ALIASING REDUCTION IN SOFT-CLIPPING ALGORITHMS

Fabián Esqueda[*], *Vesa Välimäki*

Aalto University School of Electrical Engineering
Dept. of Signal Processing and Acoustics
Espoo, Finland

Stefan Bilbao

Acoustics and Audio Group
University of Edinburgh
Edinburgh, UK

ABSTRACT

Soft-clipping algorithms used to implement musical distortion effects are major sources of aliasing due to their nonlinear behavior. It is a research challenge to design computationally efficient methods for alias-free distortion without oversampling. In the proposed approach, soft clipping is decomposed into a hard clipper and a low-order polynomial part. A technique for aliasing reduction of the hard-clipped signal is presented based on a polynomial approximation of the bandlimited ramp function. This correction function operates by quasi-bandlimiting the discontinuities introduced in the first derivative of the signal. The proposed method effectively reduces perceivable aliasing in soft-clipped audio signals having low frequency content. This work presents the first step towards alias-free implementations of nonlinear virtual analog effects.

Index Terms— Audio signal processing, antialiasing, music, nonlinear distortion

1. INTRODUCTION

Music technology is one of the few fields of electronics in which digital signal processing has not yet managed to fully displace traditional analog systems. Musicians still tend to favor analog audio equipment over their digital counterparts due to their alleged distinctive tonal qualities [1]. However, analog systems also have the disadvantage of being expensive, bulky and hard to acquire. Therefore, it is desirable to emulate their behavior in the digital domain, where more flexibility is available and costs can be significantly reduced [2].

Distortion is an example of a popular audio effect commonly implemented using analog circuitry [1]. Historically, guitarists would achieve distortion by operating vacuum tube amplifiers at high gain levels, causing the output to saturate [1]. This saturation would cut off portions of the waveform above or below certain threshold—a process known as *clipping*. Signal clipping is a nonlinear operation and thus introduces frequency components not present in the original signal. In the digital domain, when the frequencies of these components exceed the Nyquist limit, the components are reflected back into the baseband, causing *aliasing*. In this work, we will concentrate exclusively on static memoryless distortion algorithms.

Aliasing causes severe undesired distortion, inharmonicity and beating. Nevertheless, if the aliased components are sufficiently attenuated, they become inaudible [3]. Currently available methods to prevent aliasing in nonlinear systems include oversampling and the harmonic mixer [4]. However, these techniques represent a considerable increase in computational costs [5]. For instance, sampling rates in the range of several MHz are necessary to remove aliasing above -100 dB relative to full scale [6].

Aliasing in clipped signals can be attributed to the discontinuities introduced in the derivatives of the signal, which require an infinite bandwidth to be represented digitally. A similar problem arises in the field of digital subtractive synthesis, where periodic signals of a simple geometric form are typically used as source waveforms [3]. Since these signals are inherently discontinuous, synthesizing them trivially generates aliasing. One solution is to replace each step-like discontinuity in the waveform with a bandlimited step function (BLEP) [3, 7, 8]. This study proposes a similar approach, in which the discontinuities introduced in the derivatives of the signal are replaced with bandlimited versions of themselves.

This proposed solution consists of a two-stage system in which the input signal is pre-processed by a hard clipper before entering the soft-clipping stage. Between the two stages, aliasing generated by the hard clipper is attenuated using a polynomial correction function modeled after the ideal bandlimited ramp function (BLAMP). This BLAMP function was originally proposed as a way to correct aliasing in triangular oscillators [8, 9]. The BLAMP method is proved to reduce the level of aliasing distortion seen at the output of clipping algorithms.

This paper is organized as follows. Section 2 deals with the distinction between hard and soft clipping. Section 3 presents the derivation of the BLAMP correction function and proposes a polynomial approximation. Section 4 evaluates the performance of the proposed algorithm. Finally, some concluding remarks and perspectives appear in Section 5.

[*]The work of Fabián Esqueda is supported by the CIMO Centre for International Mobility and the Aalto ELEC Doctoral School.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

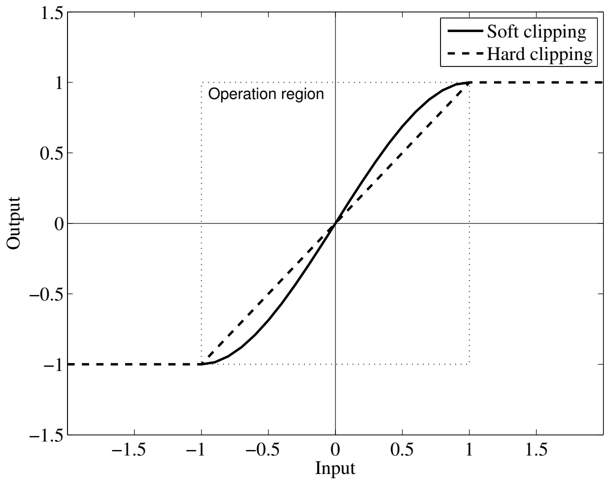

Fig. 1. Input–output relationship for the hard clipper and the soft clipper designed by Araya and Suyama [11].

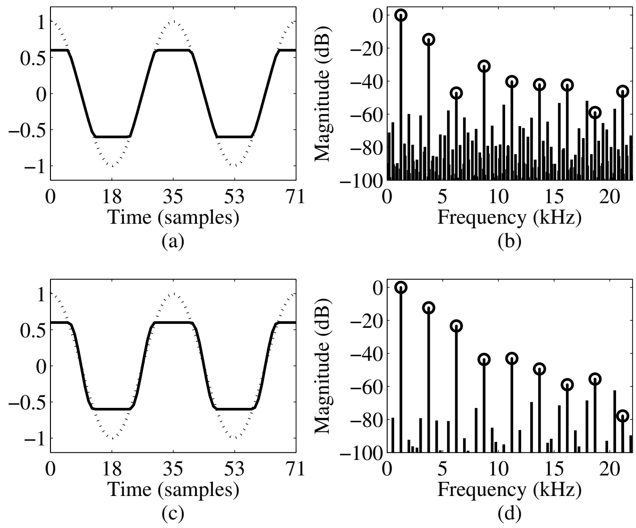

Fig. 2. Waveforms for a 1245 Hz sinewave with $L = 0.6$ (a) hard clipping, (c) soft clipping and their respective magnitude spectra (b) and (d). The circles indicate non-aliased components, which correspond to harmonic distortion.

2. ALIASING IN CLIPPING ALGORITHMS

Signal clipping can be divided into two types: hard and soft. In hard clipping, signal values that exceed a predetermined threshold are set to a maximum value (positive or negative), creating a sharp edge followed by a flat region. This sharp edge generates a discontinuity in the first derivative of the signal. In soft clipping, the transition between non-clipped to clipped samples is made gradual rather than abrupt, usually by a low-order polynomial transition region. As a result, the signal will now be continuous in its first derivative and aliasing will be significantly lower than in the hard clipping case. In general, if all derivatives up to the k^{th} derivative are continuous, the frequency spectrum will decrease at approximately $6(k + 1)$ dB per octave [10].

Digital distortion is usually implemented using soft clipping algorithms. In addition to the improved frequency response in terms of aliasing, soft clipping is usually perceived as having a "smoother" sound than the hard clipper, which is usually described as being "harsh" and "tinny". Fig. 1 compares the input–output relationships of a hard clipper and a third-order soft clipping function. The static soft clipping function $c_s(x)$ used in this example was designed as part of a multi-effects processor by Yamaha [11]. It is defined as

$$c_s(x) = \frac{3x}{2}\left(1 - \frac{x^2}{3}\right), \tag{1}$$

where x is the input signal and is defined between $[-1, 1]$. This cubic algorithm exhibits a fairly linear behavior for low input levels and a smooth curve as it approaches its maximum operation values of -1 and 1. The function for hard clipping, $c_h(x)$, can be defined as

$$c_h(x) = \begin{cases} x & \text{if } |x| < 1 \\ \text{sgn}(x) & \text{otherwise.} \end{cases} \tag{2}$$

For the sake of simplicity, we assume normalized input signals (i.e. bounded between $[-1, 1]$). Therefore, in order to implement any arbitrary clipping level, L, the signal must first be scaled by a factor $1/L$ before being processed by either one of the aforementioned clipping functions. The resulting waveform must be then scaled by L to return to the original range. Fig. 2 shows the waveform and frequency spectrum of a 1245 Hz sinewave clipped using (1) and (2). A sampling frequency of 44.1 kHz (standard for audio) was used for this and the rest of the examples in this paper. As expected, the level of aliasing distortion seen at the output of the hard clipper (see Fig. 2b) is much more dramatic than its soft counterpart (see Fig. 2d). Nevertheless, the soft-clipped signal still shows clear aliasing issues throughout its spectrum.

3. PROPOSED ALIASING REDUCTION METHOD

The main challenge when designing a correction function for soft-clipped signals is finding a solution that fits all problems. In the previous section, we used a third-order function to perform soft distortion; however, soft clippers come in different shapes and sizes. There is no standardized form to implement them. Fortunately, this is not the case for the hard clipper, which can only be implemented as defined by (2). Since the first derivative of a hard-clipped signal will always be discontinuous, we can design a general correction function based on this information. Therefore, if we wish to apply soft clipping to a signal we can pre-process it using a hard clipper and its correction function. This approach, while unconventional, yields output signals with significantly less aliasing distortion. Fig. 3 shows the block diagram for the proposed sys-

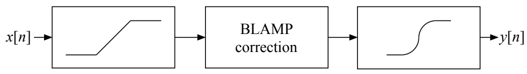

Fig. 3. In the proposed system, the input signal is first hard-clipped and corrected using the polyBLAMP method before entering the soft-clipping stage of the system.

tem. The following subsections detail the derivation of the BLAMP correction stage of the system.

3.1. Ideal Bandlimited Ramp Function

In hard-clipped signals, discontinuities in the first derivative are caused by sharp edges or corners introduced on the waveform. In order to reduce aliasing we need to replace all corners with bandlimited versions of themselves. For simplicity, we first consider the case of a single hard-clipped sinewave with fundamental frequency f_0. The amount of aliasing generated by the hard clipper will depend on two parameters: the amount of clipping (L) and f_0. At the same time, these two parameters will also determine the slope of the unclipped signal at exactly the point in time when it clips. Therefore, we can visualize the region around the clipping as a function that consists of a flat region, a hard edge or corner, and a slope. This function would resemble a ramp that starts rising when time equals zero. This ramp is defined as

$$f(t) = \begin{cases} 0 & \text{when } t < 0 \\ t & \text{when } t \geq 0, \end{cases} \tag{3}$$

where t represent time. For now, we will only consider the particular case when the ramp has a unit slope. The first and second derivatives with respect to time of this function are the unit step and the unit impulse, respectively. The analytical expression for the bandlimited unit impulse (shown in Fig. 4a) is the impulse response of the ideal brick-wall lowpass filter with cutoff at the Nyquist limit [12], defined as

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

where f_s is the sampling rate and $\text{sinc}(x) = \sin(\pi x)/\pi x$. Integrating this result yields the ideal BLEP function (Fig. 4b), defined as

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

where $\text{Si}(x)$ is the sine integral [8]. Then, we use integration by parts to derive the analytical expression for the ideal BLAMP with unity slope (Fig. 4c), which is given by

$$h^{(2)}(t) = th^{(1)}(t) + \frac{\cos(\pi f_s t)}{\pi^2 f_s}. \tag{6}$$

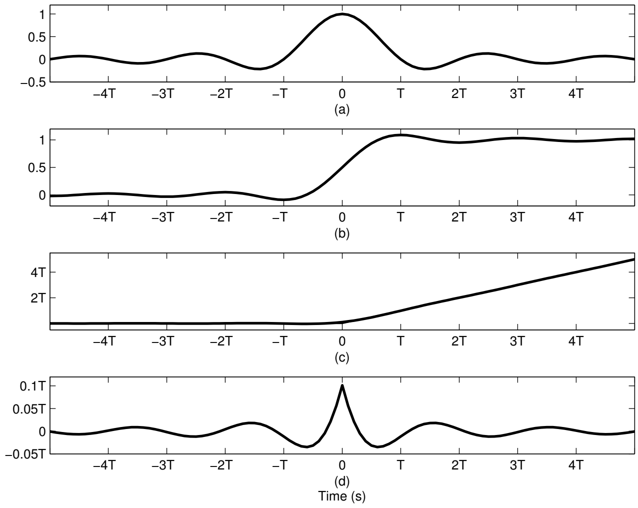

Fig. 4. Waveform for the continuous (a) bandlimited unit impulse, (b) BLEP, (c) BLAMP and (d) BLAMP residual. T is the sampling interval $1/f_\text{s}$.

As a final step, we define a residual function by computing the difference between the BLAMP and a trivial ramp. Fig. 4d shows the waveform for this correction function, which exhibits even symmetry. This function can then be used to reduce aliasing distortion by centering it around every clipping point, scaling it by m (the slope of the unclipped signal at that point), and adding the corresponding sampled values. Due to the definition we made for the trivial ramp in equation (3), the polarity of the BLAMP residual function must be inverted when correcting positive portions of the clipped signal. The amount of aliasing reduction will then depend on the number of samples corrected before and after each clipping point.

3.2. First-order BLAMP Approximation

The trivial implementation of the BLAMP method is considerably inefficient due to the presence of the sine integral. One possible improvement would be to precompute the residual function and use as a lookup table [7]. However, the efficiency and accuracy of this approach would then depend on the resolution of the table and interpolation method used, if any. Following the work of Välimäki et al. [8], we propose a first-order polynomial approximation of the BLAMP function, or polyBLAMP. This approximation will allow us to correct two samples, one before and one after each clipping point, with minimal computational costs.

First, we assume any given clipping point will occur between samples at $D + d$, where D is an integer value and d is the fractional delay ranging between [0, 1) [13]. Next, we approximate the bandlimited unit impulse (4) with the basis function for linear interpolation, the triangular pulse, and repeat the process detailed in the previous section (i.e. integrate twice and subtract the trivial ramp). Table 1 shows the

Span	Basis function: triangular pulse
$[-T, 0]$	d
$[0, T]$	$-d + 1$
Span	First integral: polyBLEP
$[-T, 0]$	$d^2/2$
$[0, T]$	$-d^2/2 + d + 1/2$
Span	Second integral: polyBLAMP
$[-T, 0]$	$d^3/6$
$[0, T]$	$-d^3/6 + d^2/2 + d/2 + 1/6$
Span	polyBLAMP residual
$[-T, 0]$	$d^3/6$
$[0, T]$	$-d^3/6 + d^2/2 - d/2 + 1/6$

Table 1. Linear interpolation polynomials, their first and second integrated forms, and polyBLAMP residual $(0 \leq d < 1)$.

intermediate steps required to derivate the polyBLAMP and its residual in terms of d. The basis polynomials were integrated twice, with their integration constants adjusted to produce continuous functions. Fig. 5 shows the waveforms that result from each one of these intermediate steps.

Overall, implementing this new method simply requires evaluating two third-order polynomials, one for each sample before and after the clipping point. As before, the resulting values must be scaled by the slope and inverted for positive portions of the signal.

4. EVALUATION OF THE PROPOSED METHOD

Up until now, we have only considered the ideal case in which the exact fractional clipping points and their respective slopes are known. Carrying on with this assumption, we can analyze the case of a single sinusoidal input signal. Figs. 6a and 6b show the waveform and frequency spectrum for a 1245 Hz sinewave hard-clipped and corrected using the polyBLAMP. The level of aliasing distortion has been dramatically reduced in comparison with the trivial implementation (see Figs. 2a and 2b). The corrected signal was subsequently processed by the soft clipper. Its waveform and magnitude spectrum are shown in Figs. 6c and 6d, respectively. As expected, aliasing at the output of the soft clipping stage has been considerably attenuated with minimal corruption of the non-aliased components (compare with Figs. 2c and 2d).

Now, in order to be able to implement the polyBLAMP method with arbitrary audio signals (e.g. a guitar track), we must find a way to estimate the fractional clipping points and their respective slopes based on the data available. The most straightforward approach is to first find the clipping boundaries of the clipped signal, i.e. the samples before and after the clipping point. Going back to the original unclipped signal, we estimate the slope at the clipping point by computing the difference between the two clipping boundaries. This approximated slope can then be used to draw a straight line between the boundaries and to estimate the fractional clipping point using inverse linear interpolation.

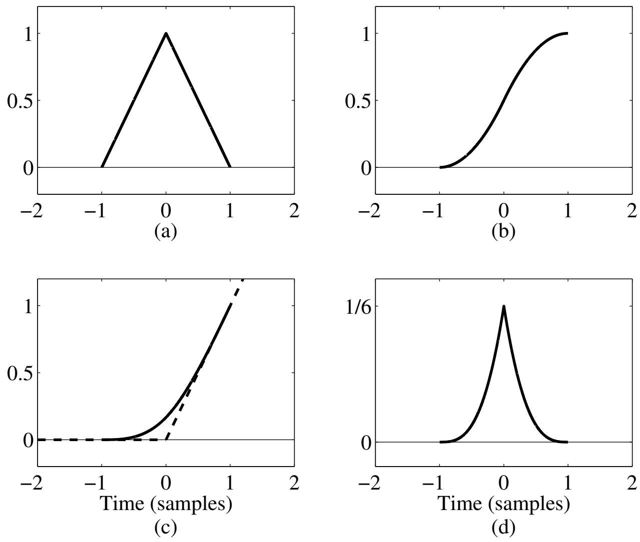

Fig. 5. (a) Linear interpolation basis function: triangular pulse, (b) its first integral, (c) second integral (solid line) and trivial ramp (dashed line), and (d) polyBLAMP residual.

Fig. 7 shows the spectrograms for two 5-second linear chirps going from 20 Hz to 5 kHz, soft-clipped trivially and with the polyBLAMP method, respectively. For the polyBLAMP implementation, fractional clipping points and their respective slopes were estimated following the aforemen-

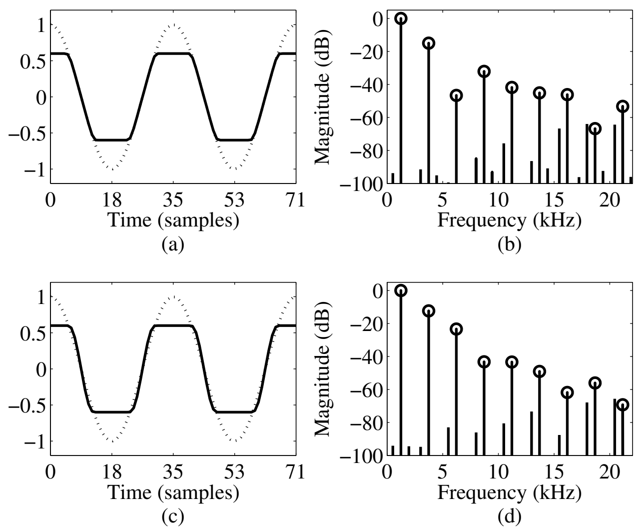

Fig. 6. Waveforms for a 1245 Hz sinewave with $L = 0.6$ (a) hard clipping with polyBLAMP correction, (c) soft clipping after polyBLAMP correction, and their respective magnitude spectra (b) and (d). Circles indicate non-aliased components.

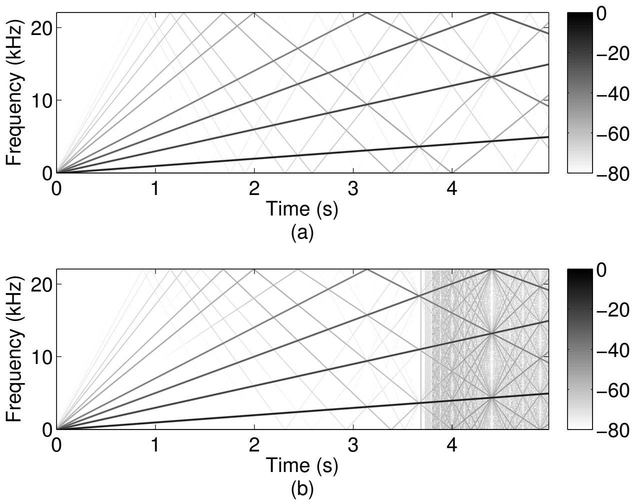

Fig. 7. Spectrograms for a linear sine sweep with $L = 0.6$ (a) trivial soft clipping and (b) soft clipping with polyBLAMP aliasing reduction, respectively.

tioned approach. This example demonstrates that the level of aliasing distortion is effectively attenuated for fundamental frequencies up to approx. 3 kHz. After this threshold, the suggested polyBLAMP implementation actually introduces severe noise artifacts and should not be used. Nevertheless, this performance can be exclusively attributed to the poor estimation of clipping points and slope values.

5. CONCLUSIONS AND FURTHER WORK

This paper discussed the issue of aliasing in clipping algorithms commonly used for musical distortion. A correction technique was introduced based on the ideal bandlimited ramp function. The correction function was derived along with a first-order polynomial approximation aimed at reducing computational costs. It was shown that the proposed method can effectively reduce the level of aliasing seen at the output of soft clipping algorithms, provided the fractional clipping points of the input signal and their respective slopes are available. For the case of arbitrary audio signals, inverse linear interpolation was suggested as a way to estimate these parameters. However, this approach fails at high fundamental frequencies, where noise artifacts are introduced.

Due to the importance of accurate edge and slope detection, the next step in this research will be to devise more robust techniques to estimate these parameters. Additionally, higher-order polyBLAMP coefficients can be derived in order to correct more samples before and after each clipping point. Either Lagrangian or B-Spline interpolators could be used for this purpose.

REFERENCES

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

[2] V. Välimäki, F. Fontana, J.O. Smith, and U. Zölzer, "Introduction to the special issue on virtual analog audio effects and musical instruments," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 713–714, May 2010.

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

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

[5] P. Dutilleux, K. Dempwolf, M. Holters, and U. Zölzer, "Nonlinear processing," in *DAFX: Digital Audio Effects*, Udo Zölzer, Ed., pp. 102–138. Wiley, West Sussex, United Kingdom, 2nd edition, 2011.

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

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

[8] 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.

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

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

[11] T. Araya and A. Suyama, "Sound effector capable of imparting plural sound effects like distortion and other effects," US Patent 5,570,424, 29 Oct. 1996.

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

[13] 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.