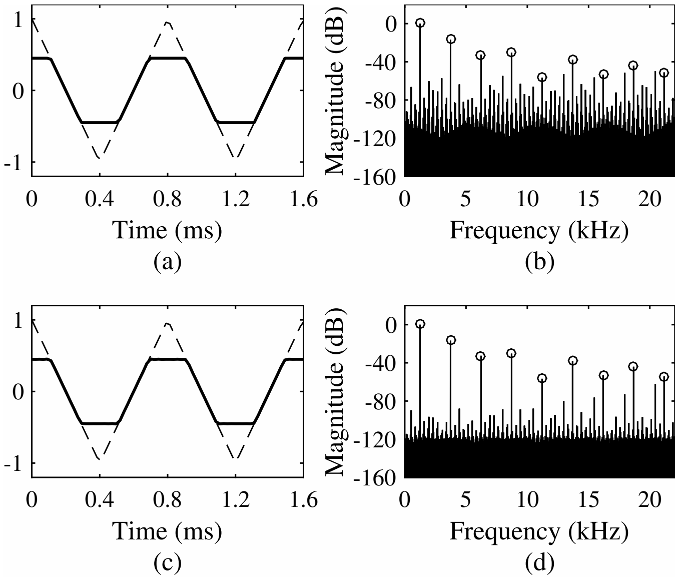

Fig. 6: (a) Waveform and (b) spectrum of a trivially clipped triangular signal and (c), (d) the same signal after eight-point BLAMP correction. This is a 17-dB improvement in SNR.

$$\beta_0(t) = \begin{cases} 1 & \text{when} \quad -T/2 < t < T/2 \\ 0.5 & \text{when} \quad |t| = T/2 \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

yielding the N^{th}-order B-spline polynomials

$$\beta_N(t) = \beta_0(t) * \beta_{N-1}(t), \quad \text{for} \quad N = 1, 2, ... \tag{15}$$

The first-order B-spline corresponds to the first-order case of every other family of interpolation polynomials, it is simply linear interpolation. Therefore, we can approximate the bandlimited continuous-time impulse (10) linearly as a triangular pulse

$$h^{(0)}_{\beta_1}(t) = \begin{cases} t + T & \text{when} \quad -T \le t < 0 \\ T - t & \text{when} \quad 0 \le t \le T \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

This function is the result of convolving the square pulse (14) with itself. Integrating this function twice and subtracting the trivial ramp function (9) would produce a cubic approximation of the BLAMP residual. This cubic polyBLAMP residual, derived from the linear approximation of the impulse function, can be used to reduce aliasing by processing two samples, one on each side of every clipping point. As before, this would be achieved by centering the residual function around each clipping point, sampling it at the nearest sample points, scaling these values by the slope and adding them to the trivially-clipped signal. Higher-order polynomial approximations of the BLAMP can be derived by double integration of higher-order B-spline basis functions. This point is explored in the following subsection of this study.

Recalling the assumption that the exact clipping points of a signal will not coincide with the system sampling intervals,

TABLE I: Linear interpolation polynomials, their first and second integrated forms, and 2-point polyBLAMP residual.

Span	Basis function: triangular pulse $(0 \leq D < 1)$
$[-T, 0]$	D
$[0, T]$	$-D + 1$
Span	First integral: polyBLEP $(0 \leq D < 1)$
$[-T, 0]$	$D^2/2$
$[0, T]$	$-D^2/2 + D + 1/2$
Span	Second integral: 2-point polyBLAMP $(0 \leq D < 1)$
$[-T, 0]$	$D^3/6$
$[0, T]$	$-D^3/6 + D^2/2 + D/2 + 1/6$
Span	2-point polyBLAMP residual $(0 \leq d < 1)$
$[-T, 0]$	$d^3/6$
$[0, T]$	$-d^3/6 + d^2/2 - d/2 + 1/6$

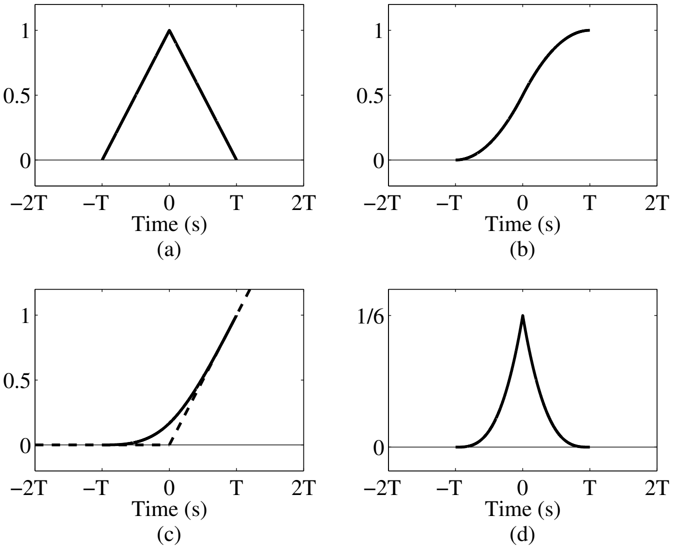

Fig. 7: (a) Triangular pulse, (b) its first integral, (c) second integral (solid line) and trivial ramp (dashed line), and (d) their difference, the polyBLAMP residual. Cf. Table I.

centering the residual function about an arbitrary number of samples can be seen as equivalent to delaying it by $D = D_{\text{int}} + d$ samples, where $D_{\text{int}} \in \mathbb{Z}_{\geq} 0$ and $d \in [0, 1)$ are the integer and fractional parts of this delay [36], respectively. In the 2-point approach, where the function has to be centered around two samples only, $D_{\text{int}} = 0$ and the delay is reduced to $D = d$. Therefore, the 2-point polyBLAMP residual can be expressed in terms of d, the fractional delay required to center it around any given clipping point. To do so, we begin by expressing the triangular basis function (16) with the first-order interpolation coefficients shown on the top two rows of Table I. A two-tap FIR filter with these coefficients will shift the input signal by D samples. The coefficients for the first and second integrals of the linear approximation of the bandlimited impulse, and the 2-point polyBLAMP residual are shown in Table I. Fig. 7 shows the functions that result from each one of these intermediate steps. The integration constants of these polynomials have been adjusted to generate continuous functions.

In most practical scenarios, the fractional delay and slope associated with each clipping point will not be available analytically. For the case of 2-point correction, linear interpolation can also be used to estimate the slope parameter. Assuming a linear behavior and access to the unclipped signal, the slope μ at a clipping point can be estimated using a *first-difference* filter [37]

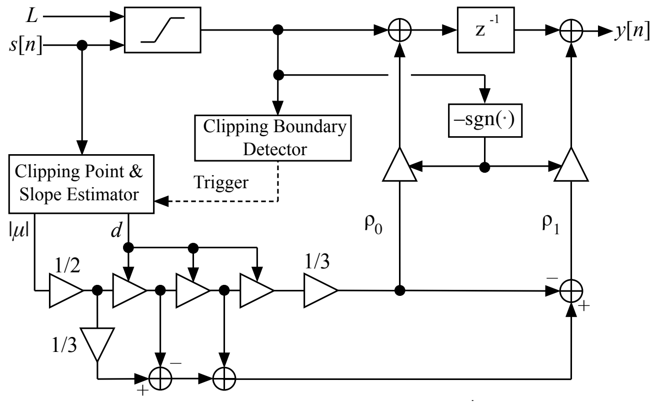

Fig. 8: Signal processing diagram to implement the 2-point polyBLAMP correction on a hard-clipped signal using the coefficients shown in Table I.

$$\mu[n_\text{a}] = s[n_\text{b}] - s[n_\text{a}], \tag{17}$$

where n_a and n_b are the sample indices of the signal before and after any arbitrary fractional clipping point, i.e. the *clipping boundaries*. Due to the assumption of a constant slope, the estimate of the slope of the signal at the clipping point will be the same as the slope at the boundaries.

The next step is to fit a straight line to the clipping boundaries and solve the intersection with the clipping threshold $\pm L$ (the polarity of L will depend on the polarity of the clipping point). The exact clipping point can then be estimated as

$$n_\text{a} + d = \frac{\mu[n_\text{a}]n_\text{a} - s[n_\text{a}] + \text{sgn}(s[n_\text{a}])L}{\mu[n_\text{a}]}. \tag{18}$$

Since the data points are evenly spaced, we can simplify this expression by creating a virtual time shift and assuming $n_\text{a} = 0$ and $n_\text{b} = 1$, while $s[n_\text{a}]$ and $\mu[n_\text{a}]$ remain unchanged. The expression is then simplified as

$$d = \frac{\text{sgn}(s[n_\text{a}])L - s[n_\text{a}]}{\mu[n_\text{a}]}. \tag{19}$$

In summary, the 2-point polyBLAMP correction method is implemented by first detecting the clipping points of a signal and identifying their boundaries. For each set of clipping boundaries, the associated slope and fractional delay are estimated using (17) and (19), respectively. These parameters are then used to compute the polyBLAMP correction coefficients, which are added to the clipped signal at the boundaries. The polarity of these correction coefficients will always depend on the polarity of the clipping point being processed.

Figure 8 shows the signal processing diagram required to implement the 2-point polyBLAMP method for hard clipping, where

$$\rho_0 = |\mu| \, (d^3/6), \tag{20}$$

TABLE II: Two-point polyBLAMP algorithm.

```
**Input:** Signal $x$, Clipping threshold $L$
**Output:** Signal $y$
*Initialisation*:
$xlen \leftarrow$ length of $x$
$flg, flg\_n1$    // Clipping flags
$y\_n, y\_n1, x\_n1 \leftarrow 0$    // State variables
*LOOP Process*:
**for** $i = 0$ **to** $xlen - 1$ **do**
  **if** $|in[i]| \geq L$ **then**
    $flg \leftarrow 1$
    $y\_n \leftarrow \mathrm{sgn}(x[i])L$
  **else**
    $flg \leftarrow 0$
    $y\_n \leftarrow x[i]$
  **end if**
  **if** $(flg - flg\_n1) \neq 0$ **then**    // Clipping boundary detection
    $m \leftarrow x[i] - x\_n1$
    $d \leftarrow (\mathrm{sgn}(x\_n1)L - x\_n1)/m$
    $p1 \leftarrow -d^3/6 + d^2/2 - d/2 + 1/6$
    $p0 \leftarrow d^3/6$
    $y\_n1 \leftarrow y\_n1 - \mathrm{sgn}(x\_n1)\mathrm{abs}(m)p1$
    $y\_n \leftarrow y\_n - \mathrm{sgn}(x\_n1)\mathrm{abs}(m)p0$
  **end if**
  $y(i) \leftarrow y\_n1$
  $y\_n1 \leftarrow y\_n$    // Update state variables
  $x\_n1 \leftarrow x[i]$
  $flg\_n1 \leftarrow flg$
**end for**
**return** $y$
```

and

$$\rho_1 = |\mu|\,(-d^3/6 + d^2/2 - d/2 + 1/6). \tag{21}$$

This design has been optimized to reduce the number of operations required by identifying the factors that are common to both polyBLAMP residual coefficients. The parameter estimators are shown as black boxes in Fig. 8 to emphasize their independence from the actual polyBLAMP method.

Table II shows pseudocode for the 2-point polyBLAMP method using (17) and (19) to estimate parameters μ and d, respectively. In this example, the "Clipping boundary detection" stage (see Fig. 8) is implemented using a control variable (flg) and assigning to it a value of 1 for clipping samples and a value of 0 for non-clipping samples. At every time step, the difference between the current and the previous control variable (flg_n1) is evaluated to identify discontinuities in the first derivative of the signal. A non-zero value resulting from this evaluation means the signal is either entering or leaving the saturating part of the clipping function. Due to the nature of the method, a latency of one sample is required to perform this clipping point detection and to compute the necessary parameters. Further comments on the latency and computational costs of the algorithm are included in Sec. V.

Figure 9 shows the results obtained from processing the 1245-Hz sinusoidal and triangular clipped waveforms following the design in Fig. 8 and using (17) and (19) to estimate the necessary correction parameters (cf. Table II). These results show a clear improvement in terms of aliasing suppression. For instance, considering once again the case of the 525-Hz alias (see Sec. III), it has now been reduced to -80.2 dB and -142 dB for the sinusoidal and triangular signals, respectively [see Figs. 9(b) and 9(d)]. These new levels represent an attenuation of 19.9 dB and 75.8 dB for each case. In terms of overall SNR, it has been increased by 12.6 dB and 13.7 dB for each signal. These results are considerably close to those of the eight-point BLAMP (see Sec. III) but with minimal computational costs (see Sec. V). Another important advantage of the 2-point polyBLAMP over the BLAMP is the lack of overshoot. After aliasing correction, the clipping threshold is preserved and all signal values remain within $[-L, L]$. This can be attributed to the range of the correction function [see Fig. 7(d)] which has a minimum value of zero.

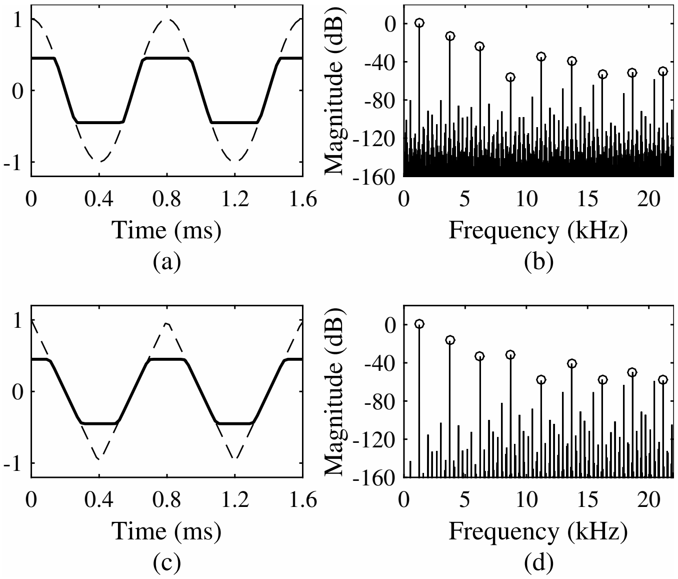

Fig. 9: Two-point polyBLAMP correction examples: Enhanced clipped (a) sinusoidal and (c) triangular signals and (b), (d) their respective spectra. Cf. Fig. 1(a-b) and Fig. 6(a-b).

B. Four-Point Polynomial Correction

The polyBLAMP method can be extended to correct two samples at each side of every clipping point. The coefficients for this 4-point polyBLAMP are derived from the second integral of the third-order B-spline basis function $\beta_3(t)$. In this case, centering the 4-point polyBLAMP residual around four samples is equivalent to a shift of $D = D_{\text{int}} + d$ samples, where $D_{\text{int}} = 1$. The polynomial coefficients for third-order B-spline interpolation, their first and second integrals, and the 4-point polyBLAMP residual are shown in Table III. A variable change was implemented to express the residual function in terms of d, the fractional part of D. Fig. 10 shows the functions that result from each one of these intermediate steps.

Since the polyBLAMP method is completely independent from the fractional delay and slope estimation process, these parameters can once again be derived as described in the previous subsection, i.e. using (17) and (19). However, extending

TABLE III: Third-order B-spline basis function polynomials, their first and second integrated forms, and polyBLAMP residual ($1 \leq D < 2$ and $0 \leq d < 1$).

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: 4-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: 4-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
$[-2T, T]$	$d^5/120$
$[-T, 0]$	$-d^5/40 + d^4/24 + d^3/12 + d^2/12 + d/24 + 1/120$
$[0, T]$	$d^5/40 - d^4/12 + d^2/3 - d/2 + 7/30$
$[T, 2T]$	$-d^5/120 + d^4/24 - d^3/12 + d^2/12 - d/24 + 1/120$

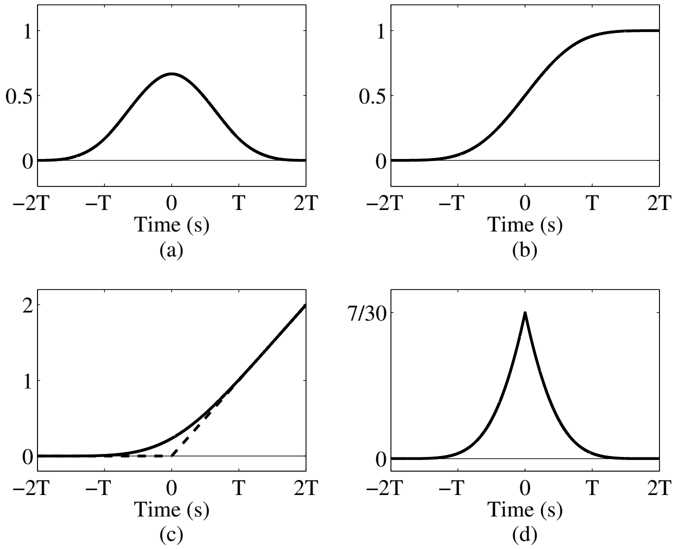

Fig. 10: (a) Third-order B-spline basis function, (b) its first integral, (c) second integral (solid line) and trivial ramp (dashed line), and (d) their difference, the 4-point polyBLAMP residual. Cf. Table III.

the range of the correction to four points means the impact of inaccurate estimation of these parameters will be higher than in the 2-point case. Therefore, it is preferable to implement the 4-point polyBLAMP method with a higher-order parameter estimation process that takes full advantage of all the available data points. One way to achieve this is by fitting a cubic polynomial to these four points and using Newton-Raphson's method to estimate the fractional clipping point. Evaluating the derivative of the cubic polynomial at this value will yield an estimate of the slope at this point.

Considering any arbitrary clipping point within the signal and recalling our definition of the clipping boundaries, we begin by considering the indices of the four samples that will be processed by the algorithm, i.e. $n_\mathrm{a} - 1$, n_a, n_b, and $n_\mathrm{b} + 1$. The aim is to fit a polynomial of the form $f(D) = aD^3 + bD^2 + cD + e$ to the unclipped signal at these four points. Lagrangian interpolation can be used to find the closed form expressions for coefficients a, b, c, and e. Given that the data points are evenly spaced, these coefficients can be written as

$$\begin{aligned}
a &= -\frac{1}{6}s[n_\mathrm{a} - 1] + \frac{1}{2}s[n_\mathrm{a}] - \frac{1}{2}s[n_\mathrm{b}] + \frac{1}{6}s[n_\mathrm{b} + 1] \\
b &= s[n_\mathrm{a} - 1] - \frac{5}{2}s[n_\mathrm{a}] + 2s[n_\mathrm{b}] - \frac{1}{2}s[n_\mathrm{b} + 1] \\
c &= -\frac{11}{6}s[n_\mathrm{a} - 1] + 3s[n_\mathrm{a}] - \frac{3}{2}s[n_\mathrm{b}] + \frac{1}{3}s[n_\mathrm{b} + 1] \\
e &= s[n_\mathrm{a} - 1].
\end{aligned}$$

Once the coefficients of the cubic polynomial are known, the next step is to obtain the intersection of this curve with the clipping threshold L. This inverse interpolation problem is equivalent to solving the following equation for D:

$$aD^3 + bD^2 + cD + e + \mathrm{sgn}(s[n_\mathrm{a}])L = 0, \tag{22}$$

where the polarity of the clipping threshold L will once again depend on the polarity of the clipping point. A solution to (22) can be estimated using Newton-Raphson's iterative method, defined as

$$D_{q+1} = D_q - \frac{f(D_q)}{f'(D_q)}, \tag{23}$$

where $q = 0, 1, 2, ..., Q - 1$, Q is the number of iterations required for $f(D_q)/f'(D_q)$ to become zero or a predetermined threshold at which it can be neglected, and D_0 is an initial guess [38]. Since we know that the solution to (22) will be between [1,2) due to the restriction on D, an appropriate initial guess would be $D_0 = 1.5$.

Coming back to (22), we can then estimate the clipping point by iterating over

$$D_{q+1} = D_q - \frac{aD_q^3 + bD_q^2 + cD_q + e + \mathrm{sgn}(s[n_\mathrm{a}])L}{3aD_q^2 + 2bD_q + c}. \tag{24}$$

The resulting value D_Q represents the fractional delay associated with the clipping point. The slope at this point is obtained as a byproduct of Newton-Raphson's method, which for the sake of completeness is given as

$$\mu(D_Q) = 3aD_Q^2 + 2bD_Q + c. \tag{25}$$

Finally, the value of d can be computed as $d = D_Q - 1$. This represents an estimated clipping point at sample $n_\mathrm{a} + d$, i.e. $|s[n_\mathrm{a} + d]| = L$. Just as with the 2-point polyBLAMP, this process has to be repeated for every set of clipping boundaries in the signal. The estimated values for μ and d are then used to compute the four correction coefficients, which are added to the clipped signal at the corresponding sample points. As

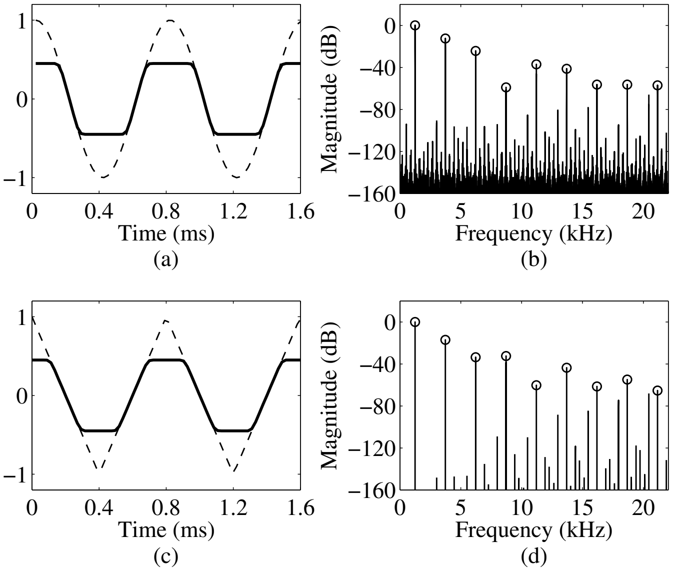

Fig. 11: Four-point polyBLAMP correction example: Enhanced clipped (a) sinusoidal and (c) triangular signals and (b), (d) their respective spectra.

before, the polarity of the correction coefficients has to be adjusted depending on the polarity of the clipping point.

Figure 11 shows the waveforms and magnitude spectra for the 1245-Hz sinusoid and trivial triangular waveform after 4-point polyBLAMP correction implemented using the aforementioned parameter estimation techniques. The results obtained are considerably better than those of the BLAMP (see Figs. 5 and 6) and 2-point polyBLAMP methods (see Fig. 9). For each respective signal, SNR has been increased by 22.5 dB and 23.4 dB. The increased computational costs associated with this boost in the performance of the method are discussed in the next section of this study. As in the 2-point case, this method does not introduce overshoot and the clipping threshold is preserved. This can once again be attributed to the range of the residual function, which will always be non-negative, as seen in Fig. 10(d).

V. Evaluation of the Proposed Algorithms

The performance of the 2-point and the 4-point polyBLAMP methods was evaluated in terms of effectiveness (improved signal quality) and efficiency (computational costs and input-output latency). Oversampling was used as a reference method since it is the only previous technique available to reduce aliasing in clipping algorithms.

In oversampling, the input signal is first *upsampled* by inserting $\nu - 1$ zeros between each sample, where ν is the oversampling factor. Then, an interpolation filter is used to smooth the discontinuities and suppress image spectra. The upsampled signal is then processed by the nonlinearity (the hard clipper in this case), which distorts the waveform causing expansion of its spectrum. The processed signal is then lowpass filtered to remove the newly generated spectral components. Finally, the signal is *downsampled* back to the original sample rate. This process suppresses many of the spectral components which will alias.

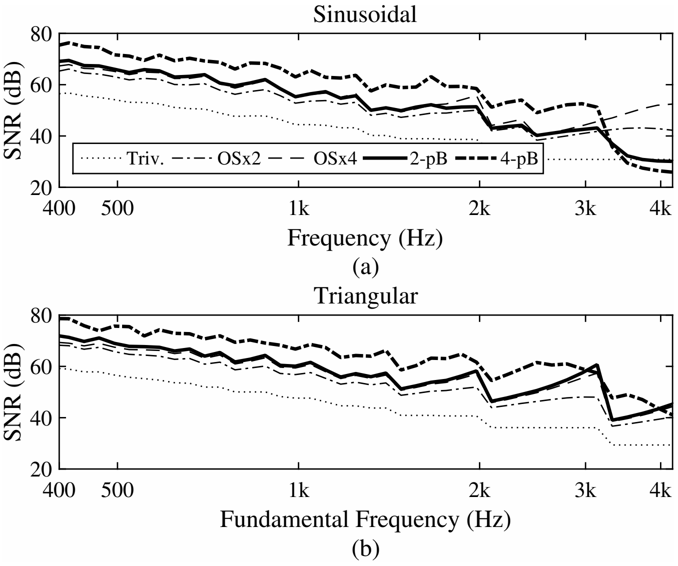

Fig. 12: SNR after hard clipping without further processing (Triv.), with oversampling (OS) by $\nu = 2$ and $\nu = 4$, and with 2-point and 4-point polyBLAMP (pB) correction for (a) sinusoidal and (b) triangular test signals.

Two oversampling factors were implemented for the evaluation: $\nu = 2$ and $\nu = 4$. To minimize computational costs, low-order linear-phase lowpass FIR filters were chosen. The z-domain transfer functions of the filters used are

$$H_2(z) = 0.5 + z^{-1} + 0.5z^{-2} \tag{26}$$

and

$$\begin{aligned} H_4(z) = 0.25 + 0.5z^{-1} + 0.75z^{-2} + z^{-3} \\ + 0.75z^{-4} + 0.5z^{-5} + 0.25z^{-6} \end{aligned} \tag{27}$$

for the cases $\nu = 2$ and $\nu = 4$, respectively. In each case, the same filter was both used after upsampling and before downsampling.

A. SNR Improvement

To evaluate the performance of the four methods in terms of aliasing suppression, the SNR seen at their output was measured for various sinusoidal and triangular test signals. Obtained results were compared against those from trivial hard clipping. The frequency range chosen for the performed tests was from 27.5 to 4186 Hz, the fundamental frequency range of a standard 88-key piano.

Fig. 12 shows the SNR measurements for sinusoidal and triangular test signals, respectively. Results for fundamental frequencies below 400 Hz have been omitted since in these cases aliasing is not as dramatic as it is at higher fundamental frequencies. Overall, these results show the superiority of the polyBLAMP method over oversampling by low factors, and for fundamental frequencies below about 3 kHz. Oversampling by factor 2 increases the SNR by 9.2 dB and 9.5 dB on average, in comparison to trivial hard clipping, for the sinusoidal and triangular test signals, respectively. It can be seen in Fig. 12 that the 2-point polyBLAMP method and oversampling by

TABLE IV: Average SNR improvement for sinusoidal and triangular signals for fundamental frequencies between 400 Hz and 3100 Hz in Fig. 12. The best results are highlighted.

	Oversampling		PolyBLAMP	
Input	$\nu = 2$	$\nu = 4$	Two-Point	Four-Point
Sinusoidal	9.2 dB	11.9 dB	11.8 dB	**19.5 dB**
Triangular	9.5 dB	12.5 dB	13.2 dB	**20.4 dB**

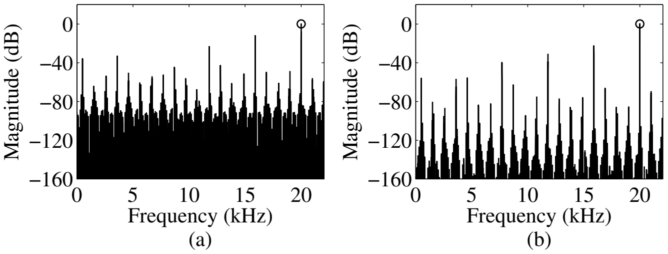

Fig. 13: Magnitude spectrum of a 20-kHz clipped sinewave (a) before and (b) after using 4-point polyBLAMP enhancement with analytic clipping point and slope values. This example shows that with improved estimation methods the polyBLAMP method can act successfully for very high-frequency signals.

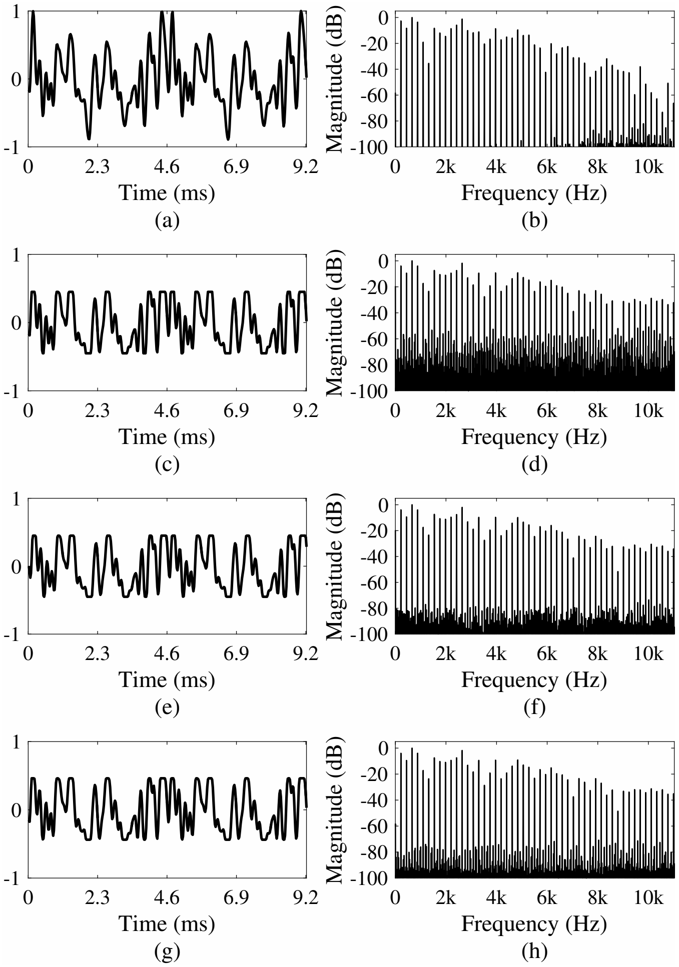

Fig. 14: Waveform and spectrum of harmonic test signal ($f_0 = 660$ Hz): (a)-(b) Original, (c)-(d) trivially hard-clipped, (e)-(f) oversampled by a factor 4, and (g)-(h) polyBLAMP-enhanced (4-point) after hard clipping.

factor 4 reach a very similar improvement, about 12 dB for the sinusoidal and about 13 dB for the triangular signals. The 2-point polyBLAMP method achieves the best results, approximately a 20-dB improvement in both cases. The average SNR improvements of the four methods presented in Fig. 12 are collected in Table IV. Similar results were obtained for other tested clipping thresholds L.

At fundamental frequencies higher than about 3 kHz, estimation of the clipping points and their respective slopes becomes inaccurate, impacting the performance of the polyBLAMP method, see Fig. 12. Nevertheless, if the clipping point and slope parameters are obtained analytically or are estimated using more robust techniques than the ones suggested in this paper, the polyBLAMP method can be implemented for input signals with higher fundamental frequencies. Figs. 13(a) and (b) show an example of the magnitude spectra of a 20-kHz clipped sinewave before and after 4-point polyBLAMP correction, respectively. The correction was implemented by computing the clipping points and their slopes analytically, yielding a total SNR improvement of 10.2 dB.

Furthermore, to demonstrate that the polyBLAMP method is not restricted to sinusoidal and triangular waveforms, its performance on a clipped synthetic guitar sound was also examined. This signal was selected due to its relatively stationary spectrum and low background noise, which allows a clean visual representation of the correction process. Figs. 14(a) and (b) show the waveform and magnitude spectrum of 1 second of the tested signal, showcasing its rich harmonic content. Even when the signal contains frequency components that extend well above the optimal working range of the polyBLAMP (cf. Fig. 12), its fundamental frequency is fairly low and falls within this optimal range. Therefore, aliasing after hard clipping can be reduced following the proposed method. Figs. 14(g) and (h) show the spectra of the clipped signal before and after 4-point polyBLAMP correction. As a reference, Figs. 14(e)-(f) show the results obtained from oversampling by a factor 4 prior to clipping. In both cases, some spurious components have been attenuated by up to 20 dB.

As before, the quality of these signals was measured comparing their SNRs before and after correction. Due to the time-varying nature of the harmonic signal chosen for the evaluation, it was not possible to proceed with the same approach used when evaluating the SNR of the sinusoidal and triangular test signals (see Fig. 12). Instead, we used *BSS_EVAL*, a MATLAB toolbox used to measure the performance of Blind Audio Source Separation (BSS) algorithms designed by Vincent et al. [39] and available online under a GNU Public License [40]. This toolbox estimates the signal-to-distortion ratio (SNR in our case) of a signal by comparing it to a so-called “true source”, which in this example would be an ideal alias-free version of the clipped string sound. An ideal signal was approximated by oversampling the input signal by a factor of 100 prior to clipping. The SNR for the trivially-clipped, with oversampling by 4 and after 4-point polyBLAMP correction were estimated to be 45.3 dB, 56.2 dB, and 58.4 dB, respectively. These numbers represent an SNR improvement of approximately 13.1 dB for the 4-point polyBLAMP method [see Fig. 14(h)] over the trivial implementation of the algorithm [see Fig. 14(d)]. The proposed method also outperforms oversampling by 4 while also minimizing computational costs, as discussed in the next subsection of this study.

Audio examples of clipped time-varying signals with and without polyBLAMP correction can be accessed at http://research.spa.aalto.fi/publications/papers/ieee-tsp-2016-clipping/. For each test signal a trivially-clipped version is compared with the four discussed methods: oversampling by 2 and 4, and with 2- and 4-point polyBLAMP correction. In order to facilitate the identification of the removed aliasing components, the residual signals for each method are also provided. These residuals are computed as the difference between each corrected signal and the trivial version, obtained by taking into account the latency of each algorithm and synchronizing the signals. For the case of oversampling, the residuals also show a severe loss of high-frequency content. This issue is also minimized in the polyBLAMP method. Overall, antialiased signals are perceived as being softer and less disturbing. The harsh and *tinny* timbre of the trivially-clipped sounds is reduced. The improvement of alias reduction is easiest to hear in harmonic signals, when the fundamental frequency is changing, because some aliases move in the opposite direction in frequency. Non-tonal components, such as attack transients of electric guitar tones, are also enhanced by the proposed method, but it is less obvious to perceive.

B. Computational Costs and Latency

The four methods were evaluated for efficiency by measuring their average run time for three different test signals: a low and a high-frequency sinusoid, and the same synthetic string sound used in the previous evaluation. The algorithms were ported from MATLAB to Python and tested using an Apple iMac with an Intel Core i5 (2.7 GHz) processor with 16 GB of 1600 MHz DDR3 RAM. All other processes were terminated and the network connection was shut down to avoid distractions in the execution of the Python scripts. Obtained results, which are an average of 20 test runs, are shown in Table V.

The 2-point polyBLAMP proved to be an outstanding technique in terms of its cost-benefit analysis. Within its optimal range of operation, it provides better aliasing reduction than oversampling by factor 4 with minimal computational costs. As expected, the 4-point polyBLAMP method is considerably more expensive than its 2-point counterpart but is still cheaper than oversampling by factor 4, as seen in Table V. The costs associated with the polyBLAMP methods are signal-dependent; i.e. the number of operations per sample required depends on the number of clipping points found within the signal being corrected. Due to the corrective nature of the polyBLAMP methods, they avoid the redundant operations associated with oversampling. In general, signals with a high-frequency content are most likely to have more clipping points in a given time frame than low-frequency signals, as shown in Table V. Still, for the case of the harmonic signal example (cf. Fig. 14), better results than those of oversampling by 4 were obtained at nearly one-third of the costs.

TABLE V: Averaged computation time (in milliseconds) for oversampling by factors of ν = 2 and ν = 4, and the 2-point and 4-point polyBLAMP methods.

	Oversampling		PolyBLAMP	
Input	$\nu = 2$	$\nu = 4$	Two-Point	Four-Point
100-Hz Sinewave	264 ms	780 ms	**64.1 ms**	104 ms
3-kHz Sinewave	263 ms	784 ms	**128 ms**	686 ms
Harmonic Signal	261 ms	788 ms	**69.1 ms**	271 ms

In real-time signal processing, it is important to consider the input-output latency caused by the algorithm. This corresponds to the required number of past input samples, which must be analyzed/processed before the algorithm is ready to produce its first output sample. In the case of the 2-point polyBLAMP method, the latency is just 1 sample, because the computing of the slope and clipping point can be determined after the first sample in the clipped region has arrived, and then the algorithm corrects the sample right before the clipping point. The 4-point polyBLAMP method generates 3 samples of latency, since it needs two samples after and two samples before the clipping point, and must then go back to first correct the signal value two samples before the clipping point.

For the oversampling methods, the latency is effectively caused by the group delay of the interpolation and decimation filters. In our test case, the oversampling methods by factor 2 and 4 use their respective filters (26) and (27) twice. As the group delay of the linear-phase filters is 1 and 3 samples, they cause a latency of 2 and 6 samples, respectively.

The proposed enhancement methods are thus preferable also in terms in latency, because they only delay the input signal by 1 or 3 samples. Additionally, both polyBLAMP methods avoid overshoot of the signal around the clipping points, an inherent issue in oversampling.

VI. Conclusion

The enhancement of clipped signals containing aliasing disturbances has been studied. A new method based on the bandlimited ramp function, or BLAMP, was introduced. This method modifies a few samples around each clipping point in the signal waveform, thus reducing the aliasing while retaining the signal content. In this paper, only symmetrical clipping was considered, but the method can be applied similarly to asymmetrically clipped waveforms.

For practical implementations, polynomial approximations based on B-spline basis functions, which correct two or four samples per clipping point, were given. The polyBLAMP method was compared against oversampling and was shown to be superior. The 2-point correction method improves the

SNR of clipped sinewaves by 12 dB, which is comparable to the enhancement obtained with oversampling by a factor 4, but with minimal computational costs. The 4-point correction method increases the SNR by about 20 dB in the case of both clipped sinusoidal and triangular signals.

The proposed methods work best for signals obeying the assumption of a piecewise ramp-like waveform, such as a trivial triangular signal fed through a hard clipper. However, excellent results were also obtained in tests using sinusoidal and harmonic input signals. Advantages of the proposed polynomial methods include their modest computational cost, lack of overshoot in the output waveform, and small input-output latency. These methods can be used to reduce disturbances in hard clipped signals for example in simulations of analog electronics. They can also be used as a pre-processing technique in alias-reduced soft clipping algorithms [19].

It should be noted that the polyBLAMP method does not depend on the availability of the original unclipped signal. However, the process of estimating the fractional delay associated with each clipping point is simplified considerably when this condition is met, as was assumed in this paper. In cases where access to the original unclipped signal is not available, oversampling is no longer a suitable method and the polyBLAMP becomes the only option to reduce aliasing. Therefore, the challenge of blind clipping point estimation without access to the original unclipped samples is a relevant research topic and is left as future work.

REFERENCES

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

[2] H.-M. Lehtonen, J. Pekonen, and V. Välimäki, "Audibility of aliasing distortion in sawtooth signals and its implications for oscillator algorithm design," *J. Acoust. Soc. Am.*, vol. 132, no. 4, pp. 2721–2733, Oct. 2012.

[3] J. Abel and J. Smith, "Restoring a clipped signal," in *Proc. Int. Conf. Acoust, Speech Signal Process. (ICASSP-91)*, vol. 3, Toronto, ON, Apr. 1991, pp. 1745–1748.

[4] A. Adler, V. Emiya, M. Jafari, M. Elad, R. Gribonval, and M. Plumbley, "Audio inpainting," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 20, no. 3, pp. 922–932, Mar. 2012.

[5] B. Defraene, N. Mansour, S. De Hertogh, T. van Waterschoot, M. Diehl, and M. Moonen, "Declipping of audio signals using perceptual compressed sensing," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 12, pp. 2627–2637, Dec. 2013.

[6] K. Siedenburg, M. Kowalski, and M. Dörfler, "Audio declipping with social sparsity," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP-14)*, Florence, Italy, May 2014, pp. 1577–1581.

[7] S. Kitić, N. Bertin, and R. Gribonval, "Sparsity and cosparsity for audio declipping: A flexible non-convex approach," in *Proc. 12th Int. Conf. Latent Variable Analysis and Signal Separation (LVA/ICA 2015)*, Liberec, Czech Republic, Aug. 2015.

[8] C. Bilen, A. Ozerov, and P. Perez, "Audio declipping via nonnegative matrix factorization," in *Proc. IEEE Workshop Appl. Signal Process. to Audio and Acoust. (WASPAA-15)*, New Paltz, NY, Oct. 2015.

[9] L. Claesson, "Making audio sound better one square wave at a time," in *Proc. Audio Eng. Soc. 137th Conv.*, Los Angeles, CA, Oct. 2014.

[10] H. Künzel and P. Alexander, "Forensic automatic speaker recognition with degraded and enhanced speech," *J. Audio Eng. Soc.*, vol. 64, no. 4, pp. 244–253, Apr. 2014.

[11] 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.

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

[13] P. Dutilleux, K. Dempwolf, M. Holters, and U. Zölzer, "Nonlinear processing," in *DAFX: Digital Audio Effects*, 2nd ed., U. Zölzer, Ed. Chichester, UK: Wiley, 2011, pp. 101–138.

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

[15] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, "Virtual analog effects," in *DAFX: Digital Audio Effects*, 2nd ed., U. Zölzer, Ed. Chichester, UK: Wiley, 2011, pp. 473–522.

[16] 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.

[17] D. T. Yeh, "Digital implementation of musical distortion circuits by analysis and simulation," PhD thesis, Stanford University, Stanford, CA, June 2009.

[18] ——, "Automated physical modeling of nonlinear audio circuits for real-time audio effects—Part II: BJT and vacuum tube examples," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 4, pp. 1207–1216, May 2012.

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

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

[21] T. Smyth, J. Abel, and J. O. Smith III, "The feathered clarinet reed," in *Proc. 7th Int. Conf. Digital Audio Effects (DAFx-04)*, Naples, Italy, Oct. 2004, pp. 95–100.

[22] J. Schattschneider and U. Zölzer, "Discrete-time models for nonlinear audio systems," in *Proc. Digital Audio Effects Workshop (DAFx-99)*, Trondheim, Norway, Dec. 1999, pp. 45–48.

[23] B. D. Man and J. D. Reiss, "Adaptive control of amplitude distortion effects," in *Proc. Audio Eng. Soc. Int. Conf.*, London, UK, Jan. 2014.

[24] J. O. Smith, *Physical Audio Signal Processing*. Palo Alto, CA: W3K Publishing, Dec. 2008.

[25] T. Stilson and J. Smith, "Alias-free digital synthesis of classic analog waveforms," in *Proceedings of the International Computer Music Conference*, Hong Kong, 1996, pp. 332–335.

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

[27] 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.

[28] 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.

[29] 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.

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

[31] 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.

[32] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Prentice Hall, Aug. 1975.

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

[34] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 6, no. 16, pp. 22–38, 1999.

[35] 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.

[36] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay—Tools for fractional delay filter design," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60, Jan. 1996.

[37] R. G. Lyons, *Understanding Digital Signal Processing*, 3rd ed. New Jersey, NJ: Prentice Hall, Nov. 2010.

[38] F. B. Hildebrand, *Introduction to Numerical Analysis*, 2nd ed. New York, NY: Dover Publications, Jun. 1987.

[39] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[40] E. Vincent, R. Gribonval, and C. Févotte. (2005) BSS_EVAL toolbox user guide. [Online]. Available: http://bass-db.gforge.inria.fr/bss_eval/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSP.2016.2585091, IEEE Transactions on Signal Processing

Fabián Esqueda was born in Mexico City, Mexico in 1989. He received the B.Eng. degree in electronic engineering with music technology systems from the University of York, York, UK, in 2012, and the M.Sc. degree in acoustics and music technology from the University of Edinburgh, Edinburgh, UK, in 2013.

He is currently a doctoral student in the Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland. His research interests include virtual analog modeling of musical systems, audio effect design, and real-time audio programming for mobile devices.

Stefan Bilbao (M'14–SM'15) was born in Montreal, Canada. He studied at Harvard University (B.A., physics, 1992), and later undertook graduate studies at Stanford University (MSc., PhD., electrical engineering, 1996 and 2001, respectively).

He is currently a Reader, and co-director of the Acoustics and Audio Group at the University of Edinburgh (Edinburgh, UK), and was previously a Lecturer at the Sonic Arts Research Centre at the Queens University Belfast. He is the PI of the European Research Council funded NESS (Next Generation Sound Synthesis) project, running jointly between the Acoustics and Audio Group and the Edinburgh Parallel Computing Centre at the University of Edinburgh from 2012 to 2016.

Vesa Välimäki (S'90–M'92–SM'99–F'15) received the M.Sc. in Technology and the Doctor of Science in Technology degrees in electrical engineering from the Helsinki University of Technology (TKK), Espoo, Finland, in 1992 and 1995, respectively.

He was a Postdoctoral Research Fellow at the University of Westminster, London, UK, in 1996. In 1997–2001, he was a Senior Assistant at TKK. In 2001–2002, he was a Professor of signal processing at the Pori unit of the Tampere University of Technology, Pori, Finland. In 2006–2007, he was the Head of the TKK Laboratory of Acoustics and Audio Signal Processing. He is currently a Professor of audio signal processing at Aalto University, Espoo, Finland. In 2008–2009, he was a Visiting Scholar at Stanford University.

Prof. Välimäki is a Fellow of the Audio Engineering Society and a Life Member of the Acoustical Society of Finland. In 2007–2013 he was a Member of the Audio and Acoustic Signal Processing Technical Committee of the IEEE Signal Processing Society and is currently an Associate Member. He is a Founding Member of the EURASIP Special Area Team in acoustic, sound and music signal processing (2015–). He served as an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS in 2005–2009 and of the IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING in 2007–2011. He was in the Editorial Board of the *Research Letters in Signal Processing* and of the *Journal of Electrical and Computer Engineering*. He was the Lead Guest Editor of a special issue of the IEEE SIGNAL PROCESSING MAGAZINE in 2007 and of a special issue of the IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING in 2010. In 2015, he was a Guest Editor of the special issue of the IEEE SIGNAL PROCESSING MAGAZINE on signal processing techniques for assisted listening. In 2008, he was the Chair of DAFX-08, the 11th International Conference on Digital Audio Effects. Since 2015 he has been a Senior Area Editor of the IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING. He is an Editorial Board Member of *The Scientific World Journal* and the Lead Guest Editor of the special issue of *Applied Sciences* on audio signal processing.