

Virtual Analog: Bandlimited Discontinuities with Infinite Response

TOM SCHOUTEN

July 24, 2003

Abstract

In this paper we discuss the usage of a bank of damped oscillators to synthesize bandlimited discontinuities. The method is the IIR equivalent of earlier methods using FIR filters and parametric synthesis.

1 Introduction

This paper discusses the synthesis of digital bandlimited signals with simple discontinuities (pulse, sawtooth, square, synced sawtooth, pwm). It is a continuation of the work presented in [1] and [2]. The contribution of this paper is the use of the sampled response of a continuous time IIR filter instead of using direct synthesis¹ (additive or through the discrete summation formula) or wavetable playback (oversampled lowpass FIR filter approach). Since this approach is purely synthetic it can be parametrized to provide an easy quality vs. execution speed tradeoff at runtime. At the cost of redesigning the IIR filter. One more tool for the toolbox...

We will use the following notational conventions and abbreviations. Lower case letters represent time sequences or functions. Upper case letters represent their spectral representation after applying the z-transform or Laplace transform.

A discrete time (DT) sequence is denoted by $h[n]$ with $n \in \mathbb{Z}$. Its z-transform is defined by

¹It could be argued that this method is in fact a direct synthesis method since the discrete IIR filters used in the implementation are basically used as oscillators. But this is a detail of course.

$H(z) = \mathcal{Z}\{h[n]\}(z) = \sum_{n \in \mathbb{Z}} h[n]z^{-n}$. The inverse z-transform is denoted by $h[n] = \mathcal{Z}^{-1}\{H(z)\}[n]$.

A continuous time (CT) sequence is denoted by $h(t)$ with $t \in \mathbb{R}$. Its Laplace transform (or s-transform) is defined by $H(s) = \mathcal{L}\{h(t)\}(s) = \int_{t \in \mathbb{R}} e^{-st} h(t) dt$. The inverse Laplace transform is denoted by $h(t) = \mathcal{Z}^{-1}\{H(s)\}(t)$.

2 Algorithm

The algorithm consists of three parts:

1. *Setup*: Design the lowpass filter and obtain its partial fraction expansion to parametrize the oscillator bank.
2. *Synthesis*: “Running” the filter by implementing a damped oscillator bank using the poles and gains from the partial fraction expansion. (The homogenous solution).
3. *State Update*: At each sample point, compute the state update for each oscillator depending on the continuous time input. (The particular solution).

2.1 Bandlimited Discontinuity Synthesis

Our main goal is to create signals containing discontinuities (impulses, steps, ramps, ...). Doing this using a DT method is non-trivial, since naive sampling will introduce aliasing. In theory, a simple method would consist of creating a CT impulse train, square wave,

triangle wave,... and filter it using a CT lowpass filter before sampling, to prevent aliasing. We will come back to the specific anti-aliasing requirements of several waveforms later.

We will construct the IIR variant of the approaches in [2] by directly synthesizing the sampled output of a CT lowpass filter driven with a CT signal containing wideband discontinuities. We will do this by manipulating the state of a bank of exponentially decaying oscillators. The state updates are determined by the theoretical CT input signal. For simple waveforms containing constant and linear continuous terms and impulse, step and ramp discontinuities this approach is analytically tractable.

Suppose $H(s)$ is the transfer function of a CT lowpass filter. Using the partial fraction (parallel) decomposition of the CT filter's transfer function

$$H(s) = \sum_{i=0}^{N-1} \frac{a_i}{s - \lambda_i}, \quad (1)$$

we can simplify the problem further. Computing the response of $H(s)$ to some signal can be split up into computing the response of complex one-pole filters represented by $H_i(s) = \frac{a_i}{s - \lambda_i}$ and summing their output. We can safely ignore the details of $H(s)$ as long as (1) exists. We will investigate the response of the complex one-pole filter to several kinds of inputs, and determine a way to synthesize them in discrete time.

The important thing to note here is that the location of the pulse or other discontinuity that will excite the CT first order system does not necessarily need to coincide with a sampling point. In fact the phase contribution of the input at time step $n \in \mathbb{Z}$ is computed out of the behaviour of the CT input waveform from time $n - 1$ to n , by solving the first order differential equation for different kinds of "stretched pulses". We then later reconstruct the input waveform by using these stretched pulses as base functions. The aliasing that will result by using this method will be completely defined by $H(s)$. By driving it with a wideband, pulse like signal, we will effectively sample its spectrum. If it has a significant contribution above Nyquist, noticeable aliasing will occur.

Due to linearity of the filter, the tail of the previous impulse or other discontinuity will still be present in the filter when we add a new phase update, so we don't need to increase the number of voices whenever the frequency rises, like in the FIR methods described in [1] and [2]. The number of voices (damped oscillators) only affects the quality of the bandlimited impulse generator, since a higher order filter (more oscillators) will enable us to have a higher bandwidth for the same aliasing level, or a lower aliasing level for the same bandwidth.

2.2 Lowpass Filter

Representing the effect of an impulse as a state update of a DT filter requires us to use a strictly proper transfer function for the IIR filter, i.e. it should be "integrating" or in other words not contain a direct input-output path. This is equivalent to saying that the impulse invariant transform can't always be used to convert CT filters to DT ones. This is not a problem since we are designing a lowpass filter. The partial fraction expansion (1) will contain only strictly proper terms and no direct path.

Instead of using a parallel set of biquads to synthesize the sum of damped sinusoids, we could also use a series configuration. However, this does complicate the matter because we need to take the zeros of the transfer function into account. The problem is no longer symmetric, i.e. each complex one-pole or two-pole filter needs to be treated different.

Ignoring the zero problem and assuming we have an all-pole filter without real poles each CT 2-nd order section will be "doubly integrating". This means when the first will receive a CT impulse, the second will receive a CT ramp, etc.. This does limit the number of contributions that need to be made since the series can be cut off at some point where the contributions are small enough (probably after the first stage since the ramp update is significantly smaller than the impulse update).

When the transfer function has zeros, (maximally one less the number of poles: it has to be strictly proper because of the lowpass characteristic), we can limit the updates to an impulse update in the first stage and step updates in the other (assuming the

maximum number of zeros).

Although the series approach has some advantages, we will not pursue this route. Sticking with the parallel implementation greatly simplifies the algorithm. Another advantage is that the oscillator bank could be used for other purposes, like additive or modal synthesis.

The oscillator bank could even be used to do a time/frequency tradeoff. When synthesizing a high frequency bandlimited waveform, there is a trade-off point where the waveform is more efficiently synthesized as a sum of non-damped sinusoids, instead of time shifted bandlimited discontinuities [1]. Since aliasing is mostly noticeable for the higher frequencies², using this fact could lower the requirements of the lowpass filter.

The choice of IIR filter design method is not limited by the parallel algorithm, as long as the filter poles are below the Nyquist frequency. This is not a problem since the filter needs to block all frequencies above Nyquist to prevent aliasing, so it will need all its poles in the passband below Nyquist.

2.3 The CT System

We will compute the state updates by working with a CT differential equation. We start from a simple building block, the CT system corresponding to the transfer function

$$P(s) = \frac{-\lambda}{s - \lambda}, \quad (2)$$

with $\lambda \in \mathbb{C}$. It is defined by the first order differential equation

$$\frac{d}{dt}y(t) = \lambda(y(t) - x(t)). \quad (3)$$

We chose this DC-normalized system to simplify the expressions for the step and ramp updates. When we integrate this equation to sample it at discrete time steps $n \in \mathbb{Z}$, this leads to the update equation

²This is assuming constant lowpass filter roll-off and/or saw or square waveforms with decaying spectrum, instead of impulse waveforms with a constant magnitude spectrum.

$$y[n] - y[n - 1] = \lambda \left(\int_{n-1}^n y(t) dt - \int_{n-1}^n x(t) dt \right). \quad (4)$$

The free system (with $x(t) = 0$) has a solution $y(t) = \alpha e^{\lambda t}$. In update form this will be $y(t_1) = y(t_0)e^{\lambda(t_1-t_0)}$ or

$$y[n] = y[n - 1]e^{\lambda} \quad (5)$$

When the input is non-zero, we will add a correction term C depending on the kind of input the system has had between the current n and previous $n - 1$ time step:

$$y[n] = y[n - 1]e^{\lambda} + C \quad (6)$$

2.4 Continuous State Update

Let's have a look at the continuous constant and ramp inputs we need to implement the full set of waveforms through state updates. Later we will have a look at the contributions for discontinuous inputs (impulse and step).

Because of the unit DC-gain, the constant unity input $x(t) = 1$ has the solution $y(t) = \alpha e^{\lambda t} + 1$. Writing this in update form from t_0 to t_1 gives us $y(t_1) = (y(t_0) - 1)e^{\lambda(t_1-t_0)} + 1$ or

$$y[n] = y[n - 1]e^{\lambda} + \underbrace{1 - e^{\lambda}}_{C_{cc}(\lambda)} \quad (7)$$

We compute the ramp update in the same way. We will concentrate on a unit slope ramp crossing the x axis at 0. The constant part can be covered by (7). When $x(t) = t$ we have the general solution $y(t) = \alpha e^{\lambda t} + t - \frac{1}{\lambda}$. Writing this as an update equation gives us $y(t_1) = (y(t_0) - t_0 + \frac{1}{\lambda})e^{\lambda(t_1-t_0)} + t_1 - t_0$. Shifting the input to $x(t) = t - (n_0 - 1)$ gives us

$$y[n] = (y[n - 1] + \frac{1}{\lambda})e^{\lambda} + 1 \quad (8)$$

Isolating the input update from the “ringing” update gives

$$y[n] = y[n-1]e^\lambda + \underbrace{\frac{e^\lambda}{\lambda} + 1}_{C_{cr}(\lambda)} \quad (9)$$

The correction equation for a generic piecewise linear input defined by the end points $x(n-1) = a$ and $x(n) = b$ is given by a combination of the continuous constant update and the continuous ramp update. $C_{pwl}(a, b, \lambda) = aC_{cc}(\lambda) + (b-a)C_{cr}(\lambda)$ or

$$C_{pwl}(a, b, \lambda) = b - e^\lambda \left(a + \frac{b-a}{\lambda} \right). \quad (10)$$

2.5 Discontinuous State Update

The state update method can be generalized to wide-band discontinuities.

In the case of a unit area impulse at $t = n_0 - t_0$ with $0 \leq t_0 < 1$ we will have 3 contributions: free ringing before and after the impulse, and a direct state contribution of $-\lambda$ at $t = -t_0$.

Splitting the contribution in 3 parts gives us $y^-(n_0 - t_0) = y(n_0 - 1)e^{\lambda(1-t_0)}$, $y^+(n_0 - t_0) = y^-(n_0 - t_0) - \lambda$ and $y(n_0) = y^+(n_0 - t_0)e^{\lambda t_0}$. This leads to the discrete update equation at time n_0

$$y[n_0] = y[n_0 - 1]e^\lambda + \underbrace{(-1)\lambda e^{\lambda t_0}}_{C_{di}(\lambda, t_0)} \quad (11)$$

In the case of a unit step function at $t = n_0 - t_0$ with $0 \leq t_0 < 1$ we can again split up the contribution in two parts: $y(n_0 - t_0) = y(n_0 - 1)e^{\lambda(1-t_0)}$ and $y(n_0) = 1 + (y(n_0 - t_0) - 1)e^{\lambda t_0}$. This leads to the discrete update equation at time n_0

$$y[n_0] = y[n_0 - 1]e^\lambda + \underbrace{1 - e^{\lambda t_0}}_{C_{ds}(\lambda, t_0)} \quad (12)$$

We can go further and compute phase corrections for the higher order discontinuities in terms of $(t-t_0)^k$ with $k \geq 1$. However, the phase correction terms will grow smaller and smaller. This raises the question if it's still worth adding these corrections for synthesis of bandlimited waveforms, since only the sharp transients produce noticeable aliasing artifacts. A more philosophical argument could be that the feature that

makes waveforms with discontinuities interesting, is their sharpness, so synthesising smooth discontinuities is not as musically interesting as non-smooth ones.

2.6 Putting Everything Together

Constructing the impulse response of a waveform built from the elementary state updates found above can be done in the following way. We were using the system defined by (2) and computed the state updates for the discrete system (5) that synthesizes the sampled response of (2). The scaling $A_i \in \mathbb{C}$ we need to apply when adding state updates C_x to the oscillator state y_i is determined by the modified partial fraction expansion

$$H(s) = \sum_{i=0}^{N-1} A_i \frac{-\lambda_i}{s - \lambda}. \quad (13)$$

The a_i and A_i are thus related by $A_i = \frac{-a_i}{\lambda_i}$, due to the (somewhat arbitrary) DC-normalization we use in (2). Not having this normalization will of course propagate the λ_i scaling to the C_x phase contributions.

Because the filters are linear, we have the tools to construct all kinds of theoretical input waveforms built by adding their respective C_x after the phase update (5) has occurred.

We've worked in the complex field for simplicity while constructing our algorithm. The partial fraction expansion has complex conjugate terms: For each $H_i(s)$ there is a $H_j(s) = \overline{H_i(\overline{s})}$ with a complex conjugate impulse response.

When driving the one-pole filters with a real valued input, like we do, the conjugate one-poles filters will produce complex conjugate responses. We can calculate one of the complex conjugate oscillators, take the real part and scale by 2 to get the same result.

3 Implementation

The only real roadblock left is the computation of the complex exponential involved in the expression for the various state updates which contain the terms

$z_i^{t_0}$ or $e^{\lambda_i t_0}$. For poles close to 1 we can probably get away with a linear or quadratic truncated power series.

Since the cost of the updates is directly proportional to the frequency of the waveform to be synthesized, this is another argument for implementing a hybrid method.

Determining the necessary order of the filter $H(s)$ depends on the quality requirements and the type of waveform that will be synthesized. In general this is hard to do in advance. Since increasing the order of the filter requires redesigning it, which for standard IIR filters is not a very complex operation, we advocate the use of a “quality button”, to allow the user to tune the quality of the oscillator depending on the needs.

We will now have a closer look to some classic waveforms and investigate the trade-offs.

3.1 Impulse Train

This is the “richest” of all waveforms and is therefore most subject to aliasing. Using the method described above will produce the sampled response function $y[n] = y(n)$, with $y(t) = \mathcal{L}^{-1}\{Y(s)\}(t)$. The input response $Y(s) = H(s)X(s)$ is equal to the product of the filter and input spectrum. The input spectrum $X(s)$ is an impulse train because $X(s) = \mathcal{L}\{x(t)\}(s)$ and $x(t) = \sum_i \delta(t - it_0)$ is an impulse train.

Because of the broad spectral nature of $X(s)$ the contribution of the spectral content above Nyquist in $H(s)$ to the sampled input response $y[n]$ will be significant, so suppressing this aliasing will require a high order lowpass or a lower bandwidth. This makes the band limited impulse train the most expensive waveform to synthesize. The algorithm is simple though. We only need to add a state update at each discrete time step n for which there was an impulse in the time between $t = n - 1$ and $t = n$.

The update for an impulse discontinuity is (11) scaled by the A_i from (13). The lambda factor cancels out so the contribution is $a_i e^{\lambda t_0}$ with a_i from (1).

3.2 Sawtooth

The sawtooth wave is less expensive because of its lowpass-like frequency spectrum. $x_{\text{saw}}(t) = \int (x_{\text{pulse}}(t) - \frac{1}{t_0}) dt$ so $X_{\text{saw}}(s)$ decays like $1/s$. In other words, frequencies above nyquist in $H(s)$ will be excited less than is the case for an impulse train, when t_0 is large, and the aliasing is less audible due to the masking effect of the larger lower frequency harmonics. This reduces the constraints on the order of $H(s)$.

3.3 Square wave

The same reasoning goes for the square wave, which has a similar spectrum. Square waves can of course be generated using a combination of two sawtooth oscillators. However since direct synthesis requires only adding a constant and a position depended update during the discontinuity, it might be more appropriate to directly synthesize the square wave. The same goes for PWM.

3.4 Pulse width modulation (PWM)

This can again be synthesized using sawtooth waves, however the pulse like nature will be more broad band than a pure sawtooth wave, so there are arguments for increasing the order of $H(s)$ to synthesize both sawtooth waves. A counter argument could be that pulselike transition during (deep) pulse width modulation has very little energy compared to the full square wave regime, so when the transition “trough” the pulse is fast, a lot of aliasing will be masked out. So this is a tricky one since it involves a lot of psychoacoustic knowledge to predict the perception of aliasing. Experimental tuning will probably be the best way to find the trade-off.

When synthesizing a static pulse wave with variable pulse width, a high order is probably necessary, since in the limit with duty cycle approaching zero we will converge to the “pure” impulse train, so aliasing will be more perceivable. This is even more important if the wave is DC corrected and power compensated.

3.5 Hard synced sawtooth

For this very rich waveform the same arguments hold as for PWM and impulse train. The range of attainable spectra contains very broad ones so a high order $H(s)$ is probably necessary.

4 Results

5 Conclusion

Definite eargasm.