

## ANTIDERIVATIVE ANTIALIASING FOR STATEFUL SYSTEMS

Martin Holters

Department of Signal Processing and Communication  
 Helmut Schmidt University  
 Hamburg, Germany  
 martin.holters@hsu-hh.de

### ABSTRACT

Nonlinear systems, like e.g. guitar distortion effects, play an important role in musical signal processing. One major problem encountered in digital nonlinear systems is aliasing distortion. Consequently, various aliasing reduction methods have been proposed in the literature. One of these is based on using the antiderivative of the nonlinearity and has proven effective, but is limited to memoryless systems. In this work, it is extended to a class of stateful systems which includes but is not limited to systems with a single one-port nonlinearity. Two examples from the realm of virtual analog modeling show its applicability to and effectiveness for commonly encountered guitar distortion effect circuits.

### 1. INTRODUCTION

Nonlinear systems play an important role in musical signal processing. In particular, there are many effects categorized as overdrive, distortion, or fuzz, whose primary objective it is to introduce harmonic distortion to enrich the signal. Usually the nonlinear behavior is in some way combined with (linear) filtering to spectrally shape the output signal or to make the amount of distortion introduced frequency dependent. While many of these systems were originally designed in the analog domain, naturally, there is interest in deriving digital models for them, e.g. [1, 2, 3, 4].

One major problem encountered in digital nonlinear systems, whether designed from scratch or derived by virtual analog modeling, is aliasing distortion. Once the additional harmonics introduced by the nonlinearity exceed the Nyquist frequency, they get folded back to lower frequencies, just as if the corresponding analog signal had been sampled without appropriate band-limiting. Contrary to the desired harmonic distortion, aliasing distortion is usually perceived as unpleasant. Therefore methods to suppress or reduce the aliasing distortion are needed.

The conceptually simplest aliasing reduction method is oversampling. However, if the harmonics decay slowly with frequency, the oversampling factor has to be high, making the approach unattractive due to the rising computational demand. Consequently, various alternatives have been proposed, e.g. [5, 6, 7, 8]. These methods, however, usually come with certain limitations, most commonly the restriction to memoryless systems. In this work, an extension of [7] is presented that loosens the restriction from memoryless systems to a certain class of stateful systems.

*Copyright: © 2019 Martin Holters et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*

### 2. ANTIDERIVATIVE-BASED ALIASING REDUCTION FOR MEMORYLESS NONLINEAR SYSTEMS

As the proposed method builds upon the approach from [7], we shall briefly summarize the latter. Conceptually, the digital signal is converted to a continuous-time signal using linear interpolation between consecutive samples, the nonlinearity is applied, and the result is lowpass-filtered by integrating over one sampling interval before sampling to obtain the digital output signal. The key insight is that, as there is a linear relationship between time and input signal amplitude (within one sampling interval), one can substitute the integration variable to integrate over amplitude instead of time. Then, by the fundamental theorem of calculus, one only needs to evaluate the antiderivative of the nonlinear mapping function at the input sample amplitudes and does not need to explicitly form the continuous signal. (For a more detailed explanation, the reader is referred to [7].)

The result is that the nonlinear system

$$y(n) = f(u(n)), \quad (1)$$

where  $f(u)$  denotes the nonlinear function, mapping input sample  $u(n)$  to output sample  $y(n)$ , is replaced with

$$y(n) = \tilde{f}(u(n), u(n-1)) = \begin{cases} \frac{F(u(n)) - F(u(n-1))}{u(n) - u(n-1)} & \text{if } u(n) \neq u(n-1) \\ f(\frac{1}{2}u(n) + \frac{1}{2}u(n-1)) & \text{if } u(n) \approx u(n-1) \end{cases} \quad (2)$$

where  $F(u) = \int f(u)du$  is the antiderivative of  $f(u)$  and the  $u(n) \approx u(n-1)$  case is treated separately to avoid numerical issues when dividing by  $u(n) - u(n-1)$ . In addition to reducing aliasing artifacts, the approach introduces a half-sample delay and attenuates high frequencies. This can be readily seen when using the identity function  $f(u) = u$  instead of a true nonlinearity. Straight forward calculation yields

$$y(n) = \frac{1}{2}u(n) + \frac{1}{2}u(n-1) \quad (3)$$

in that case, i.e. a first-order FIR low-pass filter with a group delay of half a sample. The low-pass effect can be countered by a modest amount of oversampling (e.g. by a factor of two) and the delay usually is of no concern.

### 3. EXTENSION TO STATEFUL SYSTEMS

The half-sample delay introduced by the method of [7] becomes problematic if the nonlinearity is embedded in the feedback loop of a stateful system. As noted in [7], for the particular case of an integrator following the nonlinearity and using trapezoidal rule for

time-discretization, one can simply replace the numerator of the discretized integrator's transfer function with the filter introduced by antialiasing. This fusing of antialiased nonlinearity and integrator then has no additional delay compared to the nonantialiased system, hence can be used inside a feedback system without problems.

Here, we consider systems which do not necessarily have an integrator following the nonlinearity. In particular, we shall consider the general discrete nonlinear state-space system

$$\mathbf{x}(n) = \mathbf{A}\mathbf{x}(n-1) + \mathbf{b}u(n) + \mathbf{f}_x(p_x(n)) \quad (4)$$

$$y(n) = \mathbf{c}^T \mathbf{x}(n-1) + du(n) + f_y(p_y(n)) \quad (5)$$

with

$$p_x(n) = \mathbf{c}_{p_x}^T \mathbf{x}(n-1) + d_{p_x} u(n) \quad (6)$$

$$p_y(n) = \mathbf{c}_{p_y}^T \mathbf{x}(n-1) + d_{p_y} u(n), \quad (7)$$

where  $\mathbf{x}(n)$  is the state vector,  $u(n)$  in the input,  $y(n)$  is the output,  $\mathbf{A}$  is the state matrix,  $\mathbf{b}$  is the input matrix,  $\mathbf{c}^T$  is the output matrix, and  $d$  is the feedthrough matrix, where the latter three are reduced to vectors and a scalar, respectively, as we only consider scalar input and output. The nonlinearity of the system is captured in two nonlinear functions,  $\mathbf{f}_x$  and  $f_y$ , influencing state update and output, respectively. Their arguments  $p_x(n)$  and  $p_y(n)$  are calculated by (6) and (7) similarly to the linear part of the output equation (5). Some remarks are in order:

- While we allow multiple states, collected in the vector  $\mathbf{x}(n)$ , we restrict the presentation to a single input  $u(n)$  and a single output  $y(n)$ , as that is the most common case. Extension to multiple inputs and/or outputs is straight-forward.
- The limitation to scalar-valued  $p_x(n)$  and  $p_y(n)$ , however, is necessary, as the method of [7] is restricted to nonlinear functions with scalar argument. Facilitating this is the reason why the linear parts  $\mathbf{A}\mathbf{x}(n-1) + \mathbf{b}u(n)$  and  $\mathbf{c}^T \mathbf{x}(n-1) + du(n)$  have not been subsumed in the nonlinear functions in (4) and (5), respectively.
- If the system is obtained in the context of virtual analog modeling, usually the nonlinear functions will only be given implicitly (as the solution of what is sometimes referred to as a delay-free loop), making solving a nonlinear equation necessary. However, they are typically based on a common function, only applying different weighting to its output, i.e.  $\mathbf{f}_x(p_x(n)) = \mathbf{W}_x \mathbf{f}(p(n))$  and  $f_y(p_y(n)) = \mathbf{w}_y^T \mathbf{f}(p(n))$  with  $p(n) = p_x(n) = p_y(n)$ . While this redundancy should be kept in mind for optimizing an implementation, we will derive our method for the more general case of two possibly independent nonlinear functions for state update and output.

In a first step, we may consider only applying the aliasing suppression to  $f_y(p)$ , as it is not part of any feedback loop. We have to be careful though, and may not just replace  $f_y$  with  $\tilde{f}_y$  in (5), as that would lead to a misalignment in time of the different summed terms. Instead, we have to use

$$y(n) = \frac{1}{2} \mathbf{c}^T (\mathbf{x}(n-1) + \mathbf{x}(n-2)) + \frac{1}{2} d(u(n) + u(n-1)) + \tilde{f}_y(p_y(n), p_y(n-1)). \quad (8)$$

However, any aliasing distortion introduced into  $\mathbf{x}(n)$  by (4) will not undergo any mitigation (except for the lowpass filtering).

Now, if we naively rewrite (4) as we did with (5), we modify our system in an unwanted way as we introduce additional delay in the feedback. But we do that in a very controlled way: The unit delay in the feedback is replaced by a delay of 1.5 samples. This is equivalent to reducing the sampling rate by a factor of 1.5, so we can compensate by designing our system for this reduced sampling rate, arriving at

$$\mathbf{x}(n) = \frac{1}{2} \tilde{\mathbf{A}}(\mathbf{x}(n-1) + \mathbf{x}(n-2)) + \frac{1}{2} \tilde{\mathbf{b}}(u(n) + u(n-1)) + \tilde{f}_x(p_x(n), p_x(n-1)) \quad (9)$$

$$y(n) = \frac{1}{2} \tilde{\mathbf{c}}^T (\mathbf{x}(n-1) + \mathbf{x}(n-2)) + \frac{1}{2} \tilde{d}(u(n) + u(n-1)) + \tilde{f}_y(p_y(n), p_y(n-1)). \quad (10)$$

with

$$p_x(n) = \tilde{\mathbf{c}}_{p_x}^T \mathbf{x}(n-1) + \tilde{d}_{p_x} u(n) \quad (11)$$

$$p_y(n) = \tilde{\mathbf{c}}_{p_y}^T \mathbf{x}(n-1) + \tilde{d}_{p_y} u(n) \quad (12)$$

where all coefficients are calculated for the reduced sampling rate  $\tilde{f}_s = \frac{2}{3} f_s$ . We can only do this because we do not have a delay-free loop. Or rather, the delay-free loop is hidden inside  $f(u)$ : Instead of worrying about a nonlinearity within a delay-free loop, we treat the solution of the delay-free loop as the nonlinearity to apply aliasing reduction to. Note that the behavior for frequencies above  $\frac{1}{2} \tilde{f}_s = \frac{1}{3} f_s$  is ill-defined, but with the mild oversampling suggested by [7] anyway, we do not have to worry about this.

The increased delay is not the only effect of the modification. There is also the low-pass filtering. To study this in more detail, assume  $\mathbf{f}_x(p_x)$  and  $f_y(p_y)$  to be linear so that we have a linear system, and let  $H(z)$  denote the transfer function obtained from (4)–(7). If we instead use (9)–(12) without adjusting the coefficients, it is straight forward to verify that the resulting transfer function fulfills

$$\tilde{H}(z) = \frac{1}{2} (1 + z^{-1}) \cdot H\left(\left(\frac{1}{2}(z^{-1} + z^{-2})\right)^{-1}\right). \quad (13)$$

We may observe two effects: The well-known filtering with a factor on the outside and the substitution  $z \leftarrow \left(\frac{1}{2}(z^{-1} + z^{-2})\right)^{-1}$  in the argument of  $H$ . Evaluating the latter for  $z = e^{j\omega}$ , we note that

$$\left(\frac{1}{2}(e^{-j\omega} + e^{-2j\omega})\right)^{-1} = \frac{1}{\cos(\frac{1}{2}\omega)} e^{\frac{3}{2}j\omega} \quad (14)$$

depicted in figure 1. While in the original system  $H(z)$  is evaluated on the unit circle  $e^{j\omega}$  (shown dotted) to obtain the frequency response, for the modified system, it is evaluated on the trajectory of (14). We notice that, in addition to the frequency scaling by  $\frac{3}{2}$ , there is an additional scaling away from the unit circle, increasing with frequency. Importantly, as we only evaluate  $H(z)$  for  $z$  on or outside the unit circle, we preserve stability, i.e. if  $H(z)$  is stable, so is  $\tilde{H}(z)$ . Nevertheless, especially for higher frequencies, this may cause a significant distortion of the frequency response.

An extreme example would be an all-pass filter with high Q-factor, where the transformation might result in the zero moving onto the frequency axis, turning a flat frequency response into one with a deep notch. As the examples will demonstrate, many typical systems are rather well-behaved under the transformation, but one has to be aware of this pitfall.

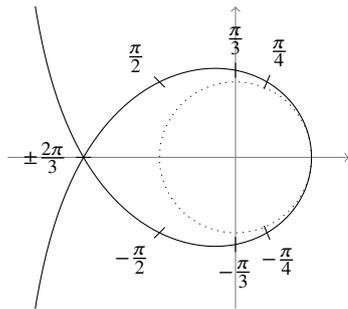


Figure 1: Trajectory of  $(\frac{1}{2}e^{-j\omega} + \frac{1}{2}e^{-2j\omega})^{-1}$  compared to the unit circle  $e^{j\omega}$  (dotted)

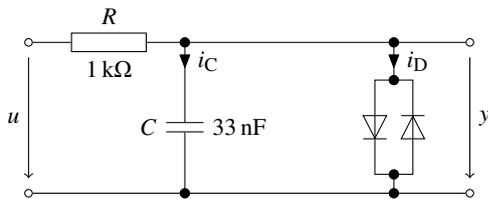


Figure 2: Schematics of the modeled diode clipper

## 4. EXAMPLES

### 4.1. Diode clipper

As a first example, we consider the diode clipper of figure 2. The circuit is simple enough that we briefly repeat the modeling process here.

From Kirchhoff's and Ohm's laws and  $i_C = C\dot{y}$ , we immediately obtain

$$y = u - R \cdot (i_C + i_D) = u - RC\dot{y} - Ri_D. \quad (15)$$

Summing over two subsequent sampling instances, we get

$$\begin{aligned} y(n) + y(n-1) = \\ u(n) + u(n-1) - RC(\dot{y}(n) + \dot{y}(n-1)) - R(i_D(n) + i_D(n-1)). \end{aligned} \quad (16)$$

We now use the trapezoidal rule to substitute

$$\dot{y}(n) + \dot{y}(n-1) = 2f_s(y(n) - y(n-1)) \quad (17)$$

and obtain

$$\begin{aligned} y(n) + y(n-1) = \\ u(n) + u(n-1) - 2RCf_s(y(n) - y(n-1)) - R(i_D(n) + i_D(n-1)). \end{aligned} \quad (18)$$

Collecting all quantities from time step  $n-1$  into canonical states

$$x(n-1) = (2RCf_s - 1)y(n-1) + u(n-1) - Ri_D(n-1) \quad (19)$$

allows simplification to

$$y(n) = x(n-1) + u(n) - 2RCf_sy(n) - Ri_D(n). \quad (20)$$

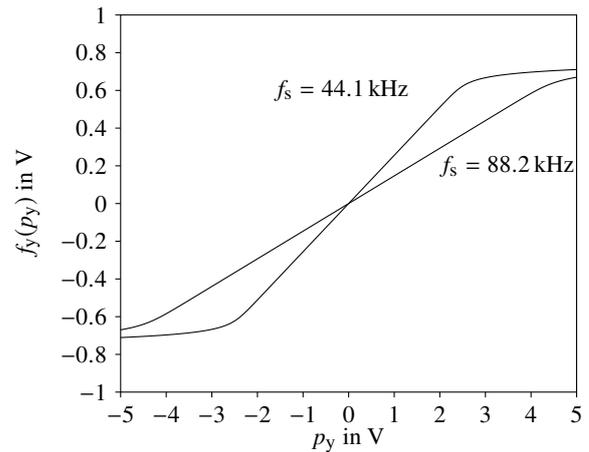


Figure 3: Nonlinear function  $f_y(p_y)$  of the diode clipper for two different sampling rates  $f_s$

Using Shockley's equation for the diodes, we get

$$\begin{aligned} i_D(n) = I_S \cdot (e^{y(n)/v_T} - 1) - I_S \cdot (e^{-y(n)/v_T} - 1) \\ = 2I_S \sinh(y(n)/v_T), \end{aligned} \quad (21)$$

where saturation current and temperature voltage have been chosen as  $I_S = 1$  fA and  $v_T = 25$  mV respectively. Inserting (21) into (20) and introducing

$$p_y(n) = x(n-1) + u(n) \quad (22)$$

then leads to the implicit equation

$$y(n) = p_y(n) - 2RCf_sy(n) - 2RI_S \sinh(y(n)/v_T) \quad (23)$$

for  $y(n)$ . Note that we do not treat this as a delay-free loop and apply the antialiasing to the sinh-function. Instead, we let  $f_y(p_y(n)) = y(n)$  denote the solution of the implicit equation. The resulting function is depicted in figure 3 (obtained using an iterative solver).

To obtain the state update equation, we rearrange (20) to

$$(2RCf_s - 1)y(n) + u(n) - Ri_D(n) = -x(n-1) + 4RCf_sy(n) \quad (24)$$

and note by comparing with (19) that the left-hand side equals  $x(n)$ . Thus introducing

$$f_x(p_x(n)) = 4RCf_sf_y(p_y(n)) \quad (25)$$

with  $p_x(n) = p_y(n)$ , we arrive at

$$x(n) = -x(n-1) + f_x(p_x(n)) \quad (26)$$

$$y(n) = f_y(p_y(n)) \quad (27)$$

of the desired form.

Applying the aliasing mitigation only to the output equation is particularly simple in this case, giving

$$y(n) = \tilde{f}_y(p_y(n), p_y(n-1)) \quad (28)$$

with  $\tilde{f}_y$  defined according to (2). The required antiderivative  $F_y(p_y)$  of  $f_y(p_y)$ , depicted in figure 4, has to be determined numerically. For the results below, it has been precomputed at 1024 uniformly

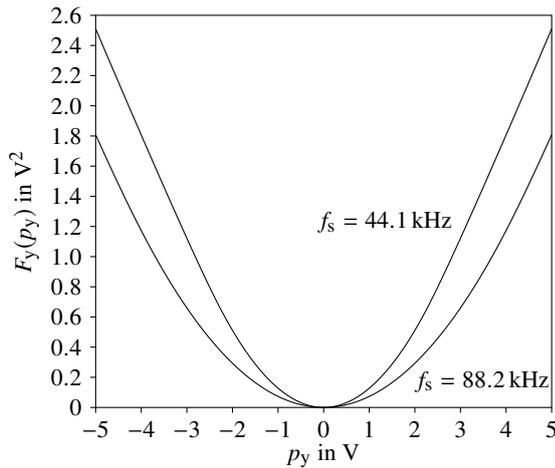


Figure 4: Antiderivative  $F_y(p_y)$  of  $f_y(p_y)$  of the diode clipper for two different sampling rates  $f_s$

distributed points in the relevant range and stored in a table, using cubic interpolation during lookup.

To also apply aliasing mitigation to the state update equation, we have to change it to

$$x(n) = -\frac{1}{2}(x(n-1) + x(n-2)) + \tilde{f}_x(p_x(n), p_x(n-1)) \quad (29)$$

and substitute  $\tilde{f}_s = \frac{2}{3}f_s$  for  $f_s$  in (23) and (25). Note that this immediately leads to

$$\tilde{f}_x(p_x(n), p_x(n-1)) = 4RC\tilde{f}_s\tilde{f}_y(p_y(n), p_y(n-1)). \quad (30)$$

To study the effectiveness of the method, we consider figure 5, where the output spectra for a sinusoidal excitation are depicted for various model configurations. Figures 5(a) and 5(b) give the baseline, the system without any aliasing mitigation at sampling rates  $f_s = 44.1$  kHz and  $f_s = 88.2$  kHz, respectively. Only applying aliasing mitigation to the output equation according to (28) is of little benefit, as seen when considering figures 5(c) and 5(d) in comparison. We do note, however, the low-pass effect in figure 5(c), where higher harmonics exhibit an attenuation of up to 10 dB.

When also applying the aliasing mitigation to the state update equation according to (29), we observe a significant aliasing reduction in figures 5(e) and 5(f). As explained, the aliasing mitigation should be combined with (modest) oversampling. In this particular case, as verified in figure 5(e), the model is still a relatively good fit even without oversampling, which however must be mainly attributed to lucky coincidence. More relevant is the case of a sampling rate of  $f_s = 88.2$  kHz, shown in figure 5(f). Comparing to oversampling to  $f_s = 220.5$  kHz without additional aliasing mitigation measures, as shown in figure 5(g), we see that the aliased components at low frequencies, where they are most easily perceived, are at a comparable level.

#### 4.2. Tube screamer-like distortion circuit

As a second example we consider the distortion circuit of figure 6, inspired by the Ibanez Tube Screamer TS-808. We shall not go into details of the modeling procedure (for which we have used ACME.jl<sup>1</sup>), but remark that if one allows the three diodes to be

<sup>1</sup><https://github.com/HSU-ANT/ACME.jl>

different, one can no longer derive a closed-form expression for their combined behavior. Instead, the nonlinear behavior is determined by a system of three equations. Nevertheless, using the dimensionality reduction approach of [9], the input  $p_x(n) = p_y(n)$  to the nonlinearity can be reduced to a scalar value, formed by linear combination of the input and the capacitor states. Hence, the proposed method is applicable.

Figure 7 again shows the output spectra for a sinusoidal excitation. As can be seen in figure 7(a), with plain oversampling to  $f_s = 88.2$  kHz, the signal contains strong aliasing components. Applying aliasing mitigation only to the output equation reduces the aliasing distortion to a limited extent, as shown in figure 7(b). In contrast, when also applying aliasing mitigation to the state update equation, the aliasing is significantly reduced, as seen in figure 7(c). Again, the aliasing mitigation is most effective at low frequencies, where it is also perceptually most relevant. As in the diode clipper example, for low frequencies the system with aliasing mitigation at  $f_s = 88.2$  kHz performs at least as good as an unmodified system at  $f_s = 220.5$  kHz, see figure 7(d).

## 5. CONCLUSION AND OUTLOOK

The presented approach for aliasing reduction generalizes the approach of [7] to all nonlinear systems that can be cast in a way that the nonlinearity takes a scalar input. This includes, but is not limited to, all models of circuits with a single one-port nonlinear element. If the system contains a delay-free loop, it has to be re-cast such that the nonlinearity is defined as the solution of the delay-free loop. Then, the delay introduced by applying the method of [7] to the nonlinearity can be compensated by adjusting the system's coefficients, even if the nonlinearity is part of a feedback loop.

As is to be expected, the achieved aliasing reduction is comparable to that of [7], allowing to significantly reduce the required oversampling especially for systems which introduce strong distortion, while the additional computational load is modest. Assuming lookup tables are used for  $f(u)$  (in general being implicitly defined) and its antiderivative  $F(u)$ , the main price to pay is in terms of memory used.

It should be noted that the extensions to higher order antiderivatives as proposed in [10] or [11] should be straight-forward, following the same principle. A more interesting future direction would be to lift the restriction on the nonlinear function to have only scalar input. If the method of [7] (or even the higher order extensions of [10] or [11]) could be generalized to nonlinear functions with multiple inputs, the method proposed in the present paper would immediately generalize to all stateful nonlinear systems.

## 6. REFERENCES

- [1] David T. Yeh, Jonathan S. Abel, and Julius O. Smith III, "Simplified, physically-informed models of distortion and overdrive guitar effects pedals," in *Proc. 10th Int. Conf. on Digital Audio Effects (DAFx-07)*, Bordeaux, France, 2007.
- [2] Martin Holters, Kristjan Dempwolf, and Udo Zölzer, "A digital emulation of the Boss SD-1 Super Overdrive pedal based on physical modeling," in *131st AES Convention*, New York, NY, USA, 2011.
- [3] W. Ross Dunkel, Maximilian Rest, Kurt James Werner, Michael Jørgen Olsen, and Julius O. Smith III, "The Fender Bassman 5F6-A family of preamplifier circuits — a wave

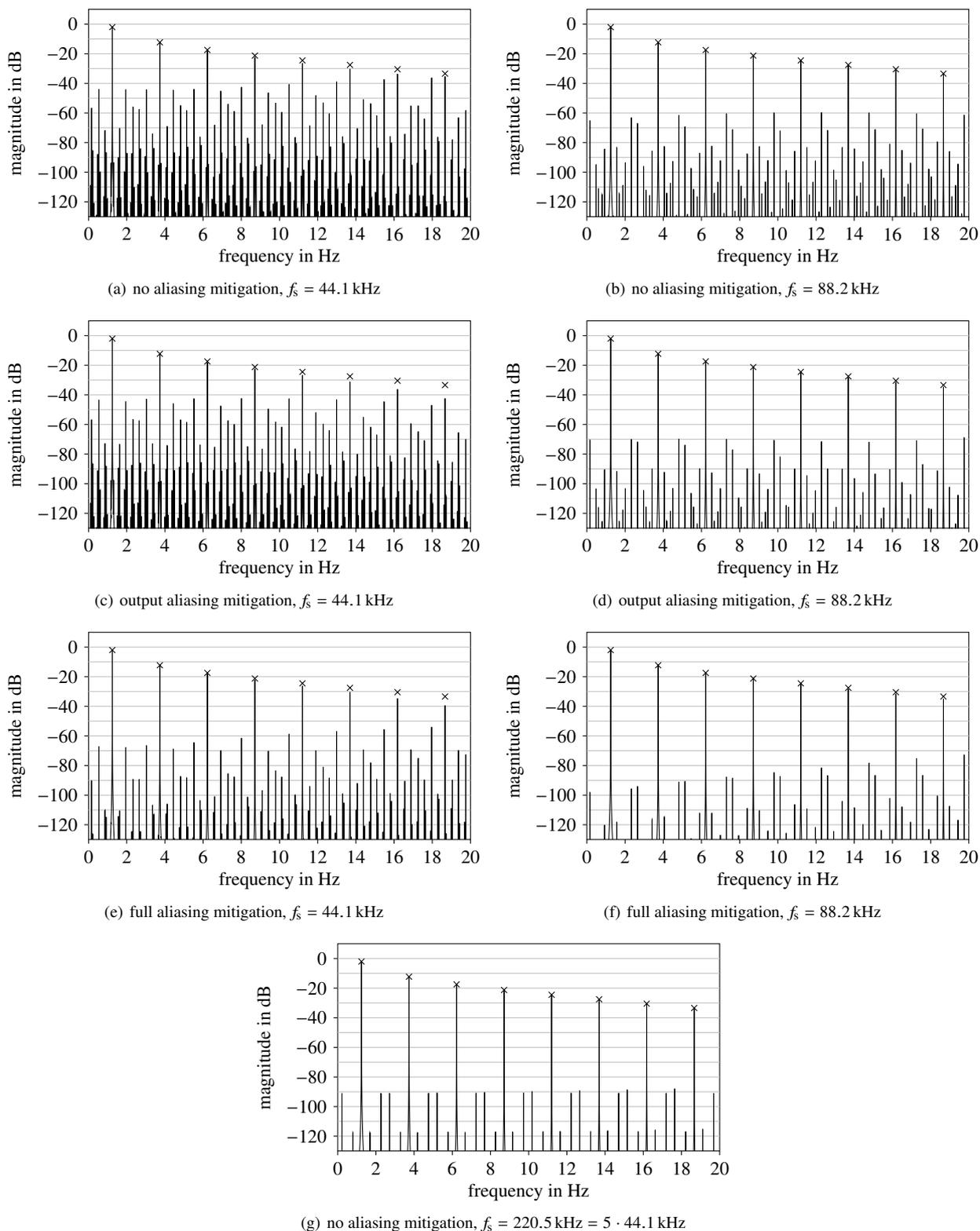


Figure 5: Output spectra of various model configurations for the diode clipper when excited with a single sinusoid of 10 V amplitude at 1244.5 Hz. Crosses mark expected harmonics.

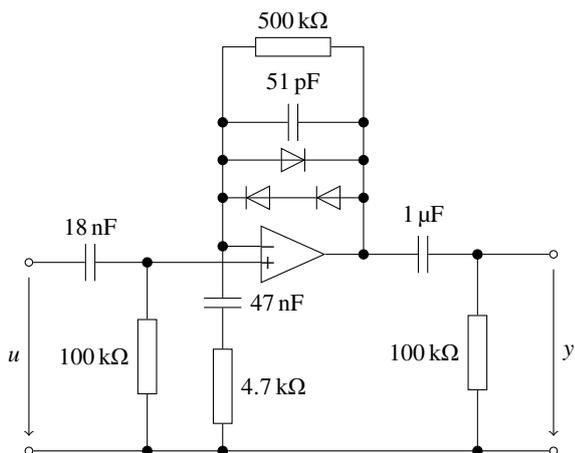
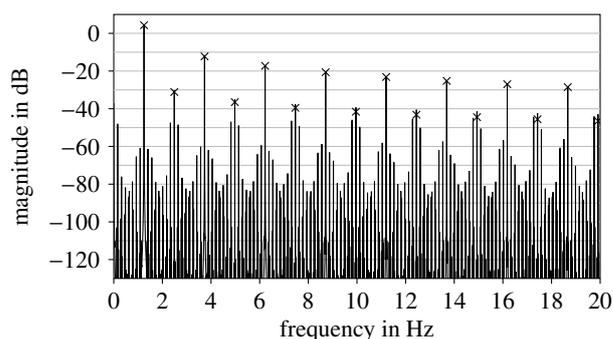


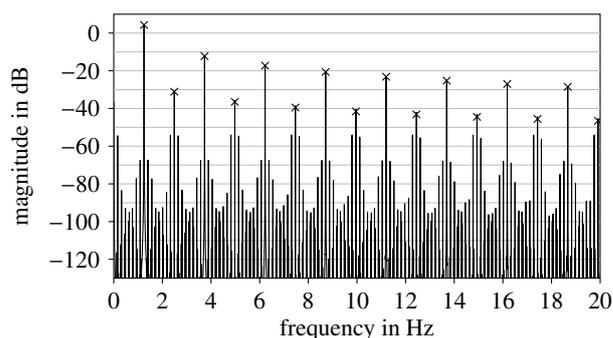
Figure 6: Tube screamer-like distortion circuit

digital filter case study,” in *Proc. 19th Int. Conf. on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016, pp. 263–270.

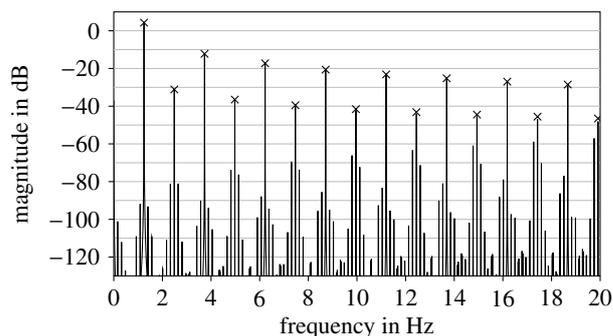
- [4] Fabián Esqueda, Henri Pöntynen, Vesa Välimäki, and Julian D. Parker, “Virtual analog Buchla 259 wavefolder,” in *Proc. 20th Int. Conf. on Digital Audio Effects (DAFx-17)*, Edinburgh, UK, 2017, pp. 192–199.
- [5] Fabián Esqueda and Vesa Välimäki, “Rounding corners with BLAMP,” in *Proc. 19th Int. Conf. on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016, pp. 121–128.
- [6] Fabián Esqueda, Vesa Välimäki, and Stefan Bilbao, “Antialiased soft clipping using an integrated bandlimited ramp,” in *Proc. 24th European Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, 2016, pp. 1043–1047.
- [7] Julian D. Parker, Vadim Zavalishin, and Efflam Le Bivic, “Reducing the aliasing of nonlinear waveshaping using continuous-time convolution,” in *Proc. 19th Int. Conf. on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016, pp. 137–144.
- [8] Rémy Muller and Thomas Hélie, “Trajectory anti-aliasing on guaranteed-passive simulation of nonlinear physical systems,” in *Proc. 20th Int. Conf. on Digital Audio Effects (DAFx-17)*, Edinburgh, UK, 2017, pp. 87–94.
- [9] Martin Holters and Udo Zölzer, “A k-d tree based solution cache for the non-linear equation of circuit simulations,” in *Proc. 24th European Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, 2016, pp. 1028–1032.
- [10] Stefan Bilbao, Fabián Esqueda, Julian D. Parker, and Vesa Välimäki, “Antiderivative antialiasing for memoryless nonlinearities,” *IEEE Signal Process. Lett.*, vol. 24, no. 7, pp. 1049–1053, 2017.
- [11] Stefan Bilbao, Fabian Esqueda, and Vesa Välimäki, “Antiderivative antialiasing, lagrange interpolation and spectral flatness,” in *2017 IEEE Workshop on Appl. of Signal Process. to Audio and Acoust. (WASPAA)*, New Paltz, NY, USA, 2017, pp. 141–145.



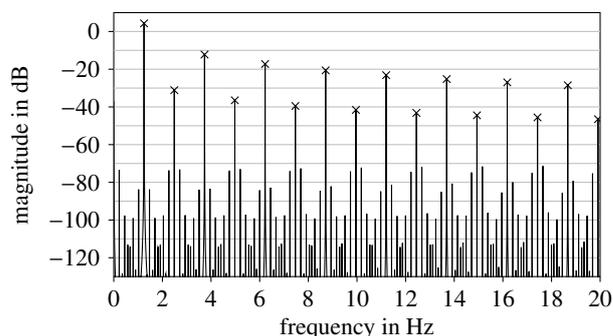
(a) no aliasing mitigation,  $f_s = 88.2$  kHz



(b) output aliasing mitigation,  $f_s = 88.2$  kHz



(c) full aliasing mitigation,  $f_s = 88.2$  kHz



(d) no aliasing mitigation,  $f_s = 220.5$  kHz =  $5 \cdot 44.1$  kHz

Figure 7: Output spectra of various model configurations for the tube screamer distortion circuit when excited with a single sinusoid of 1 V amplitude at 1244.5 Hz. Crosses mark expected harmonics.