Victor Lazzarini and Joseph Timoney
Sound and Digital Music Research Group
Maynooth University
Co. Kildare Ireland
{victor.lazzarini, joseph.timoney}@mu.ie

Linear and Nonlinear Digital Filters: From the Analog and Beyond

Abstract: A common approach in the development of digital filters is to begin with an existing analog filter and produce an equivalent computer program to realize it. This may involve, at the extreme, the detailed analysis of circuit behavior, or it may stem from a higher-level approach that looks at block diagrams and *s*-domain transfer functions. In this article, we first take the latter approach to develop a set of linear filters from the well-known state variable filter. From this we obtain a first result, which is a linear digital implementation of the Steiner design, comprising separate inputs for different frequency responses and a single output summing the responses. Turning back to the state variable design, we show that to develop a nonlinear version, an analog circuit realization can be used to identify positions in which to insert nonlinear waveshapers. This gives us our second result, a nonlinear digital state variable filter. From this analog-derived design, we then propose modifications that go beyond the original filter, developing as a final result a structure that could be classed as a hybrid of filter and digital waveshaper. As part of this process, we ask the question of whether an approach that takes inspiration from the analog world, while being decoupled from it, may be more profitable in the long run than an obsession with detailed circuit modeling.

Digital filters are traditionally designed using one of two routes. If we are interested in a fixed-parameter filter with linear phase characteristics, the standard method is to define an amplitude response from which a filter impulse response that approximates it with a defined level of precision can be obtained by an inverse discrete Fourier transform (Lazzarini 2021). Although this may be optimal for engineering applications, it is too constrained to be of use in general musical applications, as for instance in the development of computer music instruments, since dynamic spectra are fundamental in these scenarios. Therefore, if we are looking for a flexible variable-parameter filter that can be used for such applications, the approach is to look for what has been described as a *canned* filter design (Steiglitz 1996), that is, one that has been worked out in advance, fully parametrized, giving a specific amplitude response. Since Mathews's MUSIC IV (Mathews and Miller 1964), the ubiquitous *reson* two-pole filter has been a classic example of this approach. Through the application of the various techniques of discretization of continuous-time systems, such as the Euler method (Oppenheim et al. 1999) and the bilinear transformation (Steiglitz 1965), models for musical filters for computer music could be sought in the various existing forms of analog filters. This in fact eventually developed into a complete subset of digital signal processing (DSP) called virtual analog (Pakarinen et al. 2011), which encompassed other, newer techniques that also stretched beyond filter circuits into the modeling of other components such as amplifiers (Pakarinen and Yeh 2009), oscillators (Timoney et al. 2008), ring modulators (Parker 2011), and even hybrid systems such as bucket-brigade delays (Raffel 2010).

Computer Music Journal, 45:2, pp. 67–83, Summer 2021
doi:10.1162/COMJ_a_00599

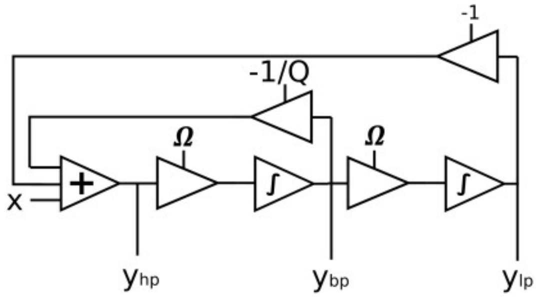
Although this appears to be an interesting way to proceed, and much profit can be taken from it, it is possible to wonder whether we may look to move forward, rather than backward, in the design of digital musical instruments. Much of the interest driving the search for ever-more-accurate reproductions of actual circuits, and even of electromechanical devices (Gabrielli et al. 2013), is of course motivated by the search for commercially successful products in the small, one might say tiny, and crowded market place of the music instrument and software industry. It is possible, however, to exploit DSP in ways that make use of its precision to pursue designs that are beyond any analog circuit and perhaps redirect the interest towards innovative computer instruments. This is of course nothing new: Frequency modulation synthesis (Chowning 1973) is a classic case from over fifty years ago in which we saw something completely innovative that required a precision of computation beyond anything that a voltage-controlled system could deliver. One could also argue that the additive synthesis routines of Jean-Claude Risset (2009) proved the same point even earlier.

In this article, we review a well-known analog design, or what Hutchins (1999) calls a *configuration*: the state variable filter, its linear digital models, and some nonlinear extensions that may be classified under the category "beyond the analog." We will proceed by briefly reviewing some key ideas about analog filters and their use in musical instruments. We will follow this by concentrating on our target, exploring its digital forms, and then introducing some modifications that are originally inspired by the nonlinear behavior of analog circuit components, but that can also be extended to the user's imagination. We will finally discuss these results and other approaches to the development of filters that are purely digital creations, which illustrate the main point of this article, that the scope of a computer musician's tools does not necessarily need to be curtailed by the imitation of classic sounds of analog synthesizers.

Analog Filters: Linear Models

Although analog filter circuits may have various forms, musical applications typically favor the use of voltage-control configurations, which can provide the required dynamic spectra conditions. With a few exceptions, this category of filters is normally represented by two designs (Hutchins 2013): (1) the fourth-order low-pass filter; and (2) the second-order state variable filter (SVF). The first of these was the invention of Robert Moog and has been implemented in various forms: from the original, using a transistor ladder (Moog 1965; Stilson and Smith 1996; Huovilainen 2004; Fontana 2007); to the use of diodes (Fontana and Civolani 2010) and operational transconductance amplifiers (OTAs) as gain elements, in both discrete (Hutchins 1978) and integrated (Wiltshire 2008) circuit forms. The topic of the ladder filter, in particular, has been covered extensively in the literature and has received perhaps a definitive treatment by D'Angelo and Välimäki (2014a, 2014b). In this article, we will be focusing on the second of these two filters, the SVF model (Chamberlin 1985), which is a highly versatile filter. It was in fact shown early on that it can be used to implement an equivalent filter to the four-pole low-pass filter (Hutchins 1982), something that was demonstrated again recently, almost forty years later, by Werner and McClellan (2020).

Figure 1. State variable filter (SVF) block diagram: a biquadratic, second-order filter.



The State Variable Filter

To our knowledge, the principles behind the SVF first appeared in the relevant literature through the work of Kerwin et al. (1967). A modular-synthesizer implementation utilizing voltage control (Colin 1971) soon followed. The fundamental elements of the design are an adder, two integrators, and feedback. A block diagram of the filter is shown in Figure 1, and the filter is capable of producing, simultaneously, high-, band-, and low-pass responses. By adding the low- and high-pass signals together, it is also possible to produce a notch response, and an equal mix of the three responses produces an all-pass output.

There are effectively only two control parameters in this design, which makes it particularly straightforward in musical applications. The first of these, Ω, is proportional to the cutoff or center frequency, whereas the second, Q, controls the peaking behavior of the filter, or its resonance. From an analysis of its block diagram, we can derive a system of equations for the filter,

$$\begin{aligned} y_{hp}(t) &= x(t) - (1/Q)y_{bp}(t) - y_{lp}(t) \\ y_{bp}(t) &= \Omega \int y_{hp}(t)dt \\ y_{lp}(t) &= \Omega \int y_{bp}(t)dt. \end{aligned} \quad (1)$$

From these and the integrator transfer function $H(s) = 1/s$, we can derive a basic set of filter frequency responses as (Hutchins 2000b):

high-pass output:

$$H_{hp}(s) = \frac{s^2}{s^2 + (\Omega/Q)s + \Omega^2}; \tag{2}$$

band-pass output:

$$H_{bp}(s) = \frac{\Omega s}{s^2 + (\Omega/Q)s + \Omega^2}; \tag{3}$$

and low-pass output:

$$H_{lp}(s) = \frac{\Omega^2}{s^2 + (\Omega/Q)s + \Omega^2}. \tag{4}$$

As shown by these equations, the SVF has a biquadratic form, which may be written as

$$H_{svf}(s) = \frac{a_h s^2 + a_b \Omega s + a_l \Omega^2}{s^2 + (1/Q)s + \Omega^2}, \tag{5}$$

where a_h, a_b, and a_l are scaling factors that determine the amplitude response to be high, band, or low pass.

Linear Digital Models

We can now look at the digital models of the state variable filter, without taking into account possible nonlinearities caused by any particular choice of components and signal levels in actual circuit implementations of the block diagram shown in Figure 1. For this, we only need to decide how the discretization of the timing elements in the filter will be carried out. The problem narrows down to implementing the two integrators in the circuit and deciding where to place the required one-sample delay that will arise in any digital feedback filter. A first pass at this yields a direct implementation of the block diagram, which is equivalent to using a Euler method to discretize the integrators,

$$\begin{aligned} y_{hp}(n) &= x(n) - (1/Q)y_{bp}(n-1) - y_{lp}(n-1) \\ y_{bp}(n) &= \Omega y_{hp}(n) + y_{bp}(n-1) \\ y_{lp}(n) &= \Omega y_{bp}(n) + y_{lp}(n-1). \end{aligned} \tag{6}$$

This form of the filter is similar to that developed by Chamberlin (1985), except for the ordering of the computation, which in his case puts the low- and high-pass outputs within the same cycle and the band-pass output in the following cycle. Chamberlin's approach was motivated by the fact that it can produce a notch output directly by mixing the low- and high-pass signals. To obtain this, in the present version, we would need to split the high-pass computation into two stages, first producing a notch output by subtracting the band-pass feedback from the input signal, and then producing the high-pass signal as the sum of this result and the negative low-pass feedback. Nevertheless, such differences are not significant when we look at the filter transfer functions.

A similar method to the one applied to derive the frequency responses in the s-domain can be applied here, in discrete time. The outline of the high- H_{hp}, band- H_{bp}, and low-pass H_{lp} transfer functions can be given as

$$\begin{aligned} H_{hp}(z) &= 1 - \frac{z^{-1}}{Q} H_{bp}(z) - z^{-1} H_{lp}(z) \\ H_{bp}(z) &= \frac{\Omega H_{hp}(z)}{1 - z^{-1}} \\ H_{lp}(z) &= \frac{\Omega H_{bp}(z)}{1 - z^{-1}} = \frac{\Omega^2 H_{hp}(z)}{(1 - z^{-1})^2}, \end{aligned} \tag{7}$$

with the integrator transfer function given as $H(z) = 1/(1 - z^1)$ and a z^{-1} factor to take account of the one-sample delay in the feedback path.

From these, we obtain the correct frequency response for the high-pass output

$$H_{hp}(z) = \frac{(1 - z^{-1})^2}{1 - (2 - \Omega/Q - \Omega^2)z^{-1} + (1 - \Omega/Q)z^{-2}}. \tag{8}$$

From $H_{hp}(z)$ we get

$$H_{bp}(z) = \frac{\Omega(1 - z^{-1})}{1 - (2 - \Omega/Q - \Omega^2)z^{-1} + (1 - \Omega/Q)z^{-2}}. \tag{9}$$

Finally,

$$H_{lp}(z) = \frac{\Omega^2}{1 - (2 - \Omega/Q - \Omega^2)z^{-1} + (1 - \Omega/Q)z^{-2}}. \tag{10}$$

Although absolutely correct, these digital models suffer from the finitude of the sampling rate f_s. In particular, the filter starts to become unstable at high frequencies, if these are beyond about $f_s/4$. Dattorro (1997), in his study of the Chamberlin filter, puts the limits of stability and tuning, which depend both on Ω and Q, as

$$0 < \Omega < \min\left(Q, 2 - \frac{1}{Q}, 2Q - \frac{1}{Q}, \frac{-1/Q + \sqrt{8 + (1/Q)^2}}{2}\right), \tag{11}$$

where Ω is given as $2\sin(\pi f/f_s)$. These results show that the filter requires a significantly higher sampling rate to operate in a range equivalent to its analog counterpart.

An analysis of the digital transfer functions indicates that it might be possible to make modifications that may improve its behavior in the particular cases where the sampling rate is not very high compared to the normal analog audio bandwidth. In particular, we may observe that in the low-pass case of the original analog filter there are two theoretical zeros placed at infinity, which may not translate well to the digital scenario where we depend on a finite sampling rate (and fairly comparable to the requisite bandwidth). What we need to do is to move these zeros so that they are at the highest absolute frequency of the digital baseband, namely, the Nyquist frequency. To realize this, it is necessary to introduce a zero in each integrator, set at $z = -1$. Replacing the pure delays z^{-1} by $1 + z^{-1}$ in Equation 7, we have

$$H_{hp}(z) = 1 - \frac{(1 + z^{-1})}{Q} H_{bp}(z) - (1 + z^{-1})H_{lp}(z), \tag{12}$$

which yields the more reasonable transfer functions

$$\begin{aligned}
H_{hp}(z) &= \frac{1}{1 + \Omega/Q + \Omega^2} \\
&\quad \times \left[\frac{(1 - z^{-1})^2}{1 - \frac{(2(1-\Omega^2)z^{-1} - (1-\Omega/Q+\Omega^2)z^{-2}}{1+\Omega/Q+\Omega^2}}\right] \\
H_{bp}(z) &= \frac{\Omega}{1 + \Omega/Q + \Omega^2} \\
&\quad \times \left[\frac{1 - z^{-2}}{1 - \frac{(2(1-\Omega^2)z^{-1} - (1-\Omega/Q+\Omega^2)z^{-2}}{1+\Omega/Q+\Omega^2}}\right] \\
H_{lp}(z) &= \frac{\Omega^2}{1 + \Omega/Q + \Omega^2} \\
&\quad \times \left[\frac{(1 + z^{-1})^2}{1 - \frac{(2(1-\Omega^2)z^{-1} - (1-\Omega/Q+\Omega^2)z^{-2}}{1+\Omega/Q+\Omega^2}}\right].
\end{aligned} \tag{13}$$

Figure 2 plots the amplitude responses given by Equation 13, together with the notch response given by $H_{lp}(z) + H_{hp}(z)$. For this example, Q was set to 5 and the filter frequency was tuned to 10 kHz, with the usual sampling rate $f_s = 44.1$ kHz. We can see that we now have a well-defined set of second-order filter equations that can emulate the linear analog state variable filter responses. It is possible now to use these to implement a digital filter with the typical biquadratic equations in canonic or direct forms. For this, one final question remains, which is how to set the value of Ω so that we can provide the frequency parameter in Hz, as usual. In another publication (Lazzarini and Timoney 2022) we provided a thorough analysis of these equations and demonstrated that the process employed to derive them is absolutely equivalent to the bilinear transformation (Steiglitz 1965). Therefore, by shifting around the zeros of the transfer function, we ended up warping the analog frequency response so that infinity now lies at $f_s/2$.

Figure 2. Amplitude responses for the low-pass (dashes), high-pass (dots), band-pass (solid), and notch (solid) outputs of the biquadratic digital state variable transfer functions, $Q = 5$, $f = 10$ *kHz, and* $f_s = 44.1$ *kHz.*

Figure 3. S*-plane pole locations of second-order analog filters, showing one of each conjugate pair for three values of damping* D.

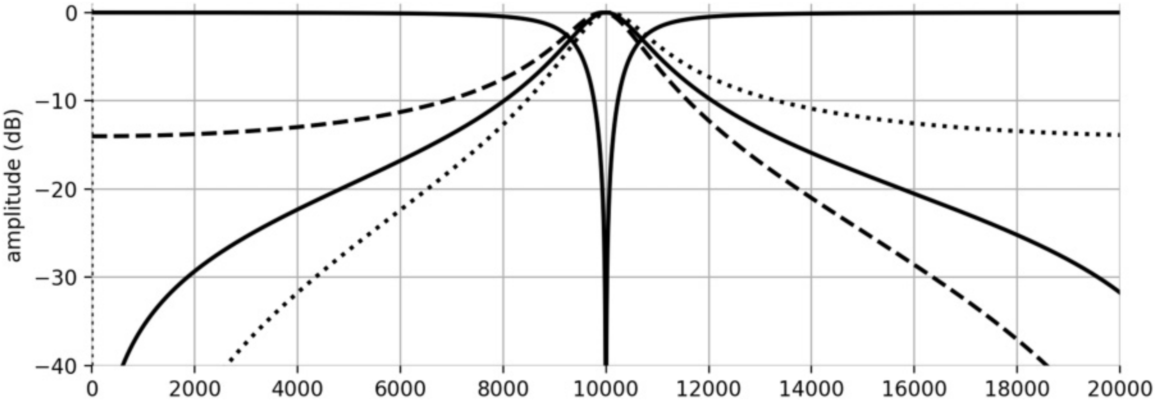


Figure 2.

This then requires us to map the digital radian frequency $\omega = 2\pi f / f_s$ to its analog counterpart Ω using

$$\Omega = \tan(\omega/2). \tag{14}$$

Sallen-Key Model

There is significant mileage to be gained from the SVF biquadratic equations we derived earlier. First, we should recognize that the SVF is just one of several analog filter configurations, and as such can realize equivalent responses to any other of these, given the correct parameters. These can be distilled into the two we have so far used, the filter cutoff (or center) frequency and Q, which is more normally given in its reciprocal form through a damping factor $D = 1/Q$. This, perhaps more appropriately, describes the peaking behavior of the filter (Hutchins 2000a). These two parameters define the type of frequency response a standard analog filter can produce, which is then realized in different configurations. A pole analysis of a second-order filter shows that the value of D will normally vary from 2, where the two conjugate poles are placed on the negative real axis of the s-plane, to zero, where they are on the imaginary axis. This implies a range of Q from 0.5 to infinity. The pole locations are always at the left-hand side of a circle with radius Ω. Different values of D will then refer to specific responses (see Figure 3): Butterworth (maximally flat, no peaking) with $D = \sqrt{2}$, Chebyshev (peaking behavior with $D > \sqrt{2}$), and Bessel ($D < \sqrt{2}$, no peaking, but an improved phase response).

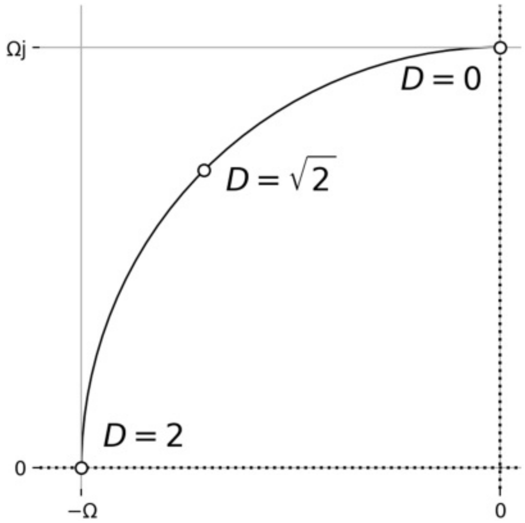


Figure 3.

For the present discussion, it is relevant that these responses can be achieved digitally with Equation 13. Therefore we can implement specific analog configurations, such as the Sallen-Key filter (see Figure 4), through these equations without having to resort to any circuit analysis. Of course, any such

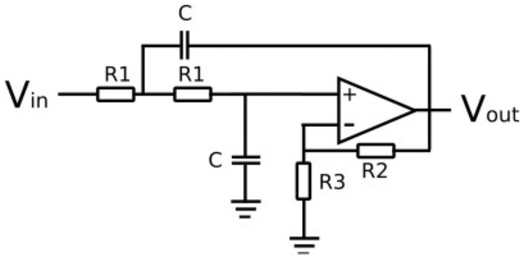


Figure 4. The Sallen-Key low-pass filter circuit, with gain K = 1 + R2/R3, *and radian cutoff frequency given as* 1/R1C.

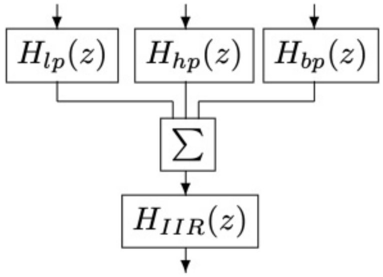


Figure 5. A linear model of the Steiner filter, made by rearranging transfer functions from the biquadratic digital SVF.

implementation provides the ideal, linear response, which nonetheless is what circuit designers would strive for. The Sallen-Key filter, which is normally offered in low- and high-pass forms, has two important parameters: its operational amplifier gain K, and the radian frequency Ω, which is determined by the values of the resistors and capacitors in a circuit that is not voltage-controlled (Hutchins 1999). The value of K is dependent on the damping D and can be easily put as $3 - D$ (Hutchins 2000a). The derivation of the SVF biquadratic equations therefore gives us a Sallen-Key model for free.

Steiner Filter Model

From the Sallen-Key model, it is possible to reach yet another classic analog filter configuration, the Steiner filter (Steiner 1977). This is an innovative approach that turns the SVF on its head: Instead of having three separate outputs and a single input, the Steiner filter has three different inputs for its high-, band-, and low-pass responses, and a single output. The circuit is designed as a modification of the second-order Sallen-Key configuration, in which the input signal is injected at different points to yield the separate responses. With Equation 13, we can emulate this approach by splitting the different numerators and the common denominator into separate filters as shown in Figure 5.

We then have a set of three parallel finite impulse response (FIR) filters mixed together, in series with a single infinite impulse response (IIR) filter that produces the output,

$$\begin{aligned} y_{hp}(n) &= x(n) - 2x(n-1) + x(n-2) \\ y_{lp}(n) &= \Omega^2[x(n) - 2x(n-1) + x(n-2)] \\ y_{bp}(n) &= \Omega[x(n) - x(n-2)] \\ y(n) &= A[y_{hp}(n) + y_{lp}(n) + y_{bp}(n)] \\ &\quad - b_1 y(n-1) - b_2 y(n-2), \end{aligned} \tag{15}$$

with $A = (1 + \Omega/Q + \Omega^2)^{-1}$, and with b_1 and b_2 set according to the SVF transfer functions (Equation 13). A plot of the amplitude responses for each individual input of this filter is shown in Figure 6. One small difference to the original Steiner filter is that its circuit could only realize a first-order high-pass filter, whereas in this implementation all responses are second order. To this end we implemented the Steiner filter in Csound as the `spf` opcode.

This is an elegant result showing that the SVF analysis is particularly useful to provide various linear filter models. We should continue to note that this is strictly linear (and fairly perfect, within the numerical constraints of a computer implementation), but a careful reading of the original intentions as reported by Steiner (1977) shows that this was exactly his goal.

The Nonlinear Digital State Variable Filter

Taken as a tolerated characteristic, or even as a desired attribute, nonlinear distortion appears to be a reality in most analog filters. This has been promoted as leading to much of the musical interest in the original circuits, even if present in fairly small amounts (if compared to, say, a typical tube distortion guitar effect, cf. Barbour 1998; Pakarinen and Yeh 2009). Such interest has inevitably been translated into a flurry of work to reproduce circuits

Figure 6. Amplitude responses for the low-pass (dashes), band-pass(solid), and high-pass (dots) inputs to the digital Steiner filter with D $= \sqrt{2}$ *(Butterworth response), centered at 4 kHz* ($f_s = 44.1$ *kHz).*

Figure 7. The original Chamberlin digital SVF block diagram.

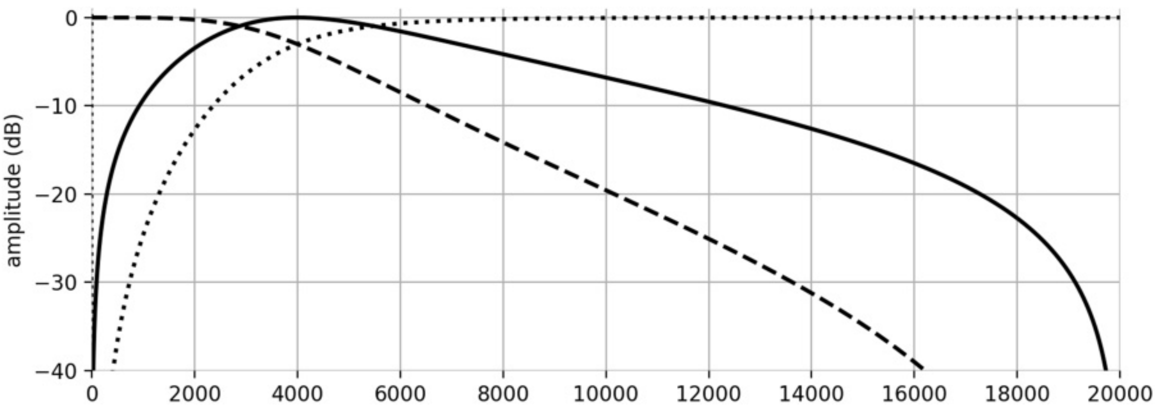


Figure 6.

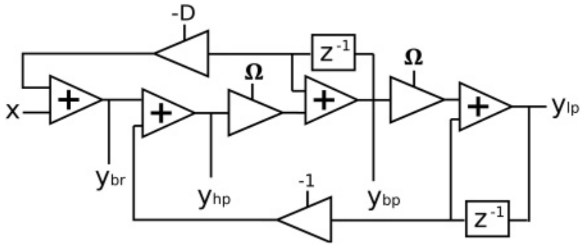


Figure 7.

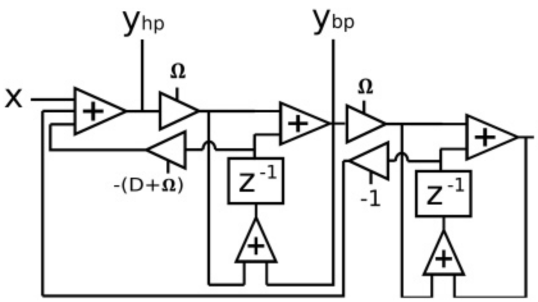
to the most-precise details by researchers and software developers of virtual analog filters. In addition to this, general interest in examining nonlinear designs continues to feature strongly in the DSP literature (cf. Holters and Zölzer 2015; Parker and Esqueda 2019). Turning back to the SVF, we have two ways to proceed if we are to consider how such nonlinear effects may be introduced in our models. We can use the biquadratic equations and try to figure how they may be modified to behave nonlinearly and then simply modify the digital filter implementation. The difficulty with this is that these equations are related to the analog filter responses and, as such, are only indirectly related to the SVF (or any other) configuration and its circuit realizations. On the one hand, that frees us from the analog models and lets us reimagine the idea of a nonlinear filter; but on the other, we do not have a clear path ahead. Nonetheless, it has been shown that it is possible to develop a generic nonlinear biquadratic digital filter (Chowdhury 2020), and so this remains a possible route that is independent of any existing analog model.

The second alternative is to look back at the SVF block diagram and an example of a real circuit realizing it, and to analyze where any nonlinearities are likely to arise. From this we would have a more definite route forward. Eventually we can start playing with the model and modifying it in ways that would not be available to an analog circuit designer. In following this path, we are again faced with two options: using the direct digital model, which may require high sampling rates for a practical realization; or deriving a new model based on the corrected responses of Equation 13. In a way, it is possible to conjecture that similar results would arise and so the point would be moot, but perhaps the best approach is to take the bilinear transformation filter as it is a more practical model that is guaranteed to work well for a range of sampling rates.

From an analysis of the biquadratic SVF equations (Lazzarini and Timoney 2022), the following variation of the Chamberlin design (whose block diagram is shown in Figure 7) can be obtained,

$$y_{hp}(n) = A[x(n) - (1/Qn + \Omega)s_{bp}(n) - s_{lp}(n)]$$
$$y_{bp}(n) = \Omega y_{hp} + s_{bp}(n)$$

Figure 8. The modified digital SVF block diagram.

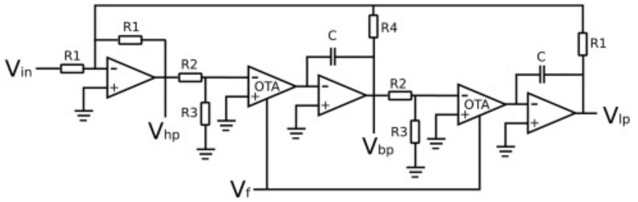


$$\begin{aligned} s_{bp}(n) &= y_{bp}(n) + \Omega y_{hp} \\ y_{lp}(n) &= \Omega y_{bp} + s_{lp}(n) \\ s_{lp}(n) &= y_{lp}(n) + \Omega y_{bp}, \end{aligned} \tag{16}$$

with $A = (1 + \Omega/Q + \Omega^2)^{-1}$. This new digital model can be easily shown as a rearrangement of the position of the two single-sample delay cells utilized in the original filter. The frequency response of this filter is exactly equivalent to the one obtained from the transfer functions of Equation 13, and so we can proceed with this design as our basic model, whose block diagram is shown in Figure 8.

Now we need to choose a real circuit that implements a voltage-controlled SVF. Many of the actual implementations are fairly equivalent to each other, using an active gain unit to control the level of the signal that is injected into each of the two integrators. Whether implemented with discrete components or with an integrated circuit, for our purposes in the present study, the standard SVF has a couple of key elements that may exhibit any significant nonlinearities. These are exactly the OTAs used as gain elements in the filter. Such components, which are current-controlled transistor-based devices, normally exhibit nonlinear saturation following a tanh() map (Lazzarini 2021). The amount of saturation depends on the signal level passing through the OTAs, and, as a rule of thumb, voltages in excess of 10 mV will put the signal within a nonlinear range. It is then common for designers to place voltage dividers at the input of the OTA that will limit the signal to the linear range. Because our purpose is to drive the signal into nonlinear territory, we may take note of this and apply a similar idea to provide a controlled amount of distortion.

Figure 9. A SVF circuit using discrete components.



An example circuit for the voltage-controlled SVF adapted from Hutchins (2000c) is shown in Figure 9. It uses inverting operational amplifiers for summing signals and integration, and OTAs for controlling the filter frequency. The voltage dividers constructed with R2 and R3 resistors are used as discussed earlier, to keep the gain cells within their linear range. The filter frequency is then given by V_f; the input signal is V_{in}; and we have V_{hp}, V_{bp}, and V_{lp}, for high-, band-, and low-pass outputs, respectively. We should also note that $R4$ controls the filter Q. In some circuits, this is also replaced by a gain cell to make the resonance voltage-controlled.

It seems reasonable to replace the linear gain elements in our digital model with tanh() waveshapers, and to apply a controllable drive inspired by the voltage dividers in the modeled circuit. This is done using the formula

$$y(n) = g^{-1}\tanh(gx(n)), \tag{17}$$

where g can now be used to either make the filter linear (if the input is beyond its linear range) or nonlinear. In the present implementation, we consider full drive (100 percent) to be defined by $g = 4$, where the absolute value of the hyperbolic tangent approximates 1. Of course, any saturation effect will depend on the input levels reaching the filter. The standard conditions under which we expect the filter to operate is to take inputs at an unchanging peak amplitude of 1.0 (0 dB full scale). In the case of signals reaching the filter at higher levels, the drive control can be used to moderate the distortion. An amplitude envelope can then be placed after the filter.

Figure 10. A translation of Equation 16 into Python code including the application of the tanh() nonlinear map (Equation 17).

Figure 11. The nonlinear digital SVF block diagram, showing the location of the nonlinearities `nln` *(a), implemented as waveshapers using a nonlinear function* f() *(b).*

```
# O is the radian frequency and D is the damping factor
A = 1/(1 + D*O + O*O)            # Scaling factor
yhp = (x - (D + O)*s1 - s2)*A    # High-pass output
w = O*tanh(yhp*g)/g              # First nonlinear mapping
ybp = w + s1                     # Band-pass output
s1 = ybp + w                     # Update state
w = O*tanh(ybp*g)/g              # Second nonlinear mapping
ylp = w + s2                     # Low-pass output
s2= ylp+w                        # Update state
```

Figure 10.

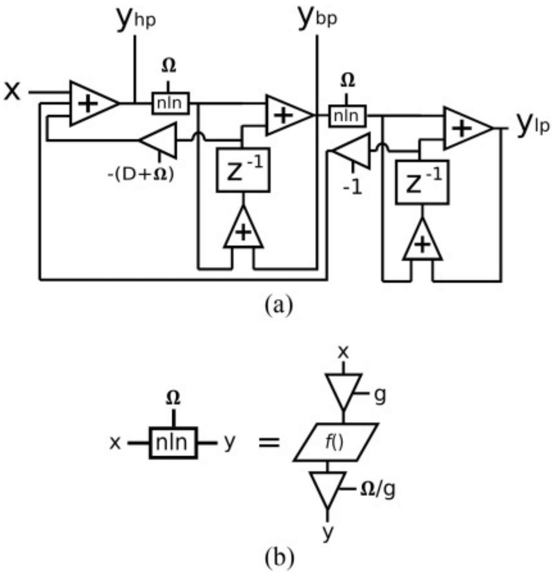


Figure 11.

An outline of the filter core in Python is shown in Figure 10, corresponding to the block diagram in Figure 11 with $f(x) = \tanh(x)$.

As noted in an earlier article (Lazzarini and Timoney 2010), the tanh() waveshaper produces a series of odd harmonics when driven by a sine wave. With enough drive to bring it to full saturation, the resulting spectrum is that of a non-band-limited square wave, but that is perhaps beyond the usual levels of distortion observed in analog filters. There is nothing stopping significant amounts of drive to be used in a creative digital context, however. Besides some aliasing and, of course, the saturation effects, there is no real limitation here, the filter continues to be stable since the tanh() effectively acts as a signal limiter. It is also interesting to note how, in some cases, the hyperbolic tangent can be used to reduce the number of harmonics, as in the case of the sine oscillator waveshaper, which produces an approximately sinusoidal signal out of a triangle wave input (Nease et al. 2018).

It is worth analyzing the spectra of the nonlinear SVF with different amounts of drive, using a typical sawtooth oscillator input. In Figure 12, we see a plot of the discrete Fourier transform of the low-pass output of the linear filter (zero drive), with cutoff frequency f_c set to 5 kHz, and $Q = 5$, where the typical resonance is observed and the filter has the expected rolloff at higher frequencies. Increasing the filter drive parameter to 0.5 ($g = 2$, or 50 percent), we observe reduced filter peaking, due to the limiting effects of the tanh() map, but we also see a somewhat steeper rolloff, particularly in the stopband (see Figure 13). Finally, with full drive, these characteristics are even more apparent: no resonance at the cutoff frequency and significant high frequency attenuation, as depicted in Figure 14. Some of these effects have already been noted in the literature (Rossum 1992; Välimäki et al. 2011).

Driving the waveshapers beyond what we are deeming 100 percent in this context does not seem to change the spectrum significantly, but it will of course increase the presence of audible aliasing. As noted earlier, the amount of nonlinear distortion in analog filter circuits tends to be small, as designers attempt to minimize its effects. Within a digital environment, however, it is possible to experiment

Figure 12. Amplitude spectrum taken from the low-pass output of a linear digital SVF, with $f_c = 5$ *kHz, Q = 5, and* $f_s = 44.1$ *kHz. The filter input is a full-spectrum digital sawtooth at 200 Hz.*

Figure 13. Amplitude spectrum taken from the low-pass output of a nonlinear digital SVF using 50 percent drive, with $f_c = 5$ *kHz, Q = 5, and* $f_s = 44.1$ *kHz. The filter input is a full-spectrum digital sawtooth at 200 Hz.*

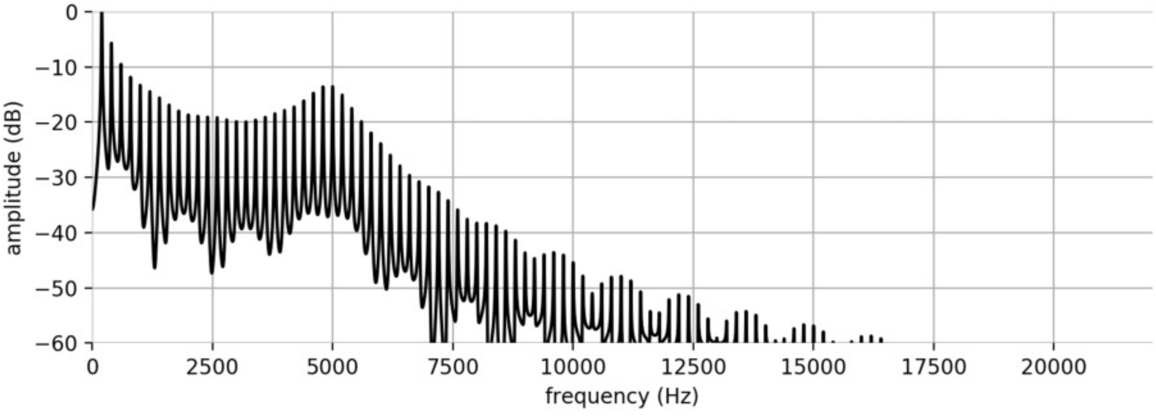


Figure 12.

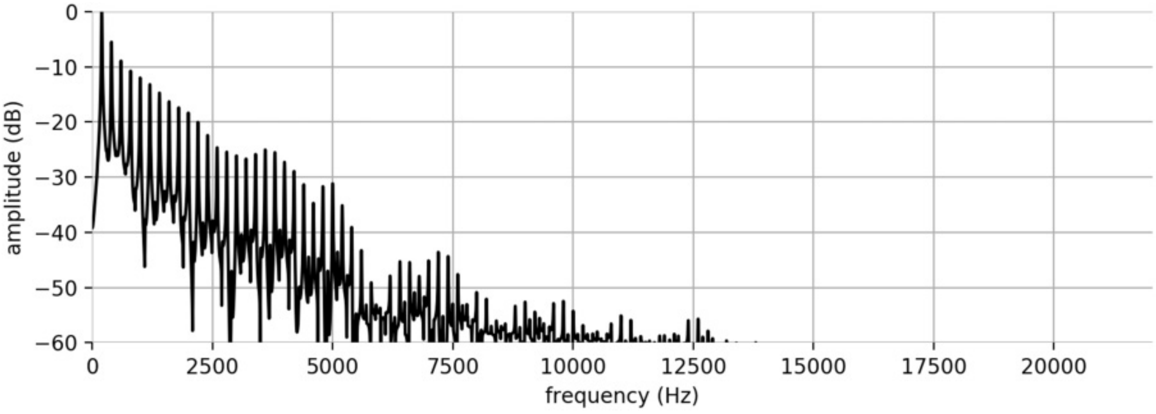


Figure 13.

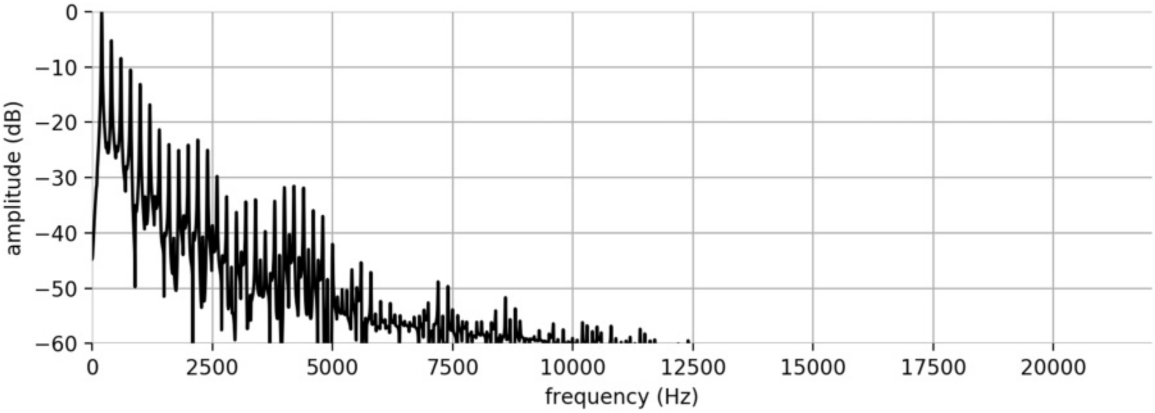
with this parameter to different levels of saturation to find the appropriate sound for each application.

Beyond the Analog Models

This brings us to the next question: Do we need to restrict ourselves to the tanh() map? It would be a failure of imagination to stop there. We have seen good reasons to insert nonlinear distortion into the model, but as we know from waveshaping, there is a large collection of functions that can be used for this purpose (LeBrun 1979; Arfib 1979; Timoney et al. 2010). We will have to take care, of course, to assure filter stability, which is perhaps the only real limitation here—we do not want the filters to break. The aliasing question, within a creative environment that may be dominated by all sorts of noise and nonstandard effects, is often moot. Although it may be an excellent exercise to study aliasing problems, they are beyond the scope of this article.

A practical approach to liberate our model from a specific analog model is to replace the tanh() code in our implementation with a lookup table and allow users to supply a function table of their choice. This

Figure 14. Amplitude spectrum taken from the low-pass output of a nonlinear digital SVF using 50 percent drive, with $f_c = 5$ *kHz,* $Q = 5$*, and* $f_s = 44.1$ *kHz. The filter input is a full-spectrum digital sawtooth at 200 Hz.*



is implemented in the Csound opcode `svn`, with the syntax

```
ahp, alp, abp, abr svn asig, xcf, xQ,
kdrv[, ifn, inrm, imx, istor],
```

providing a nonlinear SVF with adjustable drive (`kdrv`). By default it uses a tanh() map, but we can supply an alternative waveshaper in a function table (`ifn`). The unit generator simultaneously produces the four usual outputs: high, low, and band pass, and notch.

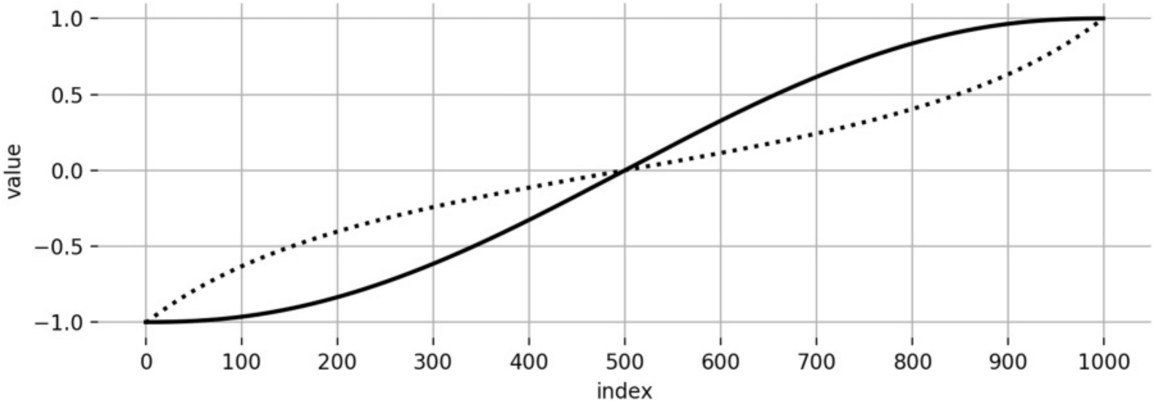
A useful way to select distortion maps is to consider the shapes provided by polynomials. There are two particular reasons for this. First, we can tap into the well-developed theory of waveshaping, which has results that are germane for this case (Lazzarini 2021). Second, a simple analysis of the polynomial shapes will help us to weed out maps that can lead to unstable filters. A sound approach is to start with polynomials that have only odd-numbered terms, that is, with all even coefficients set to zero. This guarantees, for example, that no DC term will be present at the output, something that, compounded with the feedback, can make the filter unstable right out of the box. Additionally, it will produce an odd function, which is also a good characteristic, shaping both the negative and the positive sides of a bipolar input similarly. Another interesting characteristic to note with odd polynomial functions is that, in general, with coefficients that decay significantly with term order, by alternating the signs of terms $(+, -, +, -, \ldots)$ we can generate maps with good limiting characteristics, as in the case of the tanh() function. An example is shown in Figure 15, where two normalized function tables are shown for two polynomials that differ only in terms of coefficient signs. We must stress, however, that waveshapers need not necessarily be designed with such particular characteristics.

Chebyshev Polynomials

Designing polynomials blindly can be something of a hit-and-miss process. The suggestions given earlier are a good start, but may be somewhat limiting. One interesting approach is to use Chebyshev polynomials of the first kind to construct a nonlinear map (LeBrun 1979). Their main characteristic is that, when driven by a sinusoid, they can produce a precisely defined spectrum. Although their use within the SVF is somewhat different, of course, the approach can be fertile for experimentation.

When applying a Chebyshev polynomial-based map (or indeed any nonlinear map), we may want to review the way we normalize the signal coming out of the waveshaper. Although the simple method we devised, following an analog circuit, may have been appropriate for the tanh() function, perhaps we can now look elsewhere for inspiration. A common way to provide dynamic spectra, one that is independent of peak amplitude in waveshaping synthesis, is to use a scaling function derived from the nonlinear map itself. This is constructed by examining the values of a polynomial table, so that the output can

Figure 15. Nonlinear distortion function tables constructed using $f(x) = x - 0.5x^3 + 0.15x^5 - 0.1x^7 + 0.05x^9$ *(solid) and* $f(x) = x + 0.5x^3 + 0.15x^5 + 0.1x^7 + 0.05x^9$ *(dots) in the interval [−1,1].*



be normalized for varying input amplitudes. The revised nonlinear distortion component of the SVF becomes

$$y(n) = a(g)f(gx(n)), \tag{18}$$

where $f()$ is the nonlinear function and $a()$ is the scaling function derived from it. In Csound such a function table can be produced with GEN 4. It can then be supplied to **svn** through its optional inrm parameter. This modification, combined with a well-designed Chebyshev map, has the effect of turning the nonlinear SVF into a hybrid of digital waveshaper and filter. It is clearly possible to obtain a noticeable addition of new harmonics by varying the drive control, giving the unit three means of controlling dynamic spectra (filter frequency, Q, and drive). This has always been the case with the standard tanh() map, of course, but the kind of spectral shaping it supports is somewhat more subtle. It is also important to note that, whereas in the standard case of saturation nonlinearities the filter frequency tends to be somewhat stable and well tuned, this is not guaranteed to be the general case here.

We can examine the case of the polynomial

$$f(x) = 0.916 + 4x - 6x^2 - 13.3x^3 + 10x^4 + 19.2x^5 - 5.3x^6 - 9.14x^7, \tag{19}$$

created from the Chebyshev series [0, 1, −1/2, −1/3, 1/4, 1/5, −1/6, −1/7] (see Figure 16). When driven with a sine wave at amplitude 1, this would generate a bandlimited sawtooth with 7 harmonics. Used in the SVF low-pass filter with $Q = 10$ and $f_c = 2$ kHz, at full drive with a sine wave input, it produces the spectrum in Figure 17. Note the resonance region at 3,400 Hz, which is significantly beyond the filter frequency. Such frequency shifts, away from the nominal cutoff value, are a common effect of nonlinear distortion, as noted by Rossum (1992).

This example is a somewhat extreme case, of course, because we are using a polynomial that produces a waveshaper output with significant bandwidth. With nonsinusoidal inputs, aliasing is guaranteed with even moderate amounts of drive. Also, because Chebyshev polynomials with even terms may produce some energy at 0 Hz when they are not fully driven, even if the first term of the series is zero, it is useful to utilize a DC blocker at the filter output when experimenting with such nonlinear maps. For a case that is perhaps better behaved, we may consider setting the even terms of the Chebyshev series to zero, producing a strictly odd polynomial. The example in Figure 18 shows such a nonlinear map, which may be more easy to handle.

Discussion

There is perhaps a fundamental difference between analog and digital filters, which goes beyond the divide between discrete and continuous time. It

Figure 16. Nonlinear distortion function table constructed using the polynomial $f(x) = 0.916 + 4x - 6x^2 - 13.3x^3 + 10x^4 + 19.2x^5 - 5.3x^6 - 9.14x^7$, *normalized, in the interval [−1,1].*

Figure 17. Amplitude spectrum taken from the low-pass output of a nonlinear digital SVF using a nonlinear map depicted in Figure 16, with 100 percent drive, $f_c = 2$ *kHz,* $Q = 10$, *and* $f_s = 44.1$ *kHz. The filter input is a sine wave at 200 Hz.*

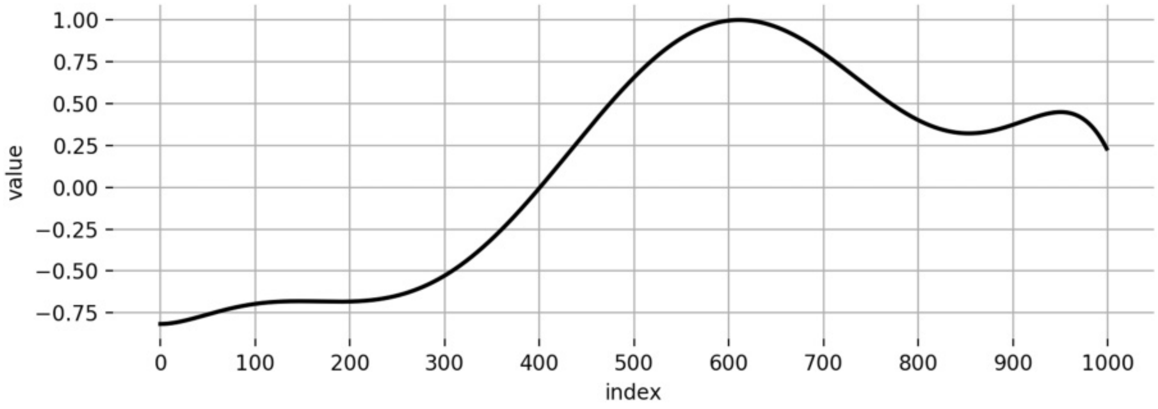


Figure 16.

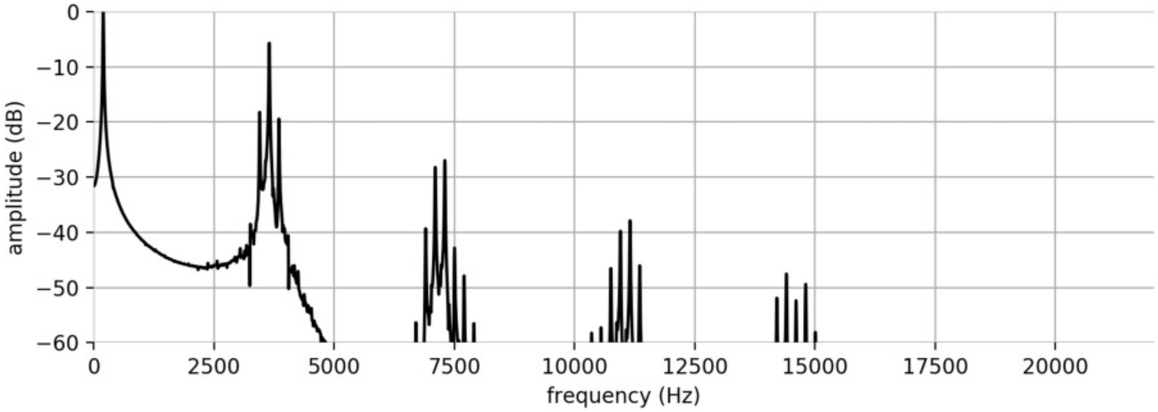
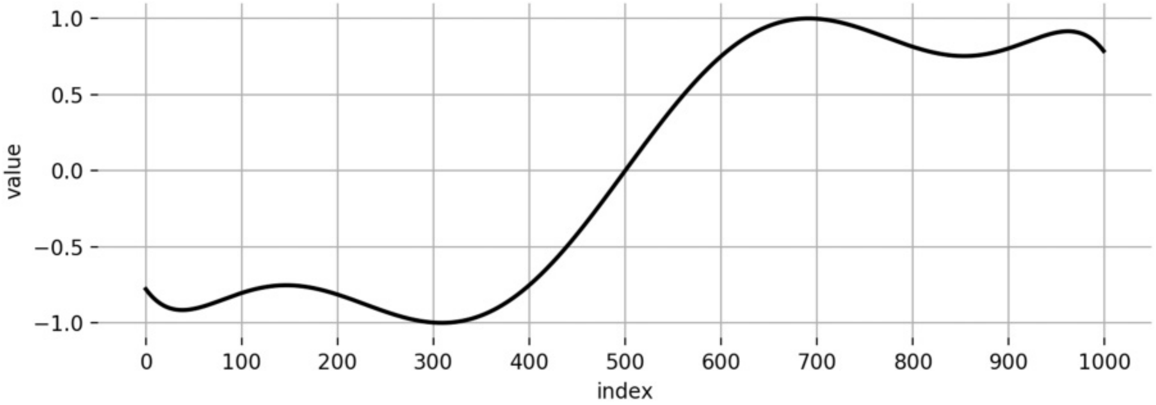


Figure 17.

relates closely to the dissimilarities between analog and digital (stored-program) computing; after all, we can trace devices such as modular synthesizers back to the traditional analog computers of yesteryear (Lazzarini and Timoney 2020). (In fact, the SVF itself can be described as an analog computer program.) Digital computer programs can be studied directly as stable implementations of mathematical equations, whereas analog circuits are more akin to simulations of these. This fact explains much of the methodological divergence in filter design that occurs between analog and digital. Hutchins (2000a) describes the development of analog filters in terms of response selection, followed by configuration choice, and completed by fine-tuning the circuits. As we noted at the outset, within a digital environment we have the choice of FIR or IIR filters, depending on the application, and can then proceed to either apply a given frequency response directly or to choose a known parametrical method to compute coefficients for a standard design (which may be the closest to the idea of configurations as we can get).

In analog filter design, of course, creative invention may be a disruptive factor. The Moog ladder filter is the classic example: Starting with a low-pass design that supported voltage control, the simple

Figure 18. Nonlinear distortion function table constructed using the strictly odd polynomial $f(x) = 4x - 13.3x^3 + 19.2x^5 - 9.14x^7$, *normalized, in the interval [−1,1].*



modification of feedback allowed a distinctly musical behavior to flourish. Only many years later was someone able to show how this lifted the poles from the real axis into a cross pattern on the *s*-plane, something unique that did not happen in other cases. Such creative invention has also been present throughout the design of computer music instruments since the beginning, and so we should not lose sight of it. Using ideas from analog signal processing as inspiration and motivation, rather than focusing only on the exact reproduction of particular circuit operations, appears to be a good way to proceed.

When we examine the applications of nonlinearities in digital filters, we observe that this is generally reduced to the use of saturation maps in general, and the hyperbolic tangent specifically (Timoney and Lazzarini 2011). Of course, there are good reasons for this, since such functions have been shown to work well and produce stable systems. Nevertheless, a strong motivation behind these uses is that they stem directly from what is measured and observed in analog circuits with certain components. Consequently, from a perspective of claims that can be made about a particular digital implementation, designers are able to show that it is strictly bound to a given analog device. This may be an important consideration, as we noted earlier, if such characteristics are needed to promote, commercially or otherwise, the resulting virtual analog model. If we, for a moment, try to decouple our work in digital systems from the realities of circuits, however, we can start exploring different forms of nonlinear distortion that have never been applied to these particular cases. In fact, in doing so, we are able to create hybrid structures, which may be considered part filter, part digital waveshaper. Although, as we noted earlier, aliasing is a factor in this, there are ways of dealing with it that may suppress or minimize its effect, or alternatively it can be used as a creative element by musicians, as a particular characteristic of the design. It is important to note that not all situations in which aliasing appears should be seen as defective a priori. If this becomes an issue, techniques of oversampling can be used to overcome it.

There are also other processes through which nonlinearity can be exploited in digital filters, also leading to such hybridization. One of these is coefficient modulation, which may be interpreted both as some sort of feedback amplitude modulation or as a periodic time-varying filter (Kleimola et al. 2011). Such an approach actually takes what is a perfectly linear time-invariant filter and makes it nonlinear through transforming it into a time-variant system. This type of invention is something that takes advantage of the fact that we have a digital signal operation at hand, completely decoupled from any existing analog idea of a time-varying filter. Another example is time-varying convolution (Lazzarini 2020), which takes a digital convolution engine and makes it a generalized cross-synthesis process in which the concept of something involving an input and an impulse response is transcended. Finally, we can still take inspiration from analog voltage-controlled environments to initiate some ideas such as filter frequency modulation, but then develop it as a purely digital-systems pursuit. This is in fact a field of computer music instrument research and development that is still in need of more attention.

Conclusions

In this article, we reviewed the fundamental principles of digital filter design for musical applications. Starting with the analog state variable filter (SVF), described at high level from its block diagram, we derived its digital biquadratic transfer function, which is equivalent to bilinear transformation applied to an s-domain biquadratic filter. We demonstrated that the approach was sufficient not only to give us a digital version of the SVF, but also to provide models for other configurations, such as the Sallen-Key and Steiner filters, without the need to perform any real circuit analysis. Moving to nonlinear implementations, we returned to a digital model of the SVF, derived from its adjusted bilinear transfer functions, and—taking inspiration from an analog circuit implementation—we inserted two hyperbolic tangent waveshapers into the filter equations. From this point, we observed that it was possible to move beyond the analog model to develop a hybrid digital-waveshaper-and-filter unit generator. Finally, we discussed the merits of such an approach that is less firmly based on detailed analysis of existing circuits, but takes advantage of aspects that may only be developed and implemented under a digital audio processing environment. It seems to us as a conclusion that, for computer music applications at least, this way of going about the development of software musical instruments is full of creative potential.

References

Arfib, D. 1979. “Digital Synthesis of Complex Spectra by Means of Multiplication of Non Linear Distorted Sine Waves.” *Journal of the Audio Engineering Society* 27(10):Paper 1319.

Barbour, E. 1998. “The Cool Sound of Tubes [Vacuum Tube Musical Applications].” *IEEE Spectrum* 35(8):24–35. 10.1109/6.708439

Chamberlin, H. 1985. *Musical Applications of Microprocessors*. Indianapolis, Indiana: Hayden.

Chowdhury, J. 2020. “Stable Structures for Nonlinear Biquad Filters.” In *Proceedings of the International Conference on Digital Audio Effects*, pp. 94–100.

Chowning, J. 1973. “The Synthesis of Complex Audio Spectra by Means of Frequency Modulation.” *Journal of the Audio Engineering Society* 21(7):527–534. Reprinted in *Computer Music Journal*. 1977. 1(2):46–54.

Colin, D. P. 1971. “Electrical Design and Musical Applications of an Unconditionally Stable Combination Voltage Controlled Filter/Resonator.” *Journal of the Audio Engineering Society* 19(11):923–927.

D’Angelo, S., and V. Välimäki. 2014a. “Generalized Moog Ladder Filter, Part I: Linear Analysis and Parameterization.” *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22(12):1825–1832.

D’Angelo, S., and V. Välimäki. 2014b. “Generalized Moog Ladder Filter, Part II: Explicit Nonlinear Model through a Novel Delay-Free Loop Implementation Method.” *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22(12):1873–1883. 10.1109/TASLP.2014.2352556

Dattorro, J. 1997. “Effect Design, Part 1: Reverberator and Other Filters.” *Journal of the Audio Engineering Society* 45(9):660–684.

Fontana, F. 2007. “Preserving the Structure of the Moog VCF in the Digital Domain.” In *Proceedings of the International Computer Music Conference*, pp. 291–294.

Fontana, F., and M. Civolani. 2010. “Modeling of the EMS VCS3 Voltage-Controlled Filter as a Nonlinear Filter Network.” *IEEE Transactions on Audio, Speech, and Language Processing* 18(4):760–772. 10.1109/TASL.2010.2046287

Gabrielli, L., et al. 2013. “A Digital Waveguide-Based Approach for Clavinet Modeling and Synthesis.” *EURASIP Journal on Advances in Signal Processing* 2013:Art. 103. 10.1186/1687-6180-2013-103

Holters, M., and U. Zölzer. 2015. “A Generalized Method for the Derivation of Non-Linear State-Space Models from Circuit Schematics.” In *Proceedings of the European Signal Processing Conference*, pp. 1073–1077.

Huovilainen, A. 2004. “Non-Linear Digital Implementation of the Moog Ladder Filter.” In *Proceedings of the International Conference on Digital Audio Effects*, pp. 61–64.

Hutchins, B. 1978. “Additional Ideas for Voltage Controlled Filters.” *Electronotes* 10(85):5–17.

Hutchins, B. 1982. “Integrated Musical Electronics, Part 3: Better Use of VCF Chips.” *Electronotes* 14(143):1–10.

Hutchins, B. 1999. “Analog Signal Processing: Chapter 1.” *Electronotes* 19(191):4–23. Available online at electronotes.netfirms.com/EN191.pdf. Accessed February 2022.

Hutchins, B. 2000a. “Analog Signal Processing: Chapter 3.” *Electronotes* 19(192):2–43. Available online at electronotes.netfirms.com/EN192.pdf. Accessed February 2022.

Hutchins, B. 2000b. “Analog Signal Processing: Chapter 6.” *Electronotes* 20(194):14–30. Available online at electronotes.netfirms.com/EN194.pdf. Accessed February 2022.

Hutchins, B. 2000c. “Analog Signal Processing: Chapter 8.” *Electronotes* 20(196):2–10. Available online at electronotes.netfirms.com/EN196.pdf. Accessed February 2022.

Hutchins, B. 2013. “Revisiting Some VCF Ideas—and a Few New Ideas.” *Electronotes* 23(215):1–23. Available online at electronotes.netfirms.com/EN215.pdf. Accessed February 2022.

Kerwin, W., L. Huelsman, and R. Newcomb. 1967. “State-Variable Synthesis for Insensitive Integrated Circuit Transfer Functions.” *IEEE Journal of Solid-State Circuits* 2:87–92. 10.1109/JSSC.1967.1049798

Kleimola, J., et al. 2011. “Feedback Amplitude Modulation Synthesis.” *EURASIP Journal on Advances in Signal Processing* 2011:Art. 434,378. 10.1155/2011/434378, PubMed: 24348546

Lazzarini, V. 2020. “Parallel Computation of Time-Varying Convolution.” *Journal of New Music Research* 5(49):1–13.

Lazzarini, V. 2021. *Spectral Music Design: A Computational Approach*. Oxford: Oxford University Press.

Lazzarini, V., and J. Timoney. 2010. “New Perspectives on Distortion Synthesis for Virtual Analog Oscillators.” *Computer Music Journal* 31(1):28–40. 10.1162/comj.2010.34.1.28

Lazzarini, V., and J. Timoney. 2020. “The Analogue Computer as a Musical Instrument.” In V. Lazzarini et al., eds. *Ubiquitous Music Ecologies*. London: Routledge, pp. 212–234.

Lazzarini, V., and J. Timoney. 2022. “Improving the Chamberlin Digital State Variable Filter.” *Journal of the Audio Engineering Society* 70(6).

LeBrun, M. 1979. “Digital Waveshaping Synthesis.” *Journal of the Audio Engineering Society* 27(4):250–266.

Mathews, M., and J. E. Miller. 1964. *MUSIC IV Programmer’s Manual*. Murray Hill, New Jersey: Bell Telephone Laboratories.

Moog, R. 1965. “Voltage Controlled Electronic Music Modules.” *Journal of the Audio Engineering Society* 13(3):200–206.

Nease, S. H., A. D. Lanterman, and J. O. Hasler. 2018. “Applications of Current-Starved Inverters to Music Synthesis on Field Programmable Analog Arrays.” *Journal of the Audio Engineering Society* 66(1/2):71–79. 10.17743/jaes.2017.0044

Oppenheim, A. V., R. W. Schafer, and J. R. Buck. 1999. *Discrete-Time Signal Processing*. Upper Saddle River, New Jersey: Prentice-Hall.

Pakarinen, J., and D. T. Yeh. 2009. “A Review of Digital Techniques for Modeling Vacuum-Tube Guitar Amplifiers.” *Computer Music Journal* 33(2):85–100. 10.1162/comj.2009.33.2.85

Pakarinen, J., et al. 2011. “Recent Advances in Real-Time Musical Effects, Synthesis, and Virtual Analog Models.” *EURASIP Journal on Advances in Signal Processing*. 10.1155/2011/940784

Parker, J. 2011. “A Simple Digital Model of the Diode-based Ring Modulator.” In *Proceedings of the International Conference on Digital Audio Effects*, pp. 163–166.

Parker, J., and F. Esqueda. 2019. “Modelling of Nonlinear State-Space Systems Using a Deep Neural Network.” In *Proceedings of the International Conference on Digital Audio Effects*, Paper 42. Available online at www.dafx.de/paper-archive/2019/DAFx2019_paper_42.pdf. Accessed February 2022.

Raffel, C. 2010. “Practical Modeling of Bucket-Brigade Device Circuits.” In *Proceedings of the International Conference on Digital Audio Effects*, pp. 50–56.

Risset, J. C. 2009. “Max Mathews’s Influence on (My) Music.” *Computer Music Journal* 33(3):26–34. 10.1162/comj.2009.33.3.26

Rossum, D. 1992. “Making Digital Filters Sound Analog.” In *Proceedings of the International Computer Music Conference*, pp. 30–32.

Steiglitz, K. 1965. “The Equivalence of Digital and Analog Signal Processing.” *Information and Control* 8(5):455–467. 10.1016/S0019-9958(65)90374-8

Steiglitz, K. 1996. *A Signal Processing Primer*. Menlo Park, California: Addison-Wesley.

Steiner, N. 1977. “A Super Simple Three-Mode Simultaneous Input Variable Resonance, Voltage-Controlled Filter for Signal Processing.” In *Proceedings of the 58th AES Convention*, pp. 1–10.

Stilson, T., and J. Smith. 1996. "Analyzing the Moog VCF with Considerations for Digital Implementation." In *Proceedings of the International Computer Music Conference*, pp. 398–401.

Timoney, J., and V. Lazzarini. 2011. "Saturation Non-Linearities for Virtual Analog Filters." In *Proceedings of Forum Acusticum*. CD-ROM.

Timoney, J., V. Lazzarini, and T. Lysaght. 2008. "A Modified Frequency Modulation Approach to Bandlimited Signal Generation." In *Proceedings of the International Conference on Digital Audio Effects*, pp. 27–33.

Timoney, J., et al. 2010. "Digital Emulation of Distortion Effects by Wave and Phase Shaping Mehtods." In *Proceedings of the International Conference on Digital Audio Effects*, pp. 419–422.

Välimäki, V., et al. 2011. "Virtual Analog Effects." In U. Zölzer, ed. *DAFX: Digital Audio Effects*. Chichester, UK: Wiley, pp. 473–522.

Werner, K., and R. McClellan. 2020. "Moog Ladder Filter Generalizations Based on State Variable Filters." In *Proceedings of the International Conference on Digital Audio Effects*, pp. 70–77.

Wiltshire, T. 2008. "CEM3320 Filter Designs." *Electric Druid* (blog). Available online at electricdruid.net/cem3320-filter-designs. Accessed March 2021.