ANTIALIASING IN BBD CHIPS USING BLEP

Leonardo Gabrielli
Dept. of Information Engineering
Università Politecnica delle Marche
Ancona, IT
l.gabrielli@staff.univpm.it

Stefano D'Angelo
Orastron Srl
Agropoli, IT
stefano.dangelo@orastron.com

Stefano Squartini
Dept. of Information Engineering
Università Politecnica delle Marche
Ancona, IT
s.squartini@staff.univpm.it

ABSTRACT

Several methods exist in the literature to accurately simulate Bucket Brigade Device (BBD) chips, which are widely used in analog delay-based audio effects for their characteristic lo-fi sound, which is affected by noise, nonlinearities and aliasing. The latter is a desired quality, being typical of those chips. However, when simulating BBDs in a discrete-time domain environment, additional aliasing components occur that need to be suppressed. In this work, we propose a novel method that applies the Bandlimited Step (BLEP) technique, effectively minimizing aliasing artifacts introduced by the simulation. The paper provides some insights on the design of a BBD simulation using interpolation at the input for clock rate conversion and, most importantly, shows how BLEP can be effective in reducing unwanted aliasing artifacts. Interpolation is shown to have minor importance in the reduction of spurious components.

1. INTRODUCTION

Delays are fundamental components in numerous audio processing algorithms and effects [1]. Historically, early delay effects were achieved using tape machines, which allowed for the creation of echo and reverberation by recording and playing back audio signals at varying intervals. The emulation of tape delay behavior has been explored in various studies in the past [2, 3]. Following the era of tape-based delays, dedicated analog integrated circuits known as Bucket Brigade Devices (BBDs) were introduced in the late 1960s. As digital signal processing chips became available, digital delay effect units became popular [4], also paving the way for artificial reverberation algorithms [5]. BBD chips, however, remained in production and are still employed in some guitar delay effect pedals. Furthermore, their discrete-time implementation can be useful for virtual instruments and digital emulation of vintage hardware effects.

BBDs are analog devices that sample input signals and transfer them through a series of capacitors, effectively creating a discrete-time delay line with real-valued signals. The length of such a delay line is fixed as the path and the capacitors are implemented in a silicon chip, therefore the only way to vary the delay time is to adjust the clock rate F_{CLK} that drives the signal propagation along the delay line. The clock rate may also be time varying to implement effects such as chorus and flanger.

BBDs are well-known for their characteristic audio qualities that include companding, saturation, aliasing and noise. Indeed, aliasing is a well known artifact in BBD chips, since they perform discrete-time sampling. While aliasing can be mitigated by an antialias filter at the input of the BBD, the clock rate F_{CLK} driving the sampling must vary to adjust the delay length. This variation is often done in a continuous fashion, e.g. for flanging and chorusing, making it difficult for an analog filter to precisely follow the variation of the bandwidth according to the Nyquist-Shannon sampling theorem. For this reason, this type of aliasing is accepted as a part of the typical BBD chip sound. On the contrary, when simulating a BBD in a digital computer at a fixed sampling frequency F_s, undesired aliasing components can also occur, which are spurious and are not desired.

Several studies have addressed the digital emulation of BBD circuits, which are useful for creating virtual analog clones of vintage effects, or can add a touch of character to digital delay-based algorithms in stead of digital delay lines. The character of BBDs is related to their intrinsic construction, which entails imperfections like aliasing, distortion, and noise. One of the earliest attempt at a digital model for BBD is found in [6]. Raffel and Smith analyzed the sonic characteristics of BBDs, focusing on modeling their nonlinearities and associated filtering circuits [7]. Macák [8] simulated analog flanger effects using BBD circuits, emphasizing the importance of accurately replicating the low-frequency oscillator and clock generation circuits. Owing from fractionally delay addressed delay lines [9], Holters and Parker in [10] proposed a model for BBDs that combines fractional read and write with the input and output filters typical of a BBD, using a variable sample-rate approach. The approach is not immune to aliasing even if in many cases it is not practically an issue. The authors proposed oversampling as a solution, since it does not excessively increase the computational cost due to the fact that part of the algorithm is driven by the BBD clock rate.

While these studies have advanced the digital modeling of BBDs, the issue of retaining the natural aliasing occurring in BBD while eliminating the spurious aliasing due to the digital simulation remains underexplored. For the sake of simplicity, let us consider the case where $F_{CLK} < F_s$[1]. Intuitively, the BBD output changes value at the simulation time step n, driven by a lower rate clock. The output, thus, is stepped as in a sample-and-hold circuit. Being discontinuous, the stepped output can be affected by aliasing, that can be corrected by smoothing the discontinuities.

In this paper, we introduce a method to mitigate aliasing caused by these discontinuities by employing the Bandlimited Step (BLEP) technique [11] when reading the BBD delay line output at clock ticks. This approach preserves the desired artifacts associated with the BBD's sampling frequency, while reducing unwanted aliasing, leading to a more accurate and faithful digital emulation of BBD-based audio effects.

[1] Please notice that aliasing is an issue even in the case $F_{CLK} < F_s$.

The paper is accompanied by Octave code and audio examples[2] that are shared with the community.

The rest of the paper is organized as follows. Section 2 introduces the BBD and a trivial discrete-time simulation. Section 3 introduces the proposed method for mitigating aliasing in BBD delay lines using BLEP techniques. Section 4 describes the experiments conducted to evaluate the effectiveness of the proposed method and discusses the results. Finally, Section 5 concludes the paper by summarizing the findings.

2. OPERATING PRINCIPLE OF A BBD

A BBD is an analog delay line that consists of a cascade of N capacitors and MOS or bipolar transistors arranged in a sequential structure. The core principle behind a BBD is the controlled transfer of charge from one capacitor to the next in response to a clock signal, with the charge being directly proportional to the value of the input signal sampled at certain time instants.

Each cell in the BBD (except from the initial and ending stages) consists of a capacitor paired with a switching transistor that controls the charge transfer to the next cell. These transistor-capacitor pairs are arranged in a cascade, forming a long chain. The transistors operate as analog switches, controlled by two-phase non-overlapping clock signals (ϕ_1 and ϕ_2), which alternately open and close at a given clock frequency. The number of capacitor stages and the clock frequency contribute to the total delay introduced by the BBD.

As exhaustively explained in [10], the working principle of the BBD entails a delay time of $N/2$ samples for a N-stages long delay line, since at any time half of the capacitors carry a signal, while the other half is at the reference voltage.

Since the charge transfer process is not perfect, BBDs exhibit certain imperfections such as charge leakage, clock feedthrough, and signal degradation due to high-frequency attenuation and noise accumulation along the chain. Furthermore, BBD chips sample a signal at the clock rate that drives both ϕ_1 and ϕ_2. Therefore, BBD are discrete-time systems and can introduce aliasing. All the aforementioned artifacts contribute to the characteristic sound of BBD-based audio effects.

However, in the following, we are interested in modeling only the discrete-time behavior of the chip, leaving aside other artifacts due to the silicon technology involved. To that extent, the BBD cells will be treated as perfect charge storage devices without any leakage, noise or nonlinear behavior, making it easier to evaluate the impact of aliasing in the discrete-time simulation.

Indeed, for the rest of the paper we will always distinguish between aliasing components generated by the BBD discrete-time sampling at its clock rate F_{CLK} and the aliasing components generated by the computing environment where the discrete-time simulation occurs, which have a sampling rate of F_s. To avoid confusion, in the following, we will always refer to BBD-generated aliasing (BGA) and simulation-generated aliasing (SGA).

2.1. Trivial Discrete-Time Simulation

For the reasons mentioned above, to simulate a BBD in a discrete-time domain we will simply treat the BBD cells as we would in a digital delay line. However, at the input and output of the BBD there is a sampling rate conversion we must deal with. In this section we employ linear interpolation, for simplicity, and it will be applied at the input of the delay line. This makes it trivial to get the BBD output by simply letting out the last sample in the BBD cascade when the clock hits. The procedure is explained by the pseudocode of Algorithm 1. Please notice that for a N-stages BBD chip, and a given clock rate F_{CLK}, the stages must be considered equal to $N/2$.

Algorithm 1 Trivial algorithm for discrete-time BBD implementation.

```
Require: Input signal $x$, sampling rate $F_s$, BBD length $N$, clock
    frequency $F_{CLK}$
Ensure: Output signal $y$
 1: Initialize variables:
 2: $t \leftarrow 0.0$
 3: $d_l \leftarrow$ delay line of size $N/2$
 4: $r \leftarrow 0$                                        ▷ read index $\in \mathbb{R}$
 5: for $n = 1$ to length of $x$ do
 6:     $\Delta t \leftarrow F_{CLK}[n]/F_s$               ▷ allows for variable $F_{CLK}$
 7:     $t \leftarrow t + \Delta t$
 8:     while $t \geq 1.0$ do                               ▷ Clock fires
 9:         $d \leftarrow 1/\Delta t \cdot (1 - t) + 1.0$
10:         $d_l[r] \leftarrow d \cdot (x[n] - x[n-1]) + x[n-1]$       ▷
    interpolate input and write to delay line
11:         $r \leftarrow (r + 1) \mod (N/2)$
12:         $t \leftarrow t - 1.0$
13:     end while
14:     $y[n] \leftarrow d_l[r]$
15: end for
16: return $y$
```

3. PROPOSED METHOD

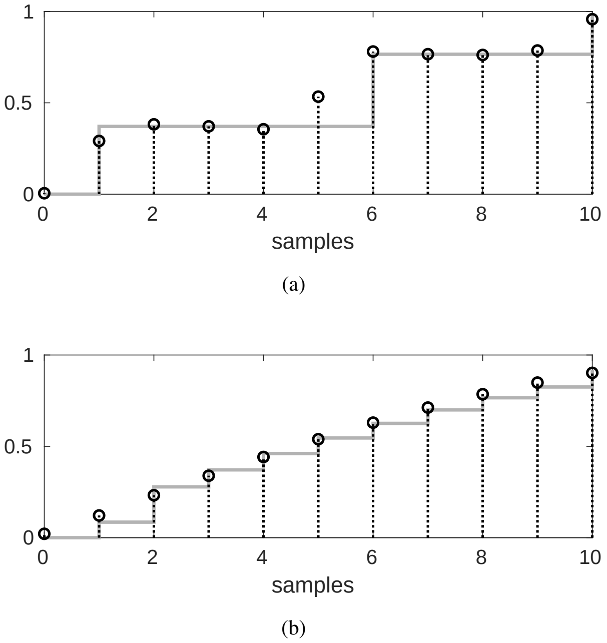
Considering the trivial implementation of the BBD of Algorithm 1, it is easy to notice that the output is stepped, as in a sample and hold, thus resulting in discontinuities in the signal that will extend the bandwidth of the signal well over the Nyquist limit and, thus, cause SGA. This is quite intuitive to see when $F_{CLK} < F_s$, see, e.g., Figure 1(a).

Drawing from the literature, at least one method is known to deal with asynchronous discontinuities in the signal, which is known as the BLEP method [11]. It was originally proposed to deal with hard sync signals, where discontinuities in an oscillator are caused by a second oscillator at arbitrary time intervals. The method has been successfully employed for virtual analog oscillators and was later extended to deal with discontinuities in the first derivative with the BLAMP technique [12].

For the stepped outputs of the BBD, the BLEP technique seems to be promising as it provides a straightforward solution to SGA by correcting the stepped waveform with the addition of samples that smooth the discontinuity, thus limiting the bandwidth of the signal.

The method requires knowledge of the time instants where each step occurs. If the algorithm described in Algorithm 1 is used, it is quite trivial to keep track of this instants, since they correspond to the discrete-time samples when the BBD clock is triggered (i.e. when the condition in line 8 is true). For SGA to be reduced, some signal samples surrounding (BLEP) or following (minBLEP) the discontinuity must be corrected by adding the BLEP correction coefficients.

[2] https://dangelo.audio/dafx2025-bbd.html



(a)

(b)

Figure 1: Detail of the BBD output in the time domain for (a) $F_{CLK} = 10\,\text{kHz}$ and (b) $F_{CLK} = 50\,\text{kHz}$. Simulation sampling rate is $F_s = 44.1\,\text{kHz}$. The trivial BBD output is shown in gray, while the output adjusted using BLEP is shown as a stem plot.

In this paper, for simplicity, we adopt the polyBLEP technique [13], which was shown to have low computational cost with respect to the BLEP and minBLEP methods. The technique consists in computing a closed form solution for the convolution integral between a rising edge of a continuous-time step and a suitable low-pass filtering function that mitigates the aliasing. The originally proposed method employed a triangular pulse, which has a $sinc^2$ magnitude frequency response. Higher-order solutions can be computed, but in this paper we will stick to that.

In the rest of the paper we adopt the aforementioned polynomial and correct the first two samples after and before the step, with a span of $N_B = 3$ discontinuities before and after the current one. The resulting solution is $\hat{y}[n] = y[n] - c[n]$, where $y[n]$ is the output of the BBD as per Algorithm 1, and the correction factor is

$$\begin{aligned} c[n] = &\sum_{j=1}^{N_B} (d_l[r-j] - d_l[r-j-1] \cdot \beta(d_{j-})) + \\ &+ \sum_{j=1}^{N_B} (d_l[r+j] - d_l[r+j-1] \cdot \beta(d_{j+})) \end{aligned} \tag{1}$$

where the function $\beta(\cdot)$ evaluates the polyBLEP residual [13] according to the fractional indices d_{j-} or d_{j+} depending whether we are looking forward or backward in time.

4. EXPERIMENTS

To evaluate the performance of the proposed method, two sets of experiments were conducted, one in static conditions, and another one with a time-varying clock rate. An additional experiment was conducted to evaluate the effect of higher order interpolation methods at the BBD input.

Our experiments focus on the aliasing artifacts of the simulated BBD system under different input conditions. In order to use real-world data, one BBD chip has been taken as a reference, the Panasonic MN3005, employed in many commercial products. It consists in a 4096-stage delay line that can be clocked at rates F_{CLK} from 10 kHz to 100 kHz, therefore able to provide delays from 20.48 ms to 204.8 ms. We did not attempt at simulating any of the characteristics of the chip (noise, etc), to avoid interfering with our analysis on the antialiasing properties of the proposed method. From now on, we will refer to the SNR only as the ratio between the input signal and all the aliasing components present in the output signal (both BGA and SGA because they can hardly be separated). All experiments have been run with a discrete-time simulation sampling rate of $F_s = 44100\,\text{Hz}$. The polyBLEP residual is computed according to the third-order Lagrange polynomials from [14] (Table III).

4.1. Static Clock Rate

The first experiments examined the behavior of the system without any modulations, i.e. at a fixed F_{CLK} and a constant sine frequency F_0. To highlight any spurious component we used a constant-frequency sine wave. The trivial implementation of the BBD device is compared to the BLEP version at three different clock rates. Results are shown in Figure 2. The SNR is detailed in the captions.

As can be seen, the SGA components are largely reduced by the use of BLEP. For a F_{CLK} of 50 kHz and 90 kHz the SNR improves by 36.2 and 27.8 dB, respectively. The data related to the lower F_{CLK} requires explanation. As discussed previously, real BBD chips can naturally exhibit aliasing at their output since they perform a discrete-time sampling. This behavior, in the context of our simulations is desired, since we aim at reproducing these characteristics of the chip. From Figure 2(a-b) it can be noticed that the spurious components highlighted with the red arrows, i.e. those at $k \cdot F_{CLK} \pm F_0, (k = 1, 2...)$ are not affected by applying the BLEP, as desired. This behavior is exhibited only when the BBD runs at $F_{CLK} < F_s$. The presence of those spurious components in the BLEP BBD output makes the SNR reduction very low (from 16.9 dB to 19.1 dB) but if the SNR estimate is corrected by removing those components from computation, the SNR for the trivial and BLEP cases are 23.2 dB and 48.0 dB, respectively. This shows the great potential for BLEP in BBD simulation.

4.2. Input Interpolation

To verify whether the input interpolation is an important factor in reducing spurious components, we conducted an additional experiment using a higher order interpolation algorithm. Since linear interpolation corresponds to a 1st order Lagrange interpolation, we decided to employ a higher-order Lagrange interpolation scheme [15], using 4 points, which constitutes a notable improvement with respect to linear interpolation while requiring a moderately higher computational effort. Figure 2 also shows, on the rightmost column, the spectra of the BBD output where BLEP is applied in conjunction with Lagrange interpolation. As can be seen, an improvement of the SNR is obtained in some cases. Specifically, at 50 kHz, an increase of the SNR by 15.4 dB is achieved. Some spurious components are reduced also in the 10 kHz case, between the fundamental and the first aliasing tone (first red arrow). The SNR shown in the figure caption is not affected by this small reduction since it includes the BGA. If correcting the SNR by removing these aliasing components, as done in Section 4.1, the SNR improves by 0.5 dB. Finally, no SNR improvement is seen in the 90 kHz case. The differences between these cases lies in the fact that some spurious components are generated by the modulation of the interpolation filter coefficients, while others are generated by the resampling, therefore only the former can be effectively reduced.

We can conclude that higher-order interpolation can slightly decrease the amount of spurious components, but its perceptual important is probably negligible. The reader can evaluate the results at the online companion page.

4.3. Time-Varying Clock Rate

To show the ability of the method to deal with time-varying F_{CLK}, which is an essential feature in flanger and chorus effects, we conducted further tests. The clock rate has been modulated using a sinusoidal oscillator $m(t)$ at 1 Hz, which modulates the BBD clock rate according to the following

$$\tilde{F}_{CLK}(t) = F_{CLK} \cdot (1 + 0.2 \cdot m(t)). \tag{2}$$

The results are shown in Figure 3 in form of spectrograms. A completely alias-free modulated delay would only show the modulated sine pointed by the blue arrow. A BBD introduces BGA components when $F_{CLK} < F_s$, indicated by the red arrow. All other aliasing components are due to the discrete-time simulation. In the trivial case, these are so widespread that in addition to thick modulated lines, the spectrograms show a noise-like background texture. On the other hand, the BLEP method is capable of reducing the aliasing significantly. Please note that the spectrograms have been computed using a minimum threshold of -100 dB.

5. CONCLUSIONS

This paper proposed the discrete-time emulation of BBD using BLEP as a method to reduce the aliasing generated by the discontinuities at the BBD output. This is related to the fact that the simulation is conducted at a sampling rate F_s and has nothing to do with the aliasing generated by the process of sampling at the BBD clock rate F_{CLK}, which is a desired artifact typical of real BBD chips.

The proposed method has been tested in a simulated environment without introducing other characteristics that are typical of BBD chips, such as noise, nonlinearities, and more. This allowed to assess the validity of the method, showing that BLEP can greatly reduce the aliasing. Even larger improvement can be achieved with a higher Lagrange or B-spline order. The importance of input interpolation has been also the object of experiments, showing that it can be partially useful to reduce spurious components, but it has a minor impact in the overall SNR.

The technique is flexible, allowing the degree of aliasing reduction and the computational cost to scale with the number of BLEP steps N_B considered, therefore trade-offs can be evaluated in practical scenarios.

Example code and audio are released by the authors in the online companion page for further dissemination within the community.

As a future work, formal subjective listening tests could be useful to assess the best trade-off in terms of audio quality between the available settings (N_B, interpolation accuracy, etc.).

6. REFERENCES

[1] Udo Zölzer, *DAFX-Digital audio effects, Second Edition*, John Wiley & Sons, 2011.

[2] Vadim Zavalishin and Julian Parker, “Efficient emulation of tapelike delay modulation behavior,” in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18)*, Aveiro, Portugal, 2018.

[3] Jatin Chowdhury, “Real-time physical modelling for analog tape machines,” in *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19)*, 2019, p. 7.

[4] Jon Dattorro, “Effect design, part 2: Delay line modulation and chorus,” *Journal of the Audio engineering Society*, vol. 45, no. 10, pp. 764–788, 1997.

[5] Vesa Välimäki, Julian Parker, Lauri Savioja, Julius O Smith, and Jonathan Abel, “More than 50 years of artificial reverberation,” in *60th International Conference of the Audio Engineering Society*. Audio Engineering Society, 2016.

[6] Antti Huovilainen, “Enhanced digital models for analog modulation effects,” in *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx-05)*, Madrid, Spain, pp. 155–160.

[7] Colin Raffel and Julius Smith, “Practical Modeling of Bucket-Brigade Device Circuits,” in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, 2010.

[8] Jaromír Mačák, “Simulation of Analog Flanger Effect Using BBD Circuit,” in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016.

[9] Davide Rocchesso, “Fractionally addressed delay lines,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 717–727, 2000.

[10] Martin Holters and Julian Parker, “A Combined Model for a Bucket Brigade Device and its Input and Output Filters,” in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18)*, Aveiro, Portugal, 2018.

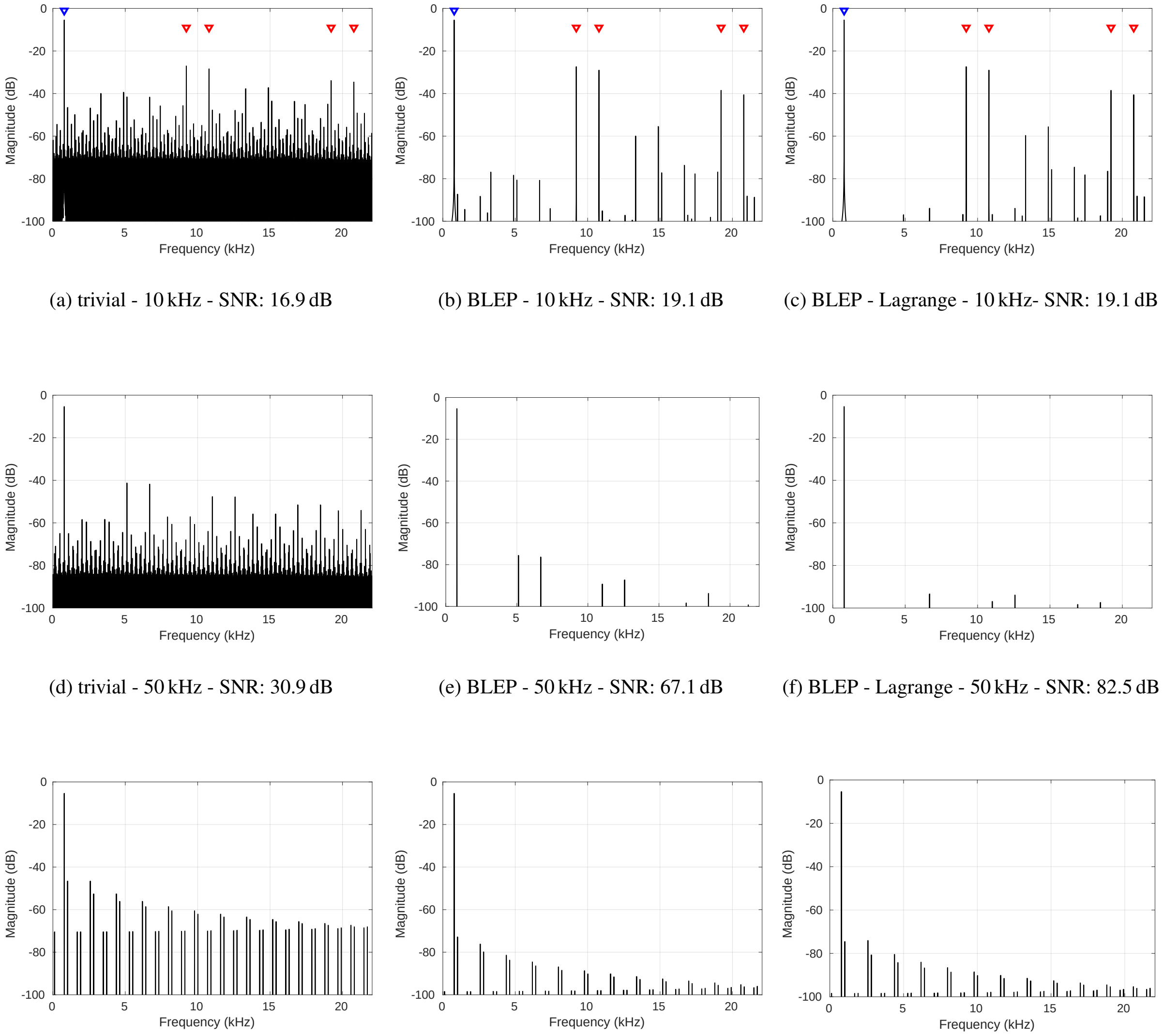
[11] Eli Brandt, “Hard sync without aliasing,” in *Proceedings of the 2001 International Computer Music Conference, ICMC 2001, Havana, Cuba, September 17-22, 2001*. 2001, Michigan Publishing.

[12] Fabián Esqueda, Stefan Bilbao, and Vesa Välimäki, “Rounding corners with blamp,” in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-19)*. Brno University of Technology, 2016.

[13] Vesa Välimäki and Antti Huovilainen, “Antialiasing oscillators in subtractive synthesis,” *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 116–125, 2007.

[14] Vesa Välimäki, Jussi Pekonen, and Juhan Nam, "Perceptually informed synthesis of bandlimited classical waveforms using integrated polynomial interpolation," *The Journal of the Acoustical Society of America*, vol. 131, no. 1, pp. 974–986, 2012.

[15] Timo I Laakso, Vesa Välimäki, Matti Karjalainen, and Unto K Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30–60, 1996.



(a) trivial - 10 kHz - SNR: 16.9 dB

(b) BLEP - 10 kHz - SNR: 19.1 dB

(c) BLEP - Lagrange - 10 kHz- SNR: 19.1 dB

(d) trivial - 50 kHz - SNR: 30.9 dB

(e) BLEP - 50 kHz - SNR: 67.1 dB

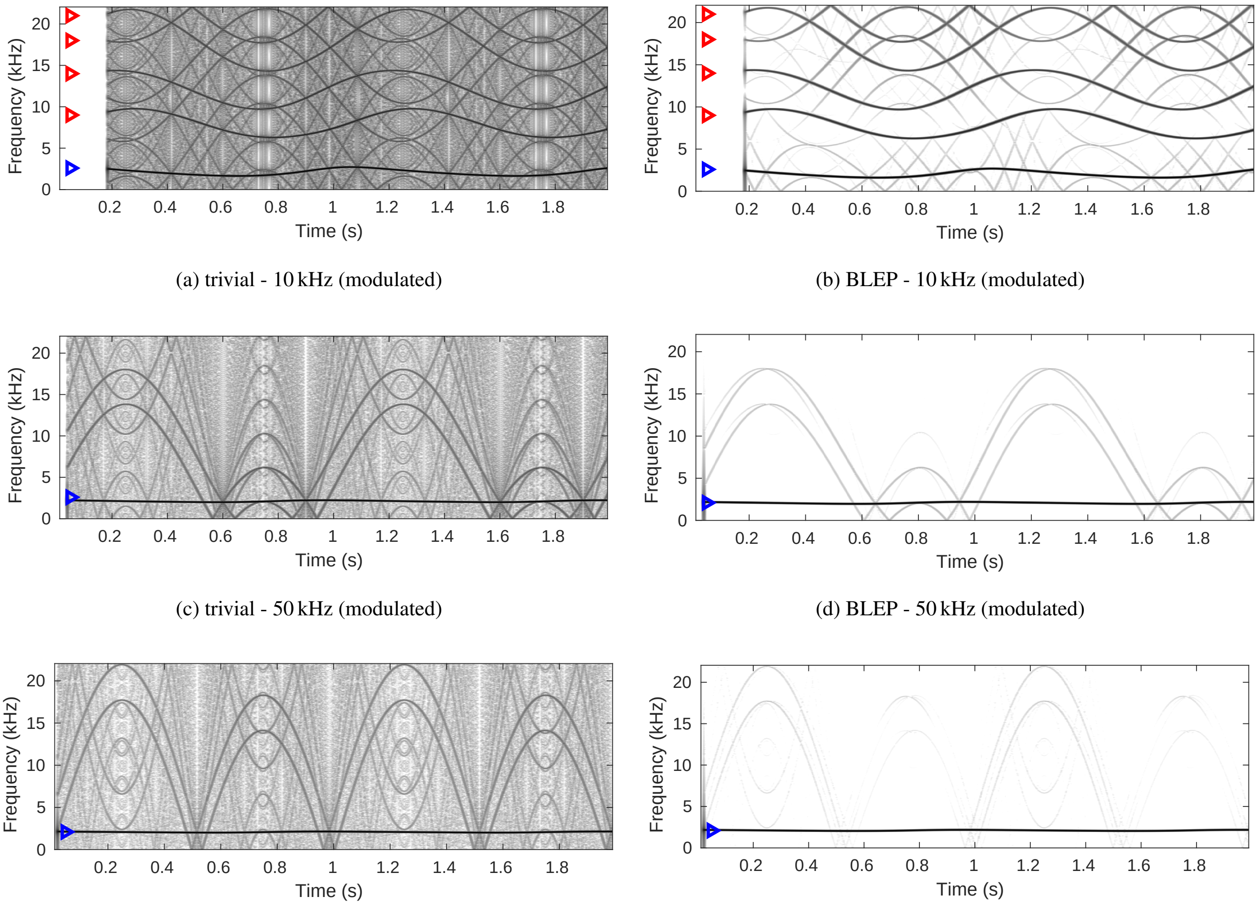
(f) BLEP - Lagrange - 50 kHz - SNR: 82.5 dB

(g) trivial - 90 kHz - SNR: 36 dB

(h) BLEP - 90 kHz - SNR: 63.8 dB

(i) BLEP - Lagrange - 90 kHz - SNR: 63.8 dB

Figure 2: Spectra of the BBD outputs for a $F_0 = 783.99$ Hz sine tone input (G5) at various F_{CLK}, using the trivial method (left column), using BLEP (center column), using BLEP and 4-point Lagrange interpolation (right column). Please note that with $F_{CLK} < F_s$, BGA components (highlighted by the red arrows) are present and are, thus, desired for a correct simulation of the component.



(a) trivial - 10 kHz (modulated)

(b) BLEP - 10 kHz (modulated)

(c) trivial - 50 kHz (modulated)

(d) BLEP - 50 kHz (modulated)

(e) trivial - 100 kHz (modulated)

(f) BLEP - 100 kHz (modulated)

Figure 3: Spectrograms showing the output of the BBD simulations using an input sine at 2093 Hz (C7 note) and the F_{CLK} modulated by the sine oscillator. The modulated sine is pointed by the blue arrow. The red arrows point to the proper aliasing components expected in a BBD. The initial delay (varying from case to case) is evident in the spectrogram as the first portion of the output is silent. Full scale is -100 dB.