REDUCING THE ALIASING OF NONLINEAR WAVESHAPING USING CONTINUOUS-TIME CONVOLUTION

Julian D. Parker, Vadim Zavalishin, Efflam Le Bivic

Native Instruments GmbH,
Berlin, Germany
julian.parker@native-instruments.de

ABSTRACT

Nonlinear waveshaping is a common technique in musical signal processing, both in a static memoryless context and within feedback systems. Such waveshaping is usually applied directly to a sampled signal, generating harmonics that exceed the Nyquist frequency and cause aliasing distortion. This problem is traditionally tackled by oversampling the system. In this paper, we present a novel method for reducing this aliasing by constructing a continuous-time approximation of the discrete-time signal, applying the nonlinearity to it, and filtering in continuous-time using analytically applied convolution. The presented technique markedly reduces aliasing distortion, especially in combination with low order oversampling. The approach is also extended to allow it to be used within a feedback system.

1. INTRODUCTION

Nonlinear waveshaping has been part of the toolbox of musical signal processing since the 1960s, when overdrive and fuzz pedals became popular for treating the sound of the electric guitar. Its application in the digital signal processing domain began in the 1970s with exploration of synthesis methods employing waveshaping using Chebyshev polynomials [1, 2]. In more recent times, investigation of waveshaping has mainly been pursued within the domain of virtual analog modelling. A particularly active area of research has been filters with embedded nonlinearities, including the Moog ladder filter [3–7], the diode ladder filter [5, 8] and Sallen-Key based filters [9, 10]. Much work has also been done on the digital emulation of analog overdrive and fuzz pedals [11, 12] and tube amplifiers [13, 14]. Other nonlinear analog devices have been modelled, including the ring-modulator [15, 16] and bucket-brigade based effects [17, 18]. Recent work has applied correction functions usually used for oscillator antialiasing to the problem of antialiasing signals processed by a hard clipper [19], and has also considered more abstract applications of nonlinear waveshaping [20, 21].

One of the primary problems encountered when dealing with nonlinearities is that of aliasing distortion. Aliasing distortion is present in many types of digital signal processing algorithm, and is generally perceived to be disturbing or unpleasant [22]. In this paper, we describe a method for reducing the aliasing produced by processing a signal with a memoryless nonlinearity, as well as describe how the same method can be applied inside a filter. This method is based on forming a continuous-time approximation of the signal, applying the nonlinearity, and analytically deriving the result of applying convolution with a continuous-time lowpass filter kernel. The application of the filtering process in the continuous domain is crucial to working of the method, as it allows suppression of components even beyond the original Nyquist frequency of the system. The method is related to the Differentiated Polynomial Waveform (DPW) approach to antialiasing oscillator waveforms [23–25], which also applies filtering in the continuous-time domain. There is also some relation to methods of antialiasing wavetable playback using integrated wavetables [26, 27].

In Sec. 2, we describe the simplest formulation of the technique. Sec. 3 describes how the method can be extended to use any piecewise polynomial filter kernel, with the example of the triangular or linear interpolation kernel given. In Sec. 4, we discuss some drawbacks of the method—specifically its delay, and filtering effects below Nyquist. Examples of applying the method to a number of nonlinearities are given in Sec. 5. In Sec. 6, we broaden the method to apply to feedback systems, where the delay of the antialiased nonlinearity can be directly compensated by removal of equivalent parts of the filter structure. Finally, in Sec. 7 we draw conclusions about the presented work.

2. APPROXIMATION OF CONTINUOUS DOMAIN NONLINEAR WAVESHAPING

Nonlinear waveshaping in a digital signal processing context is generally applied directly to a discrete-time signal:

$$y[n] = f(x[n]) \tag{1}$$

where y denotes the output, x the input, n the discrete-time sample index, and f is an arbitrary nonlinear function. The waveshaping process is not bandlimited, and therefore depending on the input can generate frequency components exceeding the Nyquist frequency of the system (in many cases, even an infinite series of components). The components are reflected around the Nyquist frequency, and appear within the output spectrum as *aliasing* distortion. This is a well-known and common problem in musical digital signal processing. The most prominent method of reducing aliasing is to oversample the processing of the signal through the nonlinearity. This raises the Nyquist frequency, and hence the point at which the generated harmonics alias. However, this approach is still far from ideal given that the sequence of harmonics can be infinite.

The ideal result of the process, that of applying the nonlinearity to the input signal without any generated aliasing, can be thought of as perfect sampling of the same nonlinearity applied in continuous-time:

$$y(t) = f(\tilde{x}(t)) \tag{2}$$

where $\tilde{x}$ is a continuous-time reconstruction of our input signal, y is the continuous-time output signal, and t is the time variable. Given that we are working within a purely discrete-time context and presumably don't want to pass our signal out to the continuous domain for nonlinear processing, the challenge is to approximate this expression as accurately as possible whilst staying within the discrete-time domain.

2.1. Approximating a continuous-time input signal

The first challenge is to devise a version of the discrete-time input signal which can be treated in the same way as its continuous-time reconstruction. In the case of a known input signal, for example a sinusoid, this can be done trivially. In the case of an arbitrary signal, the problem is more complicated—we need to draw a function through the known sample-points of the input signal. One way of approaching this problem is to utilise a standard interpolation technique to 'fill in the gaps' between sample points—resulting in a piecewise approximation of the ideal continuous-time input signal. In the case of linear interpolation, and assuming a unit sampling interval, this would result in the following:

$$\tilde{x}(t) = \begin{cases} x_1 + \tau(x_0 - x_1), & 0 \leq t < 1 \\ x_2 + \tau(x_1 - x_2), & 1 \leq t < 2 \\ \vdots & \\ x_n + \tau(x_{n-1} - x_n), & (n-1) \leq t < n \end{cases} \tag{3}$$

where $x_n \equiv x[n]$ is shorter notation for the samples of the discrete-time input signal, and $\tau = 1 - (t \mod 1)$, a time variable that runs $1 \ldots 0$ between each sample.

Higher-order interpolation methods could be applied here, with the result of better suppression of the image spectra that repeat infinitely above the original Nyquist frequency. However, approximating the input signal as locally linear allows an analytic solution of the method to be derived, as will be seen in the following section.

2.2. Returning to the discrete-time domain

Now, given an expression for $\tilde{x}(t)$, we are able to calculate $y(t)$ at any specific t via (2). This can be considered to be sampling the approximated continuous-time signal at an arbitrary point. However, if we want our discrete-time output signal to be free of aliasing, we need to apply some kind of filtering to the continuous-time signal before it is sampled to remove components above our original Nyquist frequency. This can be done by applying a continuous-time convolution with some filter kernel h to $y(t)$:

$$\tilde{y}(t) = \int_{-\infty}^{\infty} h(u) y(t-u) du \tag{4}$$

where $\tilde{y}(t)$ is the approximately bandlimited continuous-time output. $\tilde{y}(t)$ can then be trivially sampled at the original sampling times (again assuming unit sample interval):

$$y[n] = \tilde{y}(n) \tag{5}$$

A very simple example of a lowpass kernel is a rectangular function of unit width:

$$h_{\text{rect}}(t) = \begin{cases} 1, & 0 \leq t \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Fig. 1 shows the amplitude response of this kernel, along with its time-domain form. From the amplitude response, we can see that the convolution will significantly attenuate any harmonics which exceed the original Nyquist frequency.

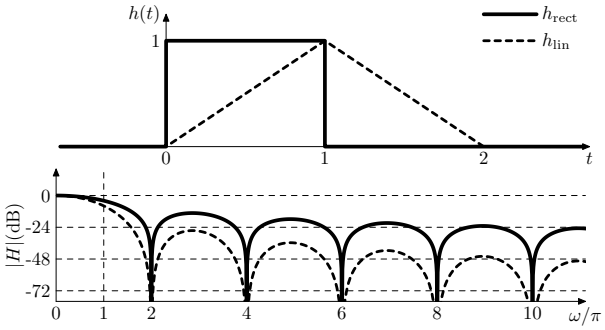

Figure 1: *Rectangular and linear continuous-time convolution kernels and their continuous time amplitude responses.*

We can now write an expression for the output $y[n]$:

$$\begin{aligned} y[n] = \tilde{y}(n) &= \int_{-\infty}^{\infty} h_{\text{rect}}(u) y(n-u) du \\ &= \int_0^1 y(n-u) du \\ &= \int_0^1 f(\tilde{x}(n-u)) du \end{aligned}$$

using (3), and noticing that over this interval $u = \tau$, we can write:

$$\begin{aligned} y[n] &= \int_0^1 f(\tilde{x}) d\tau \\ &= \int_0^1 f(x_n + \tau(x_{n-1} - x_n)) d\tau \end{aligned} \tag{7}$$

From integration by substitution, we can write:

$$\int_0^1 f(\tilde{x}) \frac{d\tilde{x}}{d\tau} d\tau = \int_{x_n}^{x_{n-1}} f(\tilde{x}) d\tilde{x}$$

The piecewise linear nature of $\tilde{x}$ now becomes useful as it means that $\frac{d\tilde{x}}{d\tau}$ is constant over the extent of the τ integration, and can be factored out of the integral to produce:

$$\begin{aligned} y[n] = \int_0^1 f(\tilde{x}) d\tau &= \frac{d\tau}{d\tilde{x}} \int_{x_n}^{x_{n-1}} f(\tilde{x}) d\tilde{x} \\ &= \frac{1}{x_{n-1} - x_n} \int_{x_n}^{x_{n-1}} f(\tilde{x}) d\tilde{x} \end{aligned} \tag{8}$$

Finally, by applying the fundamental theorem of calculus, we produce:

$$y[n] = \frac{F_0(x_n) - F_0(x_{n-1})}{x_n - x_{n-1}} \tag{9}$$

where F_0 is the antiderivative of f.

2.3. Precision and ill-conditioning concerns

In a digital context with finite precision arithmetic, (9) becomes ill-conditioned when $x_n \approx x_{n-1}$ due to a $0/0$-type uncertainty, resulting in precision loss or even a division by zero. Assuming floating point numeric representation the precision loss occurs from the subtraction of two values of the same sign and comparable magnitude. This kind of precision loss in the numerator can be

minimized by choosing the integration constant for F_0 such that $F_0(0) = 0$. In this case the precision loss in the entire formula is determined by that in the denominator.

According to the derivation given in Appendix A:

$$\frac{F_0(x_n) - F_0(x_{n-1})}{x_n - x_{n-1}} = f\left(\frac{x_n + x_{n-1}}{2}\right) + O((x_n - x_{n-1})^2) \tag{10}$$

where $O(\dots)$ denotes the order of the neglected terms, and hence the order of the error. This value can then be substituted for (9) when the value of $x_n - x_{n-1}$ becomes very small.

3. EXTENSION TO HIGHER-ORDER FILTER KERNELS

The filter kernel described in (6) is the simplest possible kernel that can be used in this technique. A more complex kernel can be used, as long as the convolution can be performed analytically. The triangular or linear-interpolation kernel is an example of such a kernel:

$$h_{\text{lin}}(t) = \begin{cases} t, & 0 \le t < 1 \\ 2 - t, & 1 \le t \le 2 \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

Substituting this into (4), and following the same procedure as previously, we obtain:

$$\begin{aligned} \tilde{y}(n) &= \int_{-\infty}^{\infty} h_{\text{lin}}(u) y(n-u) du \\ &= \int_0^1 u y(n-u) du + \int_1^2 (2-u) y(n-u) du \\ &= \int_0^1 \tau f(x_n + \tau(x_{n-1} - x_n)) d\tau + \\ &\quad \int_0^1 (1-\tau) f(x_{n-1} + \tau(x_{n-2} - x_{n-1})) d\tau \end{aligned} \tag{12}$$

The two integrals can be evaluated by noting that over the interval we're considering $\tau = \frac{\tilde{x} - x_n}{x_{n-1} - x_n}$. Therefore, by again applying integration by substitution, we have:

$$\begin{aligned} \tilde{y}(n) = &\int_{x_n}^{x_{n-1}} \frac{\tilde{x} - x_n}{(x_n - x_{n-1})^2} f(\tilde{x}) d\tilde{x} + \\ &\int_{x_{n-1}}^{x_{n-2}} \frac{x_{n-2} - \tilde{x}}{(x_{n-1} - x_{n-2})^2} f(\tilde{x}) d\tilde{x} \end{aligned} \tag{13}$$

As before, we can use the fundamental theorem of calculus to write this expression in terms of antiderivatives of f. Additionally, the antiderivative of $xf(x)$ is needed. We denote this as $F_1(x)$. After some simplification, the expression for the output can be written as:

$$\begin{aligned} \tilde{y}(n) = &\frac{x_n(F_0(x_n) - F_0(x_{n-1})) - (F_1(x_n) - F_1(x_{n-1}))}{(x_n - x_{n-1})^2} + \\ &\frac{x_{n-2}(F_0(x_{n-2}) - F_0(x_{n-1})) - (F_1(x_{n-2}) - F_1(x_{n-1}))}{(x_{n-2} - x_{n-1})^2} \end{aligned} \tag{14}$$

The frequency response of the triangular kernel can be seen in Fig. 1. As expected, the suppression of harmonics above Nyquist will be improved compared to the rectangular kernel. However, the expression needed to compute the anti-aliased output is more complex than in the rectangular kernel case. This is ameliorated somewhat by the fact that many terms can be calculated by storing and re-using the values calculated at previous time steps. Consequently, there should only be one evaluation of F_0 and of F_1 per sample.

The same derivation can be followed for any higher order kernel which consists of piecewise polynomial sections, for example a Hermite interpolator. However as the polynomial order of the kernel grows, the higher-order counterparts of the analytical convolution expressions (9) and (14) become more ill-conditioned. Therefore the precision requirements of the computation grow and so potentially the computational load.

3.1. Precision and ill-conditioning concerns

As in the rectangular window case, when $x_n \approx x_{n-1}$ or $x_{n-1} \approx x_{n-2}$, (14) becomes ill-conditioned. Again, this can be resolved as described in Appendix A. In this case, we must deal with the first and second terms of (14) separately. The first term becomes:

$$\frac{1}{2} f\left(\frac{x_n + 2x_{n-1}}{3}\right) \tag{15}$$

when $x_n \approx x_{n-1}$. Similarly, the second term becomes:

$$\frac{1}{2} f\left(\frac{x_{n-2} + 2x_{n-1}}{3}\right) \tag{16}$$

when $x_{n-1} \approx x_{n-2}$.

4. LINEAR RESPONSE AND GROUP DELAY OF METHOD

The conversion to the continuous-time domain and back can be seen as oversampling to an infinitely large sampling rate. The linear interpolation (3) and convolution (4) in that regard can be seen as lowpass filters used in resampling.

Additionally, many nonlinearities of interest become transparent at very low signal levels: $f(x) \approx x$. In this case the whole system becomes a combination of an upsampler and a downsampler. Therefore the system behaves as an LTI filter, where the overall filtering effect is a combination of the lowpass filters (3) and (4) and the aliasing occuring due to both filters being non-brickwall.

Using (41) of Appendix A we obtain the linear-case version of (9):

$$y[n] = \frac{x_n + x_{n-1}}{2} \tag{17}$$

Therefore at low signal levels (9) can be viewed as a half-sample fractional delay using linear interpolation (i.e. its group delay is 0.5 samples).

In a similar fashion for (14) we obtain

$$y[n] = \frac{x_n + x_{n-2}}{6} + \frac{2}{3} x_{n-1} \tag{18}$$

corresponding to a group-delay of 1 sample.

The respective amplitude responses of (17) and (18) are plotted in Fig. 2. Higher-order kernels can be treated in a similar fashion. Extension to the case with non-unity linear scaling at low signal levels is also trivial.

5. EXAMPLES AND RESULTS

In the following section, we describe how the methods described above can be applied to a number of nonlinearities.

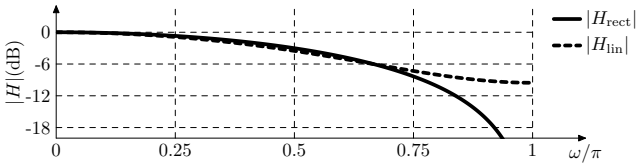

Figure 2: *Low signal-level discrete-time frequency responses of the method when using rectangular and linear continuous-time convolution kernels.*

5.1. Example: $\tanh()$

One of the most common nonlinear functions used in music signal processing is the hyperbolic tangent:

$$f(x) = \tanh(x) \tag{19}$$

the antiderivative is given by:

$$F_0(x) = \log(\cosh(x)) \tag{20}$$

therefore:

$$y[n] = \frac{\log(\cosh(x_n)) - \log(\cosh(x_{n-1}))}{x_n - x_{n-1}} \tag{21}$$

in the case where the rectangular kernel is used.

The antiderivative of $x\tanh(x)$ is given by:

$$F_1(x) = \frac{1}{2}\left(x\left(x + 2\log\left(e^{-2x}+1\right)\right) - \text{Li}_2\left(-e^{-2x}\right)\right) \tag{22}$$

where Li_2 is the dilogarithm function. The full expression for $y[n]$ can be obtained by substituting the expressions for $F_0(x)$ and $F_1(x)$ into (14).

The expression (22) is rather expensive to compute. Therefore it may be beneficial to tabulate it. Due to the ill-conditioned nature of (14) the tabulation needs to be done with high precision. Particularly, usage of piecewise segments of higher than linear order may be advised for the table. Also, warping of the argument scale (for example, making it logarithmic) can further reduce the table size. The unbounded argument range for the tabulated function can be achieved by noting that $\tanh(x) \approx \text{sgn}(x)$ within very high precision for $|x| \gg 1$, thus $F_1(x)$ can be approximated by

$$F_1(x) \approx F_1(x_0\,\text{sgn}(x)) + \frac{x^2 - x_0^2}{2}\,\text{sgn}(x),\ |x| \geq x_0 \gg 1 \tag{23}$$

for some sufficiently large x_0. The tabulation approach also allows us to deal with nonlinearities which cannot be analytically integrated.

Fig. 3 shows spectrograms of a linear sine sweep, processed by a $\tanh()$ nonlinearity with a linear input gain of 5, at two different sample rates, without and with the continuous-time convolution applied. At both 44.1kHz and 88.2kHz, there is a significant reduction in aliasing when the method is applied. At 44.1kHz a small loss of high-frequency content is visible, due to the effective discrete-time frequency response of the method. This effect is explained in Sec. 4. As can be seen, the triangular kernel produces noticeably greater suppression of aliased components than the rectangular kernel. However, the difference is not as large as that between no antialiasing and the rectangular kernel.

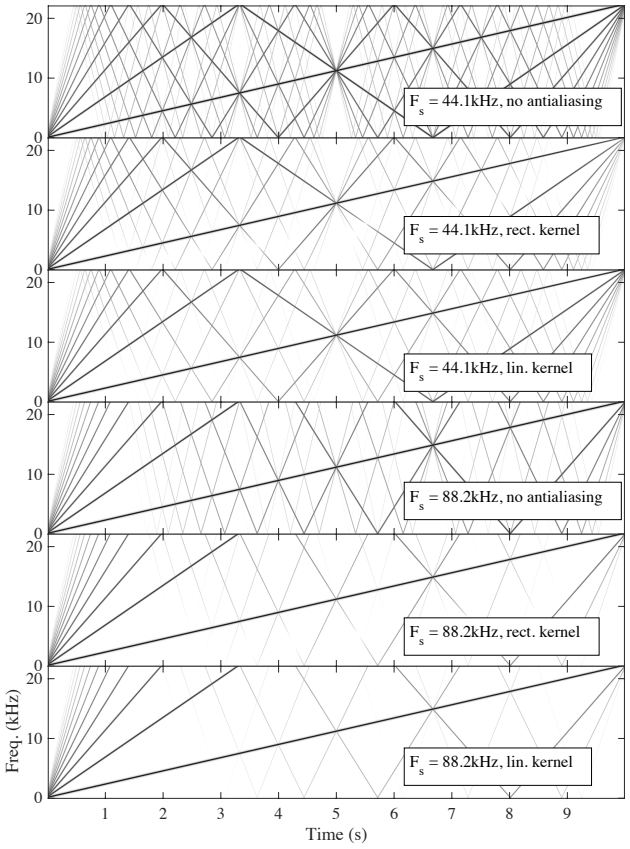

Figure 3: *Spectrogram of linear sine sweep from 0…22kHz, processed with* $\tanh()$ *function with an input gain of 5. The minimum amplitude visible is -80dBFS.*

5.2. Example: Hard Clipper

Another common saturating nonlinearity in music signal processing is the hard clipper, defined by:

$$f(x) = \begin{cases} x, & -1 \leq x \leq 1 \\ \text{sgn}(x), & \text{otherwise} \end{cases} \tag{24}$$

Recent work has considered alias-suppression for this function by applying a correction function to the transition between linear and clipped regions [19].

The antiderivative of (24) is trivially calculated, taking care to set the arbitrary constant of integration so that the function passes through the origin:

$$F_0(x) = \begin{cases} \frac{1}{2}x^2, & -1 \leq x \leq 1 \\ x\,\text{sgn}(x) - \frac{1}{2}, & \text{otherwise} \end{cases} \tag{25}$$

Similarly, the antiderivative of $xf(x)$ can be calculated:

$$F_1(x) = \begin{cases} \frac{1}{3}x^3, & -1 \leq x \leq 1 \\ (\frac{1}{2}x^2 - \frac{1}{6})\,\text{sgn}(x), & \text{otherwise} \end{cases} \tag{26}$$

In order to illustrate the effectiveness of the technique, a test was performed. A linear sine sweep wave processed by the nonlinearity with a very large input gain, in this case 10. The sample

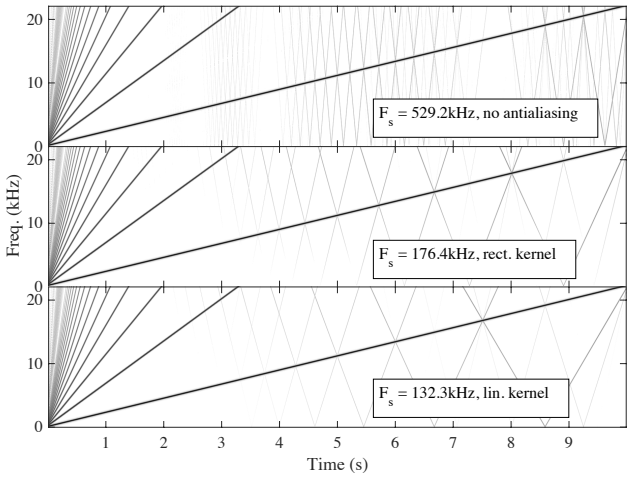

Figure 4: *Spectrogram of linear sine sweep from 0... 22kHz, processed with hard clipping function with an input gain of 10. The minimum amplitude visible is -80dBFS.*

rate of the cases was adjusted until roughly the same amount of aliasing was present. Signal-to-noise ratio (SNR) was calculated by creating an image mask denoting the non-aliased part of the spectrogram and calculating the ratio of the power inside the image mask to that outside the image mask. The mask is created from the spectrogram of an 'ideal' version of the processed signal (in this case calculated at $F_s = 11.2$MHz) by picking bins which have a power greater than –30dB. The results are given in Table 1 and shown in Fig. 4.

Table 1: *Oversampling required for similar aliasing level of hard clipper with input gain of 10.*

Kernel	F_s(kHz)	$F_s/44.1$kHz	SNR(dB)
None	529.2	12	46.7
Rectangular	176.4	4	46.3
Triangular	132.3	3	46.6

6. APPLICATION TO SYSTEMS WITH FEEDBACK

As discussed above in Sec. 4, the antialiasing method introduces delay into the signal path, which poses a problem in systems with feedback. This becomes especially critical in the systems with delayless feedback, which occurs in implicit time-discretization schemes, most prominently in trapezoidal integration. In the following we propose an approach to address this problem.

6.1. Delay elimination

Equation (17) is equivalent to the equation of the FIR part of a DF1 (Direct Form 1) trapezoidal integrator, shown in Fig. 5. Consider a serial connection of an antialiased nonlinearity Φ, as described in (9), and a DF1 trapezoidal integrator. This configuration is shown in Fig. 6. Since, according to (17), the nonlinearity (9) is already implementing the FIR part of the integrator, we can drop this part of the integrator from the structure, thereby eliminating the extra delay. The resulting structure is shown in Fig. 7.

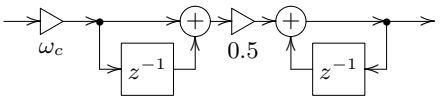

Figure 5: *Direct form 1 trapezoidal integrator (ω_c is the embedded cutoff control gain).*

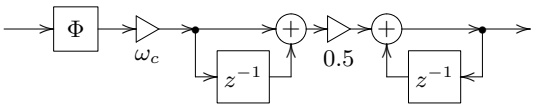

Figure 6: *A serial chain of an antialiased nonlinearity* (9) *(denoted by Φ) and a DF1 trapezoidal integrator.*

The nonlinearity and the integrator do not have to be located immediately next to each other in the feedback loop. However, any structural elements in the intermediate area must be considered. For example, consider a serial chain of a nonlinearity and a 1-pole lowpass filter, as shown in Fig. 8. In this case, as there is a summation node in between the nonlinearity and the integrator we must also introduce the same FIR part in the other path into the summation. This structure is shown in Fig. 9. It is interesting to note that the structure in Fig. 9 is equivalent to a naive 1-pole lowpass filter with an adjusted cutoff, as shown in Fig. 10.

Such manipulations change the topology of the system, and thus have the potential to change its time-varying behaviour. In practice, these changes are not usually severe enough to be a problem, but care must be taken.

The approach of Fig. 7 can be used as long as nonlinearities and integrators are interleaved in the feedback path. In the case where a nonlinearity does not have an associated integrator to absorb its delay, we can insert a very high cutoff (close to Nyquist) 1-pole lowpass filter immediately following (or preceding) it. This allows the delay to be absorbed, whilst hopefully having a minimal effect on the behaviour of the system. However, as the inserted lowpass has the potential to alter the linear (small signal) frequency response of the filter, care must again be taken.

6.2. Solving the antialiased implicit equation

Topologies containing delayless feedback require an implicit equation to be solved in order to be computable. If the implicit equation is transcendental, which is common with the types of saturating nonlinearity often used in musical filters, then linearisation is usually performed. A common approach to this problem is to apply the Newton–Raphson method. To this end, when the systems contains antialiased nonlinearities we need to be able to differentiate the antialiased nonlinearities with respect to their instantaneous in-

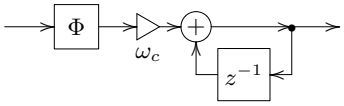

Figure 7: *A version of Fig. 6 with eliminated antialiasing delay.*

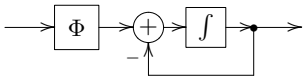

Figure 8: *A serial chain of an antialiased nonlinearity* (9) *and a 1-pole low pass filter.*

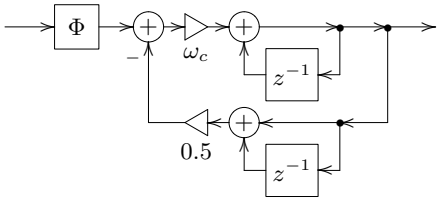

Figure 9: *Application of the delay elimination across a summation node in Fig. 8.*

put signal. Consider $y[n]$ in (9), expressed as a function of x_n:

$$y[n] = \Phi(x_n) = \frac{F_0(x_n) - F_0(x_{n-1})}{x_n - x_{n-1}} \tag{27}$$

Then

$$\frac{d\Phi}{dx_n} = \frac{f(x_n) \cdot (x_n - x_{n-1}) - \big(F_0(x_n) - F_0(x_{n-1})\big)}{(x_n - x_{n-1})^2} \tag{28}$$

Similarly to (9), the equation (28) becomes ill-conditioned when $x_n \approx x_{n-1}$. The ill-conditioned case substitute for (28) can be obtained by differentiating (10) with respect to x_n resulting in:

$$\frac{d\Phi}{dx_n} = \frac{1}{2} f'\left(\frac{x_n + x_{n-1}}{2}\right) + O(x_n - x_{n-1}) \tag{29}$$

6.3. Example: Moog Ladder Filter

Consider the Moog-ladder-like structure [4, 28] in Fig. 11, using trapezoidal integration. By folding the nonlinearity into the first of the four 1-pole lowpasses (as shown in Fig. 9) we eliminate the delay. Then we can apply Newton–Raphson to compute the solution of the implicit equation.

The performance of the approach was tested by driving the filter in Fig. 11 at $k = 8$ (strong selfoscillation) with a 5kHz unit amplitude sine oscillator, while sweeping the filter cutoff from 1Hz to 21kHz. We have plotted the spectrograms for non-antialiased and antialiased versions of filter, where we iterated Newton–Raphson until the convergence reached -60dB or better.

The output is shown in Fig. 12, at 44.1kHz and 88.2kHz sampling rates. The output at high sampling rate (576kHz) without antialiasing is provided as a reference. As can be easily seen, the antialiasing produces a noticeable reduction in aliasing. This has a particularly strong effect on the dynamics of the filter as the frequency is swept, as it prevents the resonant peak from locking on to particular frequencies where aliased components coincide. This change in dynamics is very clearly audible, with the non antialiased versions sounding 'steppy' as their cutoff is swept.

Additionally, the extra lowpass method was tested in the same filter topology. The extra lowpass filter's normalized prewarped cutoff was set to $\omega_c = 30$ (the corresponding unprewarped cutoff $\omega \approx 0.98\pi$). The results are also shown in Fig. 12, and appear

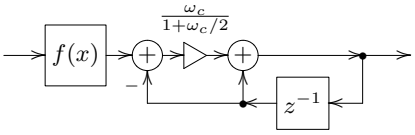

Figure 10: Structure of Fig. 9 with resolved local delayless feedback.

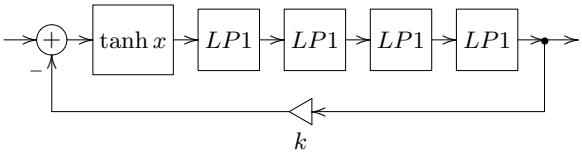

Figure 11: *A Moog-ladder-like filter.*

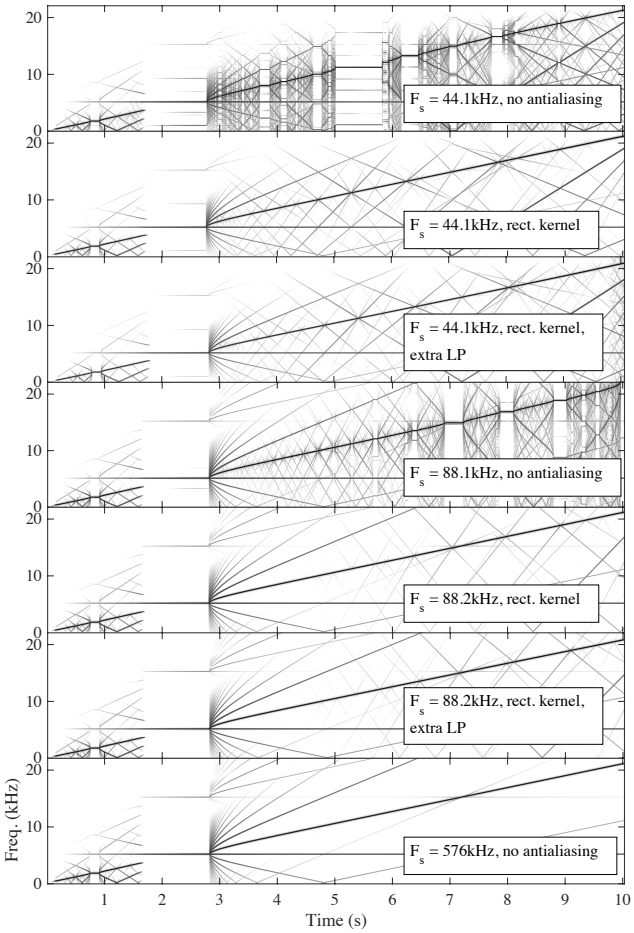

Figure 12: *Cutoff sweep of Moog-ladder-like filter with high resonance and 5kHz unit sinusoidal input. Shown without antialiasing, with antialiasing and integrator delay elimination, and with antialiasing and extra lowpass delay elimination.*

to be very similar to those given by the standard antialiasing tech-

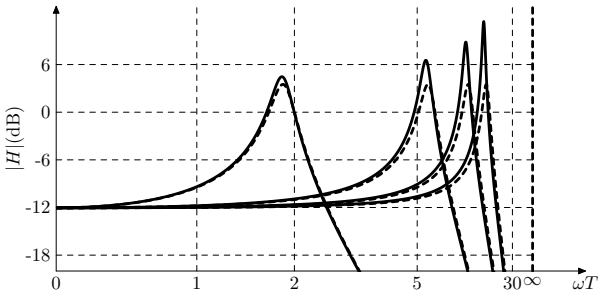

Figure 13: *Effect of the extra lowpass on the amplitude response of the Moog-ladder-like filter at $k = 3$. Dashed curves show the amplitude response in the absence of the extra lowpass. The ω axis is using the prewarped frequency scale (∞ is Nyquist, 2 is half Nyquist frequency, 30 is the extra lowpass's cutoff).*

nique. However, the inserted lowpass increases the filter's resonance and lowers the resonant frequency at high cutoff settings, as illustrated by Fig. 13. This effect is not strongly audible at sampling rates of 88.2kHz or above.

7. CONCLUSION

In this work, we have described a new method of suppressing aliasing when processing a digital signal with a nonlinear waveshaper. The method allows generated harmonics to be suppressed above Nyquist, by constructing a continuous-time approximation of the input signal, applying the waveshaping, and then analytically applying a convolution with a continuous-time filter kernel. The method is especially effective in situations where the nonlinearity generates a large amount of harmonics, such as a saturating nonlinearity with large input gain. In these situations, applying the antialiasing can give a similar improvement to a much larger level of oversampling. Techniques for applying the method within feedback systems were also given.

Sound and code examples illustrating the techniques described in this paper are available at the accompanying website[1].

8. ACKNOWLEDGEMENTS

After the initial submission of this work, the authors were made aware of a private unpublished work by Antti Huovilainen from 2010 that describes a subset of the technique detailed in this paper. The authors were not aware of this work during the writing process, but would like to acknowledge its existence.

[1] `http://github.com/julian-parker/DAFX-AntiAliasing`

9. REFERENCES

[1] M. Le Brun, "Digital waveshaping synthesis," *J. Audio Eng. Soc.*, vol. 27, no. 4, pp. 250–266, 1979.

[2] C. Roads, "A tutorial on non-linear distortion or waveshaping synthesis," *Computer Music Journal*, vol. 3, no. 2, pp. 29–34, 1979.

[3] A. Huovilainen, "Non-linear digital implementation of the Moog ladder filter," in *Proc. 7th Int. Conf. Digital Audio Effects*, Naples, Italy, Oct. 2004, pp. 61–64.

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

[5] V. Zavalishin, *The art of VA filter design*, Native Instruments, 2012.

[6] S. D'Angelo and V. Välimäki, "Generalized Moog ladder filter: Part I–linear analysis and parameterization," *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 22, no. 12, pp. 1825–1832, 2014.

[7] S. D'Angelo and V. Välimäki, "Generalized Moog ladder filter: Part II–explicit nonlinear model through a novel delay-free loop implementation method," *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 22, no. 12, pp. 1873–1883, 2014.

[8] F. Fontana and M. Civolani, "Modeling of the EMS VCS3 voltage-controlled filter as a nonlinear filter network," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 18, no. 4, pp. 760–772, May 2010.

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

[10] J. D. Parker and S. D'Angelo, "A digital model of the Buchla low-pass-gate," in *Proc. 16th Int. Conf. Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, 2013, pp. 278–285.

[11] D. T. Yeh, J. S. Abel, and J. O. Smith, "Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations," in *Proc. 10th Int. Conf. on Digital Audio Effects (DAFx07)*, Bordeaux, France, 2007, pp. 197–204.

[12] D.T. Yeh, J.S. Abel, and J.O. Smith, "Automated physical modeling of nonlinear audio circuits for real-time audio effects; part I: Theoretical development," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 18, no. 4, pp. 728–737, May 2010.

[13] 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.

[14] R. C. D. de Paiva, J. Pakarinen, V. Välimäki, and M. Tikander, "Real-time audio transformer emulation for virtual tube amplifiers," *EURASIP J. Advances Signal Process.*, 2011.

[15] R. Hoffmann-Burchardi, "Digital simulation of the diode ring modulator for musical applications," in *Proc. 11th Int. Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland, Sept. 2008.

[16] J. D. Parker, "A simple digital model of the diode-based ring-modulator," in *Proc. 14th Int. Conf. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 2011, pp. 163–166.

[17] A. Huovilainen, "Enhanced digital models for analog modulation effects," in *Proc. Int. Conf. Digital Audio Effects*, Madrid, Spain, 2005, pp. 155–160.

[18] C. Raffel and J. O. Smith, "Practical modeling of bucket-brigade device circuits," in *Proc. 13th Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sep. 2010, pp. 50–56.

[19] F. Esqueda, V. Välimäki, and S. Bilbao, "Aliasing reduction in soft-clipping algorithms," in *23rd European Signal Processing Conference (EUSIPCO)*, Nice, France, Aug. 2015, pp. 2014–2018.

[20] B. De Man and J. D. Reiss, "Adaptive control of amplitude distortion effects," in *Proc. Audio Eng. Soc. 53rd Int. Conf.*, Jan 2014.

[21] J. Kleimola, V. Lazzarini, J. Timoney, and V. Välimäki, "Vector phase shaping synthesis," in *Proc. 14th Int. Conf. Digital Audio Effects (DAFx-11)*, Paris, France, 2011.

[22] 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, 2012.

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

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

[25] V. Välimäki, J. Pekonen, and J. Nam, "Perceptually informed synthesis of bandlimited classical waveforms using integrated polynomial interpolation," *J. Acoust. Soc. Am.*, vol. 131, pp. 974–986, 2012.

[26] G. Geiger, "Table lookup oscillators using generic integrated wavetables," in *Proc. 9th Int. Conf. on Digital Audio Effects (DAFx-06)*, Montreal, Canada, 2006, p. 169.

[27] A. Franck and V. Välimäki, "Higher-order integrated wavetable and sampling synthesis," *J. Audio Eng. Soc.*, vol. 61, no. 9, pp. 624–636, 2013.

[28] V. Zavalishin, *Preserving the LTI system topology in s- to z-plane transforms*, Native Instruments, 2008.

A. DERIVATION OF RESOLVED OUTPUT IN ILL-CONDITIONED AREA AND SYSTEM RESPONSE IN LINEAR CASE

Let us consider one linear segment of $\tilde{x}(t)$, the continuous-time approximation of the input signal, being processed by the nonlinearity f:

$$y(\tau) = f(x_n + \tau(x_{n-1} - x_n)) \tag{30}$$

In order to calculate the output of the continuous-time convolution, we must calculate one or more integrals of the form:

$$\int_0^1 y(\tau)\bar{h}(\tau)\,d\tau \tag{31}$$

where $\bar{h}$ is one particular *unipolar* piecewise segment of the convolution kernel. The analytical expressions for integrals (31) become ill-conditioned at $x_n \approx x_{n-1}$, therefore we wish to obtain alternative expressions that can be substituted for the ill-conditioned case.

(31) can be seen as a weighted average of $y(\tau)$, where $\bar{h}(\tau)$ is the weight function. From the mean value theorem there exists $\tau_0 \in [0, 1]$ such that:

$$\int_0^1 y(\tau)\bar{h}(\tau)\,d\tau = y(\tau_0)\int_0^1 \bar{h}(\tau)\,d\tau \tag{32}$$

For $x_n \approx x_{n-1}$, in principle any $\tau_0 \in [0, 1]$ can be chosen for the approximation. However, we wish to find the most optimal choice. Introducing a normalized weight function:

$$w(\tau) = \bar{h}(\tau) \Big/ \int_0^1 \bar{h}(\tau)\,d\tau \tag{33}$$

we rewrite (32) as:

$$\int_0^1 y(\tau)w(\tau)\,d\tau = y(\tau_0) \tag{34}$$

For $x_n \approx x_{n-1}$ the function $y(\tau)$ can be considered approximately linear on $[0, 1]$:

$$y(\tau) = a + b\tau + O((x_n - x_{n-1})^2) \tag{35}$$

where the error term is assuming the bounded second derivative of $f(x)$. From (34) we then have:

$$\begin{aligned} y(\tau_0) &= \int_0^1 y(\tau)w(\tau)\,d\tau \\ &= \int_0^1 (a + b\tau + O((x_n - x_{n-1})^2))w(\tau)\,d\tau \\ &= (a + O((x_n - x_{n-1})^2))\int_0^1 w(\tau)\,d\tau + b\int_0^1 \tau w(\tau)\,d\tau \\ &= a + bM_1 + O((x_n - x_{n-1})^2) \\ &= y(M_1) + O((x_n - x_{n-1})^2) \end{aligned} \tag{36}$$

where

$$M_1 = \int_0^1 \tau w(\tau)\,d\tau \tag{37}$$

is the first moment of the weight function $w(\tau)$. That is

$$y(\tau_0) = y(M_1) + O((x_n - x_{n-1})^2) \tag{38}$$

and therefore we choose

$$\tau_0 = M_1 = \int_0^1 \tau\bar{h}(\tau)\,d\tau \Big/ \int_0^1 \bar{h}(\tau)\,d\tau \tag{39}$$

Using this τ_0 and (32), we can then compute the integrals (31) in the ill-conditioned case.

Further, for $f(x) = x$ the equation (30) turns into

$$y(\tau) = x_n + \tau(x_{n-1} - x_n)) \tag{40}$$

while the term $O((x_n - x_{n-1})^2)$ vanishes from (35) and the following equations. Using (40) and (39) we turn (32) into

$$\int_0^1 y(\tau)\bar{h}(\tau)\,d\tau = (x_n + (x_{n-1} - x_n)M_1)\int_0^1 \bar{h}(\tau)\,d\tau \tag{41}$$

which allows us to calculate the response of the system in cases where $f(x)$ is approximately transparent.