Digital Sound Generation – Part 2

Beat Frei
Institute for Computer Music and Sound Technology (ICST)
Zurich University of the Arts
Baslerstrasse 30, CH-8048 Zürich, Switzerland
beat.frei@zhdk.ch, http://www.icst.net

Preface

This part covers filters for sound synthesis including practical examples of oversampling, amplitude compression, and efficient parameter update schemes. Filters for equalizers and vocoders will be treated in a future part on effect design.

Table of Contents

3 Main Filters

3.1 General Requirements

Apart from stability and the desired frequency response, a good general purpose synthesizer filter has to meet additional criteria, some of which are scarcely discussed in classical filter theory. The following list gives an overview of the key requirements. Items 6 and 7 apply specifically to virtual analog filters.

1) Independent control of the cutoff frequency f_c and the quality factor Q ("resonance").

   Although musical reasons call for decoupled external control inputs, the internal filter coefficients may well be complex functions of both parameters as long as they can be interpolated efficiently such that the filter remains stable and retains its characteristics during a sweep. In this case, the interpolation runs at audio rate whereas the expensive mapping of the control parameters to the coefficients takes place at a slower control rate. However, if items 6 and 7 are of major concern, it's advisable to strive for filter structures with inherent decoupled Q control in order to facilitate efficient and systematic amplitude limiting.

2) Constant Q and a well-defined slope at any cutoff frequency in the audio band.

   This is vital for a full sound. Simple filters are typically compromised in the top octave: Their magnitude response flattens, f_c and Q deviate and become coupled. Comparative listening tests with various f_c sweeps on a sawtooth and an excellent filter as reference are recommended as a first metric to rule out inferior designs. Common flaws widely perceived as such are: Reduced filter effectiveness already at $f_c \approx 10$ kHz, a maximum cutoff frequency well inside the audio band at high Q, residual lowpass action at maximum f_c, bandwidth narrowing of the resonant peak at high f_c, a decreasing f_c when Q is raised. Considering today's standards, designs suffering from any of these flaws run a significant risk of being rejected by musicians.

3) Exponential mapping of the frequency control input to f_c. See section 3.5.

4) Stability and musical behavior when f_c is modulated by a fast control or even an audio signal. See section 3.5.

5) No perceivable additional noise. See section 3.6.

6) A sonically pleasing saturation that produces smooth f_c sweeps at high Q settings. See sections 3.4 and 3.7.

7) Self oscillation at maximum Q. See sections 3.4 and 3.7.

An overview of digital filter design strategies is given in [1]. One of the key findings is that good compromises between efficient audio processing and low control complexity are obtained by discretizing the integrators or entire first order blocks of a continuous time prototype filter while preserving its original topology. Unit delays are inserted to eliminate delay-free loops. The resulting filters not only behave well with time-varying coefficients, something that is crucial to synthesizer applications, but also generate only negligible noise when processed with single precision arithmetic. Actually, all designs in the main filter section are based on this approach.

3.2 Chamberlin State Variable Filter

The Chamberlin structure is obtained by replacing the integrators of a continuous time state variable filter with a forward and a backward difference block. It has been introduced to computer music in [2] and attained much popularity and successive refinement since. A variety of advantages makes it a good starting point for high quality synthesizer filters: Reasonable decoupling and straightforward control of f_c and Q, computational efficiency, excellent numerical properties, and a complete set of filter types. A major drawback is the limited f_c range, particularly for low Q factors.

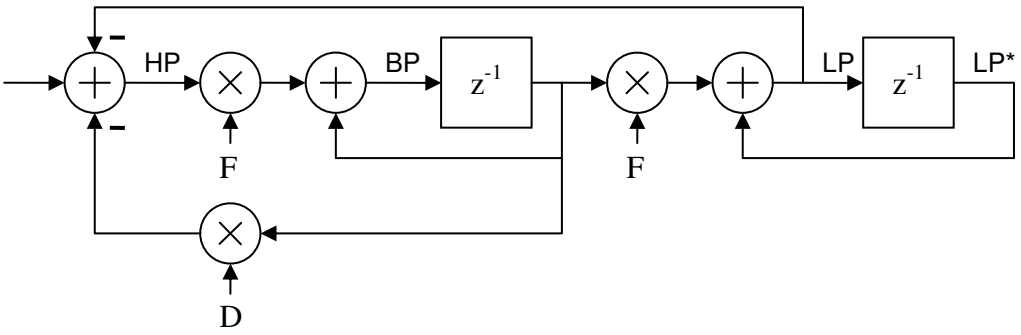


Fig. 1: Chamberlin State Variable Filter

The transfer functions of the most important filter types are listed below. Notch (= bandstop) and peaking filters are created by taking the sum and the difference of the low and high pass output. F controls the natural frequency, whereas D sets the damping ratio, which in turn is inversely proportional to Q. Both relations are approximately linear for low values.

$$\Delta = z^2 + \left(F^2 + DF - 2\right)z + \left(1 - DF\right)$$

Lowpass: $$H_{LP}(z) = \frac{F^2 z}{\Delta}$$

Bandpass: $$H_{BP}(z) = \frac{Fz(z-1)}{\Delta}$$

Highpass: $$H_{HP}(z) = \frac{(z-1)^2}{\Delta}$$

Notch: $$H_N(z) = H_{LP}(z) + H_{HP}(z) = \frac{F^2 z + (z-1)^2}{\Delta}$$

Peaking: $$H_{PK}(z) = H_{LP*}(z) - H_{HP}(z) = \frac{F^2 - (z-1)^2}{\Delta}$$

In the undamped case (D = 0), the natural frequency coincides with the cutoff or center frequency f_c, and the following relation holds with f_s denoting the sample rate:

$$F = 2\sin(\pi f_c / f_s)$$

Based on the stability triangle (see section 3.5.3), the filter is found to be stable for:

$$F^2 + 2DF < 4 \qquad \text{with} \qquad F > 0,\ D > 0$$

We conclude our primary analysis of the digital filter by comparing it to the analog archetype. The Chamberlin lowpass is chosen as example and its frequency response calculated:

$$H_{LP}\left(e^{j\Omega}\right)=\frac{F^2 e^{j\Omega}}{e^{2j\Omega}+\left(F^2+DF-2\right)e^{j\Omega}+\left(1-DF\right)}=\frac{F^2}{F^2+\left(1-\cos\Omega\right)\left(DF-2\right)+jDF\sin\Omega}$$

An approximation for low signal frequencies including second order terms yields:

$$H_{LP}\left(e^{j\Omega}\right)\approx\frac{F^2}{F^2-\Omega^2\left(1-\frac{DF}{2}\right)+jDF\Omega}=\frac{F^2}{F^2-\left(\frac{2\pi}{f_s}\right)^2\left(1-\frac{DF}{2}\right)f^2+j\frac{2\pi DF}{f_s}f}\quad;\ f<<\frac{f_s}{2\pi}$$

The continuous time state-variable lowpass filter with natural frequency f_o and damping ratio d has the following transfer function $G_{LP}(s)$ and frequency response $G_{LP}(jf)$:

$$G_{LP}(s)=\frac{\omega_o^{\,2}}{s^2+2d\omega_o s+\omega_o^{\,2}}\qquad G_{LP}(jf)=\frac{f_o^{\,2}}{f_o^{\,2}-f^2+2jdf_o f}$$

A comparison leads to approximate formulae, valid for F << 1 and DF << 1:

$$f_o\approx\frac{Ff_s}{2\pi\sqrt{1-\frac{DF}{2}}}\approx\frac{Ff_s}{2\pi}\left[1+\frac{DF}{4}\right]\approx\frac{Ff_s}{2\pi}\qquad d=\frac{1}{2Q}\approx\frac{D}{2\sqrt{1-\frac{DF}{2}}}\approx\frac{D}{2}\left[1+\frac{DF}{4}\right]\approx\frac{D}{2}$$

We see now that the natural frequency and the quality factor depend on both filter coefficients and how the discretized filter converges to the continuous one with decreasing F. Simulations reveal the behavior at high F and f_s = 48 kHz in Fig. 2-4 pointing out that the discretization is fairly precise for F < 0.5 and D < 1.

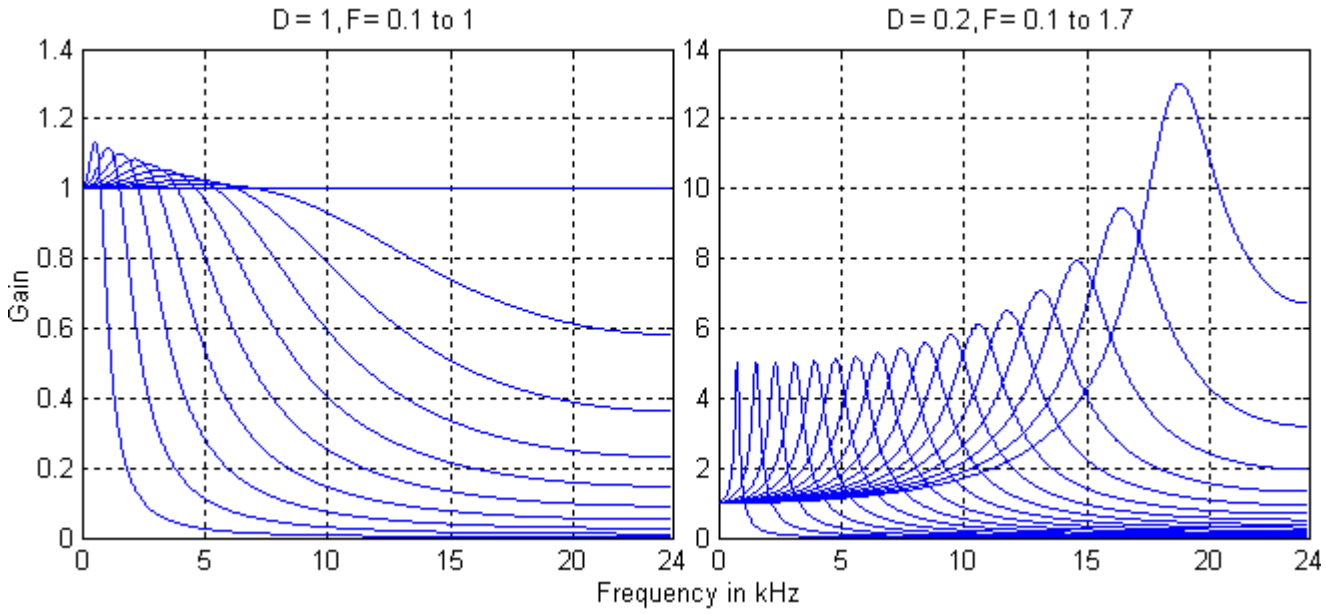


Fig. 2: Magnitude Response of the Chamberlin Lowpass Filter for Varying F

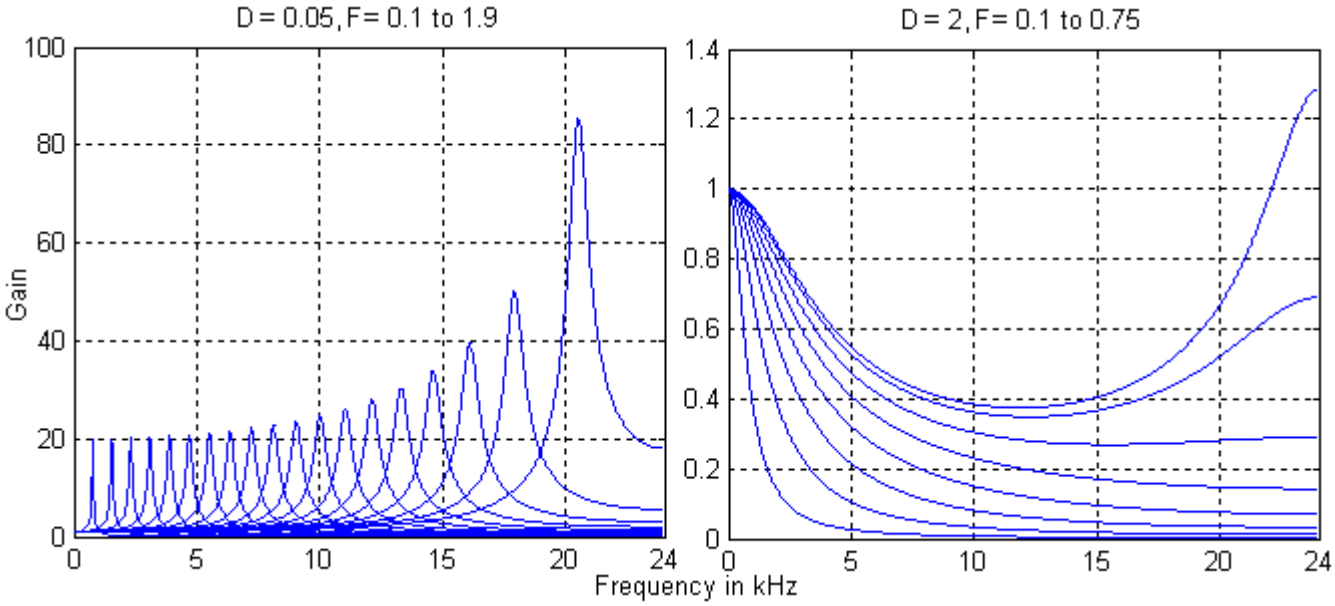


Fig. 3: Magnitude Response of the Chamberlin Lowpass Filter for Varying F (cont.)

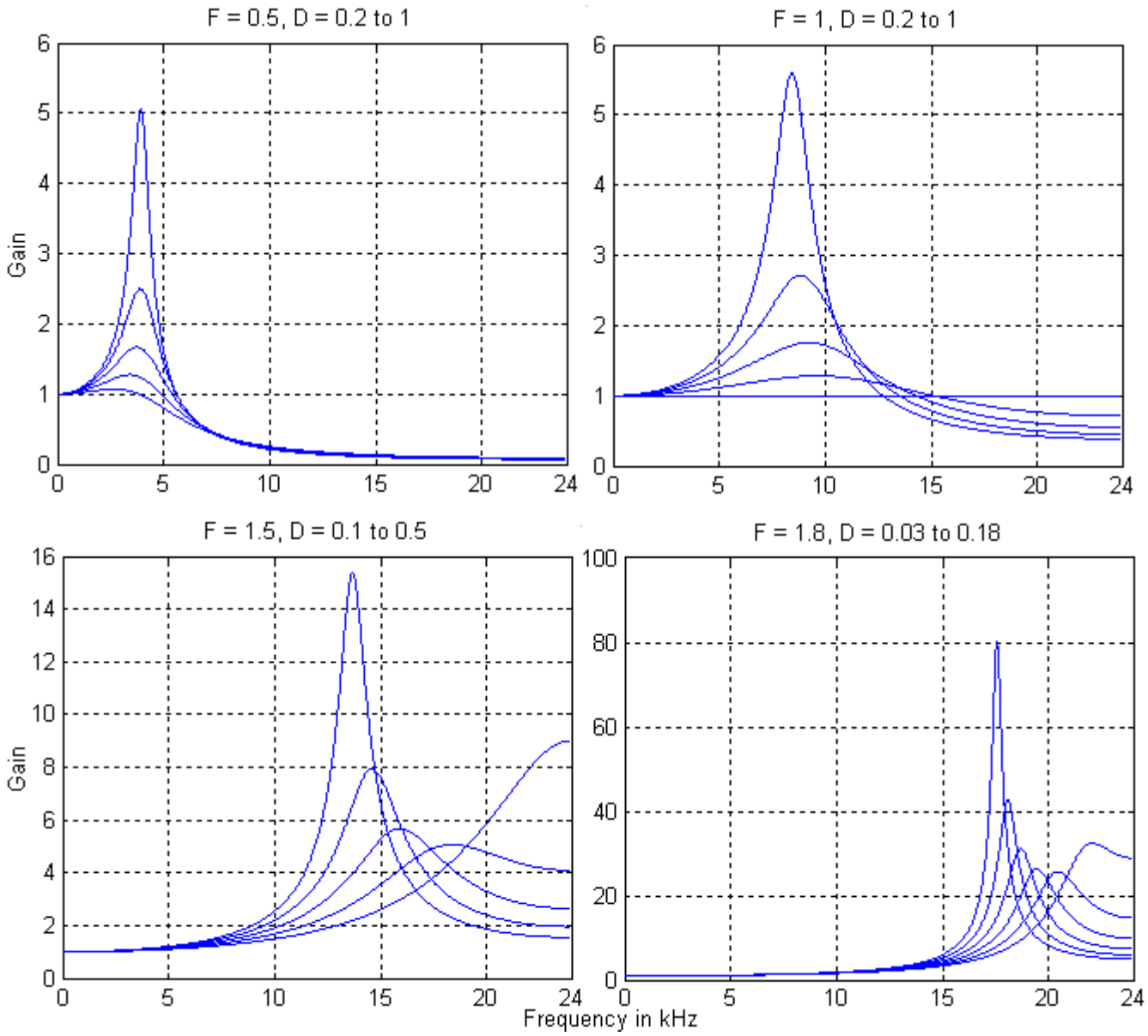


Fig. 4: Magnitude Response of the Chamberlin Lowpass Filter for Varying D

We observe a perfectly flat magnitude response for F = D = 1. It can be concluded from the transfer function that it's the only value pair with that property.

The most prominent deviation from ideal behavior is found for large D, where the filter starts to peak around $f_s/2$ and finally goes unstable with increasing F while still acting as a lowpass. Simple designs circumvent the problem by clamping D to values below 1, but unfortunately, their inability to operate at low Q makes them unattractive for non-electronic timbres.

A promising approach is to progressively reduce the maximum value of D with increasing F. This also helps to improve the effectiveness of the filter at high F. Aiming at a flat magnitude response for $D \geq 1$ at maximum cutoff frequency, we choose $F_{max} = 1$, $D_{max} = 2$, and calculate the filter coefficient D from the new control variables D_c and F_c as follows:

$$D = \min(D_c, 2 - F_c) \qquad ; 0 < F_c \leq 1, 0 < D_c \leq 2$$

This leaves a safety margin of 1 in the stability criterion and yields the result shown in Fig. 5.

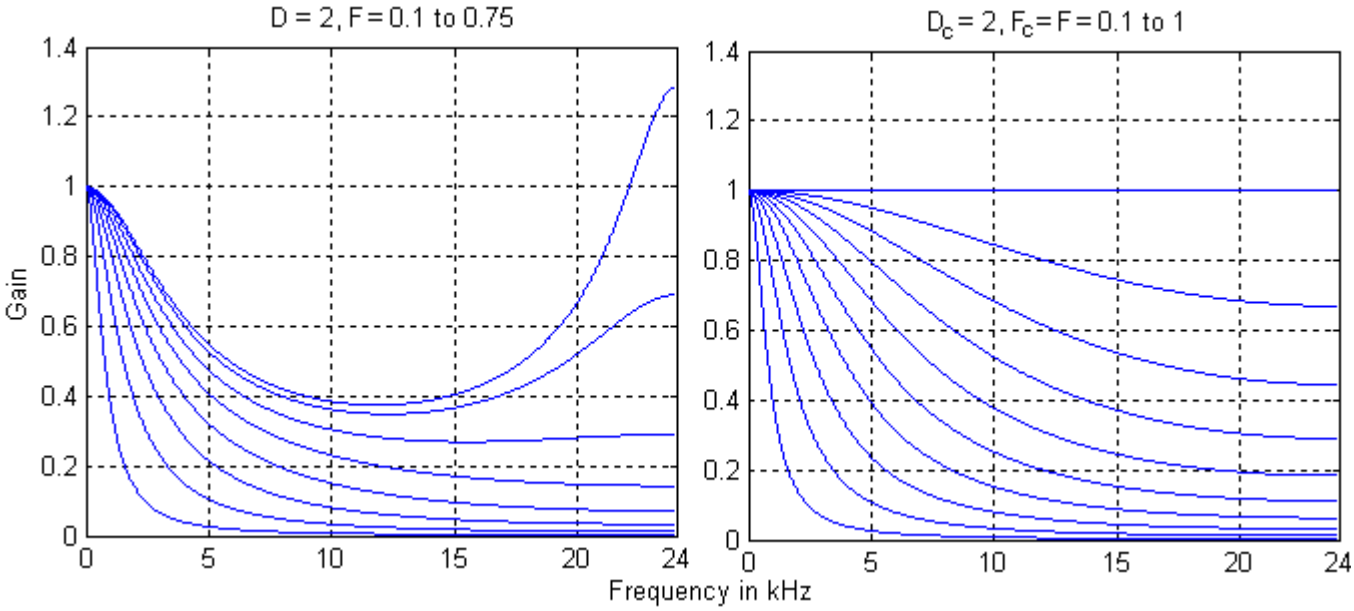


Fig. 5: Magnitude Response affected by D Correction (Left: Original, Right: Corrected)

A significant musical flaw is the decreasing cutoff frequency when Q in increased while F is held constant. We deduce from Fig. 4 that the effect becomes more pronounced at high values of F and D. A heuristic approach, along with the side condition of transparency for $D_c \geq 1$ at maximum F_c, yields the improved filter coefficient F for the frequency control variable F_c:

$$F = F_c(1.85 - 0.85DF_c) \qquad ; 0 < F_c \leq 1$$

The value of 1.85 is a trade-off between maximum cutoff frequency and excess peak gain. The resulting filter is stable for any combination of $0 < D_c \leq 2$ and $0 < F_c \leq 1$ with a worst case stability margin of about 0.5. Because the linear filter has no means for amplitude compression, $D_c \geq 0.2$ is recommended in order to maintain smooth frequency sweeps and limit dynamic peaking (see section 3.5.3). In consequence, the corrected Chamberlin filter in its current form is adequate predominantly for basic filtering tasks, for example in sample players. Fig. 6-8 give an impression of the performance.

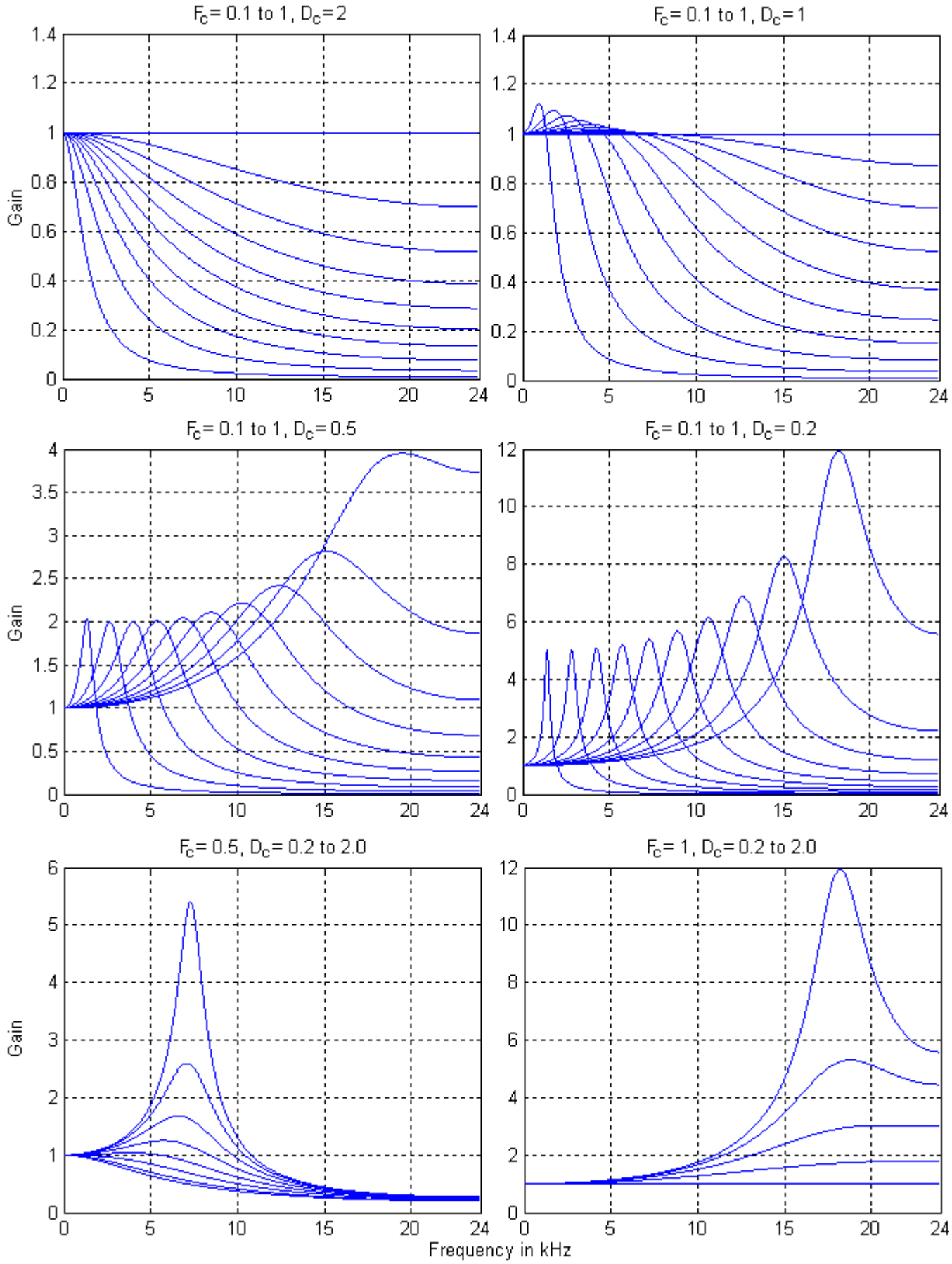


Fig. 6: Magnitude Response of the Corrected Chamberlin Lowpass Filter

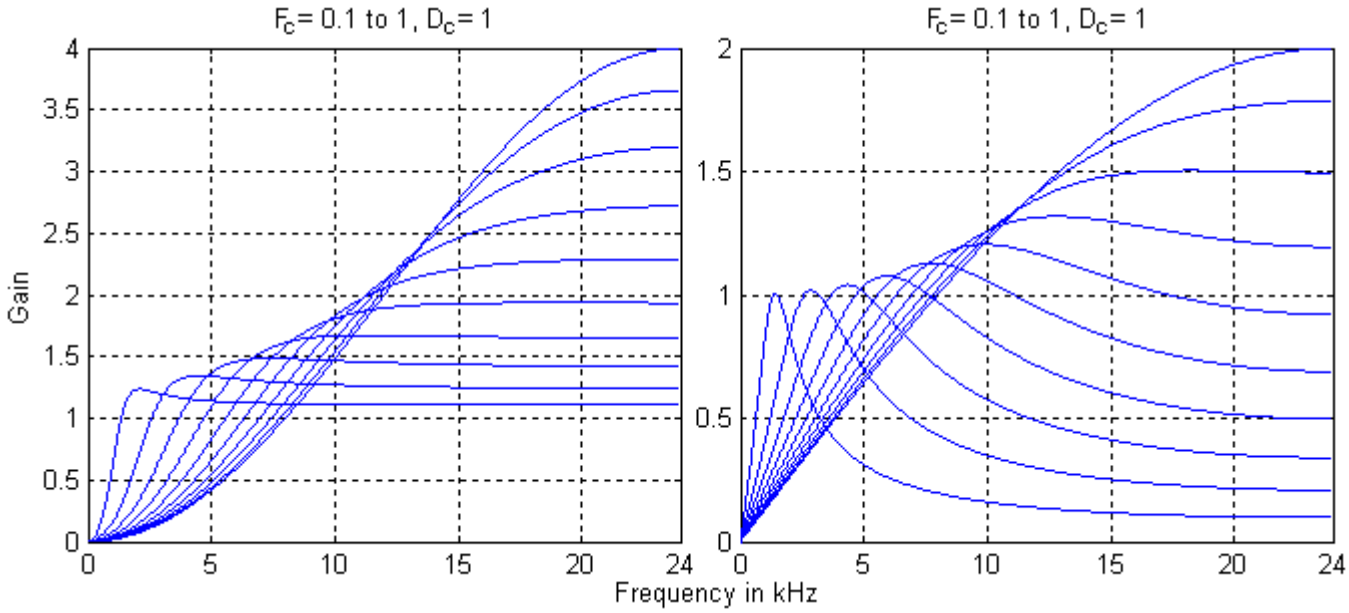


Fig. 7: Magnitude Response of the Corrected Chamberlin High and Band Pass Filters

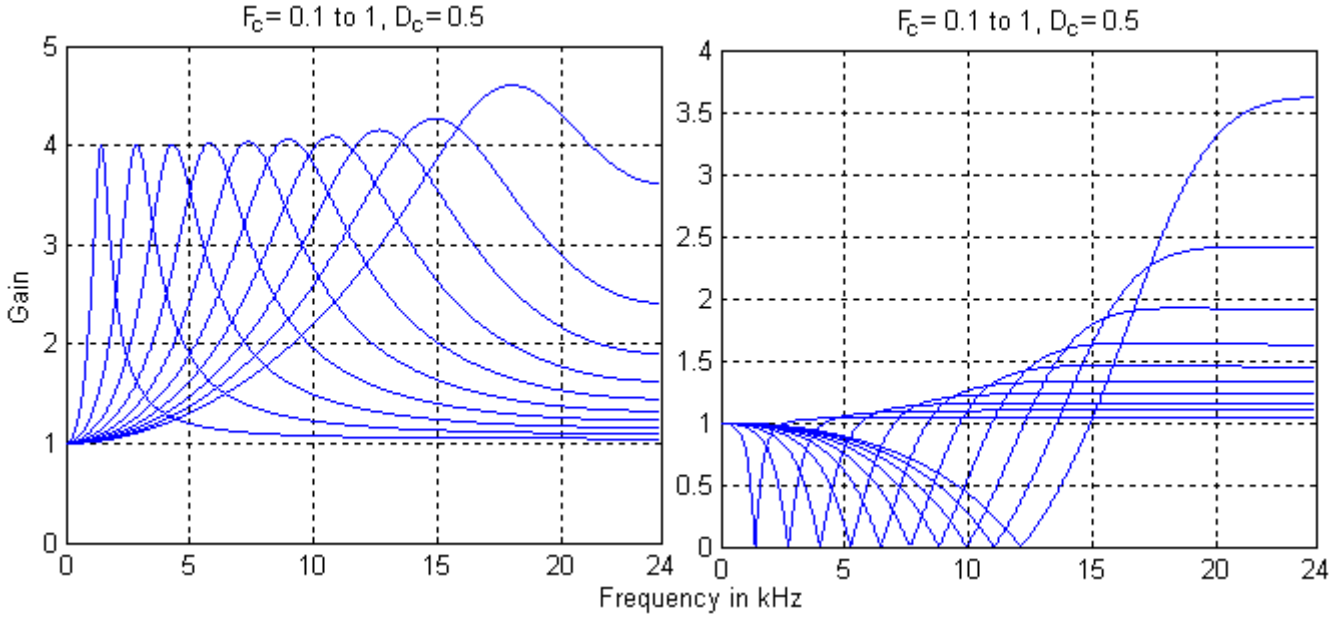


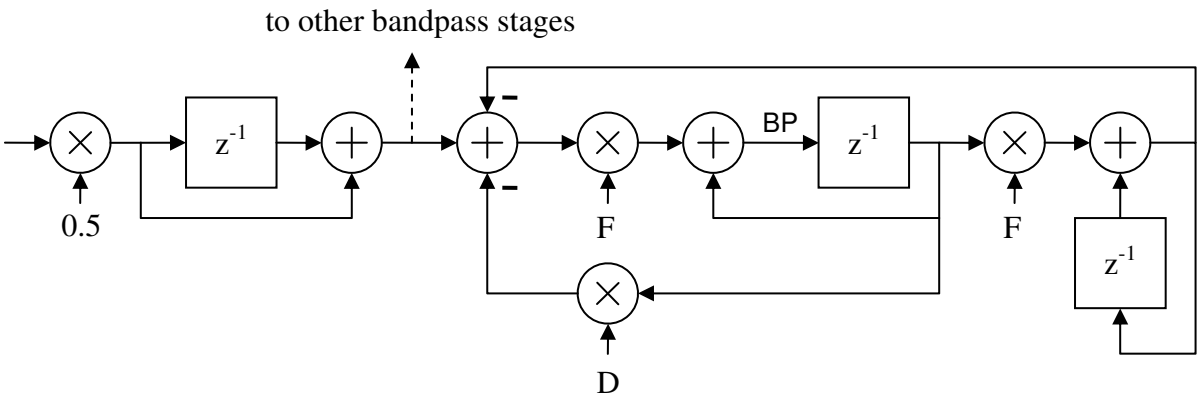
Fig. 8: Magnitude Response of the Corrected Chamberlin Peaking and Notch Filters

The excess gain in the top octave is perceived as pleasant rather than deficient in various applications. Moreover, we could add further corrections, of which only the bandpass and peaking cases will be discussed, since today's prevalence of single-cycle multipliers makes the oversampled Chamberlin filter the preferred choice in most instances (see section 3.3).

Many synthesizers update the filter coefficients at a comparably slow control rate and only interpolate between consecutive sets at audio rate. In this case, both F and D corrections can be moved to the control update stage to boost computational efficiency.

The bandpass and peaking types are interesting low-noise building blocks for time-varying formant and vocoder filter banks because the center frequency is restricted to lower values, Q is on the high side, and consequently, both corrections are omitted.

A nice extension that may be shared among multiple filters perfectly eliminates the excess gain at high center frequencies in the bandpass [7]. (Fig. 9 and 10)



$$F = F_c(1.85 - 0.75DF_c) \qquad ; 0 < F_c \leq 1, D \leq 1$$

Fig. 9: Corrected Chamberlin Bandpass Filter with High F_c Equalization

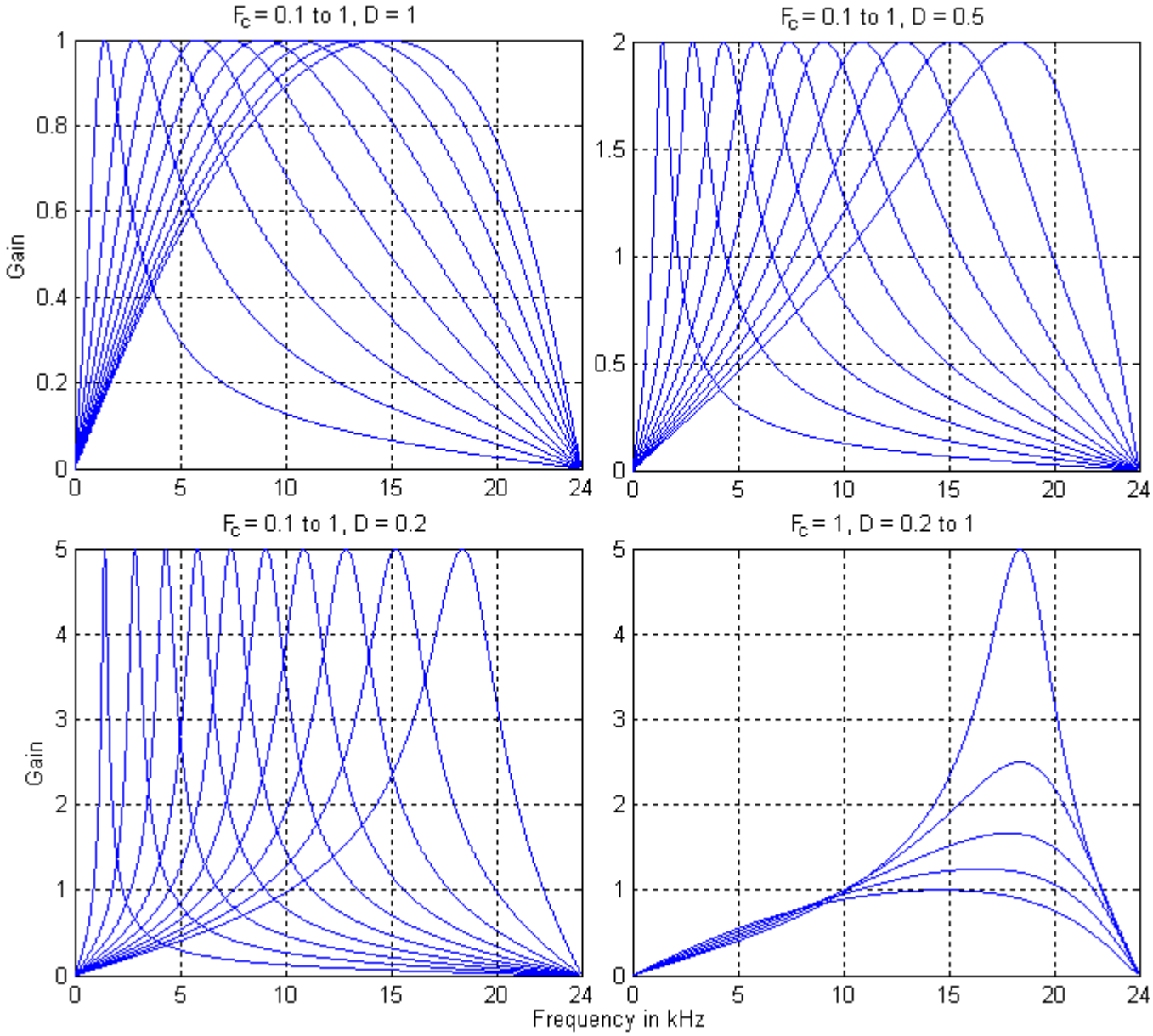


Fig. 10: Magnitude Response of the Corrected Chamberlin Bandpass Filter with High F_c Equalization

The equalized bandpass has the transfer function $H_{BPeq}(z) = \dfrac{0.5F(z^2-1)}{z^2+(F^2+DF-2)z+(1-DF)}$.

While the simple peaking filter inverts components above the center frequency and develops slight excess peaking for D > 0.5, an alternative filter created by summing the equalized bandpass signal with the input avoids these issues (Fig. 11). It also offers more flexibility because the balance of the mix may be tailored to the application with little additional effort. We also see how the equalization is merged into the filter circuitry saving a memory operation and providing additional headroom (as long as the signal energy is concentrated below $f_s/3$).

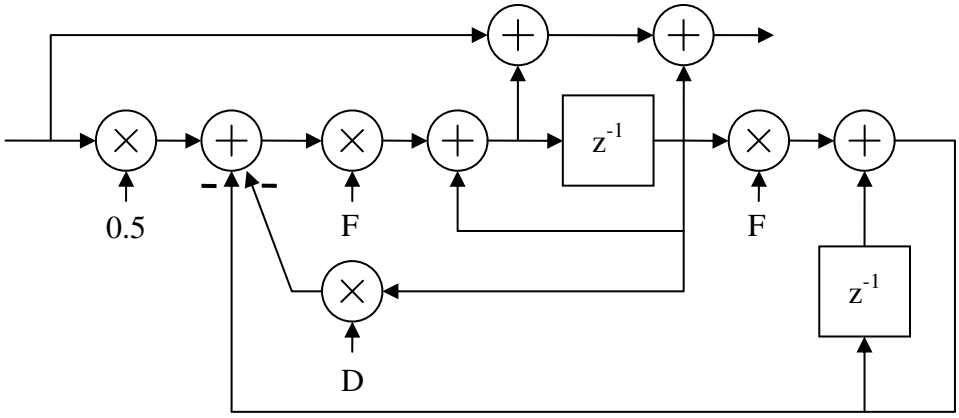


Fig. 11: Corrected Chamberlin Peaking Filter without Phase Inversion

Its transfer function is: $H_{PKeq}(z) = 1 + \dfrac{0.5F(z^2-1)}{z^2+(F^2+DF-2)z+(1-DF)}$

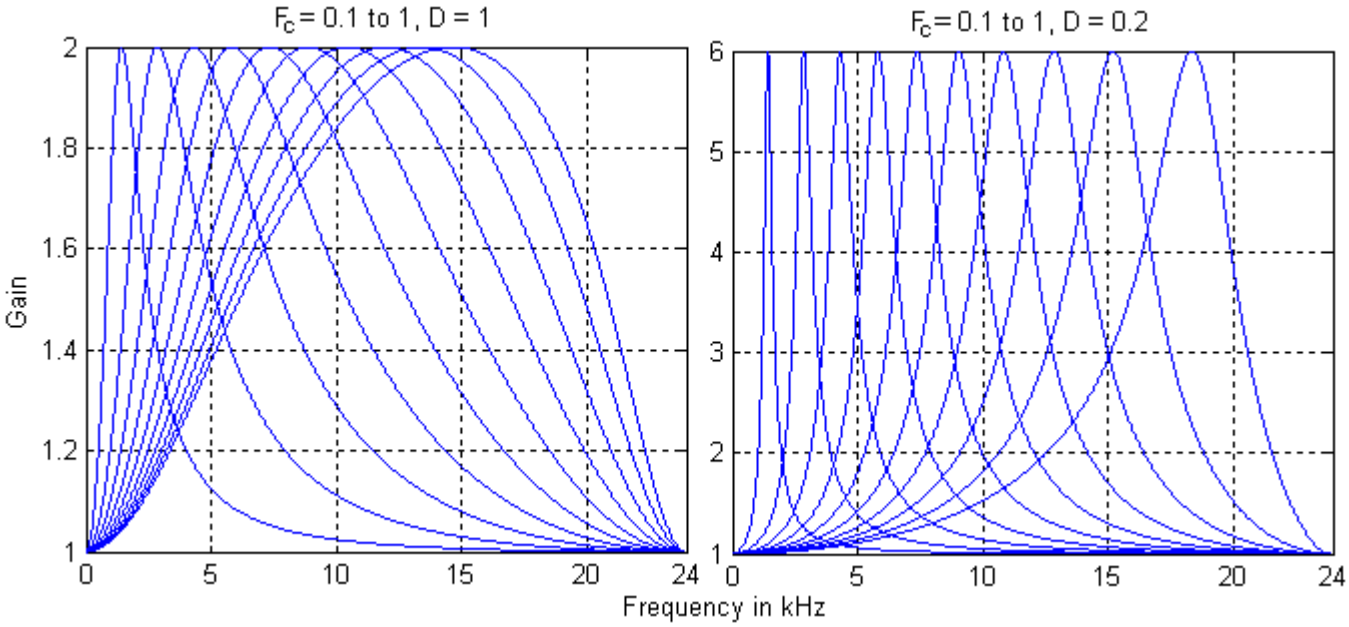


Fig. 12: Magnitude Response of the Corrected Chamberlin Peaking Filter without Phase Inversion

After some graph reordering, a minimum-phase version of the Chamberlin filter is obtained (Fig. 13). This compact form of a structure discussed in [7] has the same band and high pass transfer function as the original Chamberlin filter, whereas the lowpass differs only in that it got rid of the superfluous unit delay. The notch output N is accessible by reading the highpass node before the lowpass signal is subtracted - a little trick that works in the original as well.

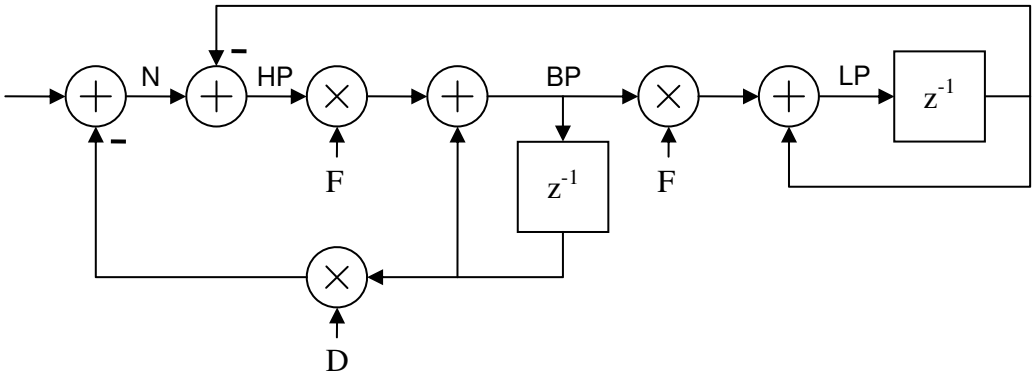


Fig. 13: Minimum-Phase Chamberlin Filter

A simple peaking filter can be constructed by taking the difference of the low and high pass output. We conclude from the transfer function

$$H_{PKmp}(z) = H_{LPmp}(z) - H_{HPmp}(z) = \frac{F^2 z^2 - (z-1)^2}{z^2 + (F^2 + DF - 2)z + (1 - DF)}$$

that it exhibits the same magnitude but a different phase response compared to the original.

It mainly depends on the pipeline delays of the hardware whether the minimum-phase filter runs faster or slower than the original. In contrary to certain rumors, both versions perform equally well with time-varying coefficients.

We conclude the section about the non-oversampled Chamberlin filter with some remarks on its behavior under time-varying conditions. The filter coefficient F has to be large in order to achieve a reasonably high maximum cutoff frequency at high Q, but as a consequence, a single coefficient change can result in an overshoot exceeding 180% at the lowpass output (compared to only 50% in the oversampled version of section 3.3). This is a major issue with virtual analog synthesizers where the whole cutoff frequency range is swept within a fraction of a millisecond. It's highly undesirable to provide additional headroom for the emerging transients due to the negative effect this would have on the perceived punch of the sound. Moreover, the limiting nonlinearities of the VA filter do not significantly suppress these transients either, because without oversampling, they have to be bandlimited and therefore are unable to react quickly enough.

As a rule of thumb, we state that whenever the non-oversampled Chamberlin structure is considered as a synthesizer filter with $F > 1.25$, a listening test should be carried out to check the dynamic properties at the fastest rate of change the user may set. On double rate systems ($f_s \geq 88.2$ kHz), the non-oversampled version is recommended without restrictions, since F remains below 1.25 for any cutoff frequency in the audio range.

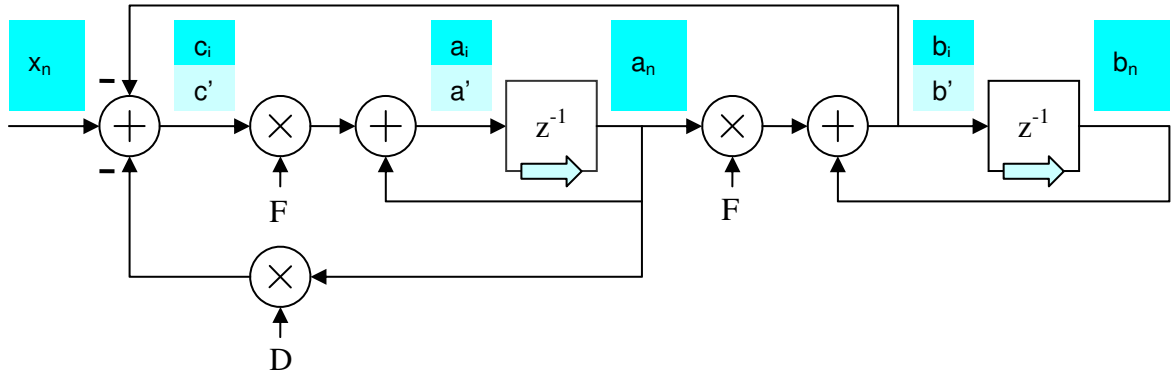
3.3 Oversampled Chamberlin Filter

The efficiency of the Chamberlin filter and its close approximation of the continuous time prototype for cutoff frequencies up to $f_s/6$ encourage oversampling by a factor of two. This won't double the computational effort because the coefficients are updated only once and the state variables are already in the processor's registers for the second pass.

Advantages:

- Decreased excess peaking at high cutoff frequencies.
- Improved effectiveness and response fidelity of all filter types. This is especially beneficial for the highpass.
- The required range of F is reduced to the point where the filter always remains stable when D is moved towards 1 without an accompanying correction of F. This is vital for certain amplitude compression techniques in virtual analog filters.
- Extended cutoff frequency range beyond 20 kHz.
- When the coefficients are modulated quickly, artifacts from state mismatch, like dynamic peaking and transients, diminish close to the level of the analog original. Although rarely discussed, the influence on the perceived filter quality is substantial.
- If extended with nonlinearities to build a virtual analog filter, less aliasing occurs.

The oversampled structure of Fig. 14 will be examined without multirate analysis. Therefore, we point out the values in the two passes of the calculation process instead of cluttering the diagram with upsampling and decimation blocks. The input enters in both passes, which strongly reduces frequency components around Nyquist immediately after upsampling and is essential if nonlinear stages are employed (see section 3.4). Fortunately, it does not prevent the filter from becoming perfectly transparent as the frequency response is superimposed by a shifted version in the inherent downsampling process when the output is formed.



1. $b_i = b_n + Fa_n$
   $c_i = x_n - b_i - Da_n$
   $a_i = a_n + Fc_i$
   $b' = b_i + Fa_i$
   $c' = x_n - b' - Da_i$
   $a' = a_i + Fc'$
2. Update state variables: $a' \rightarrow a_{n+1}$, $b' \rightarrow b_{n+1}$
3. (Update filter coefficients F and D)

Fig. 14: Two-Fold Oversampled Chamberlin Filter with Update Sequence

We obtain the transfer functions by following the path from a_n to a_{n+1} and b_n to b_{n+1}:

$$\Delta = z^2 + \left(4F^2 - F^4 - 2DF^3 - D^2F^2 + 2DF - 2\right)z + (1 - DF)^2$$

$$H_{x\to a} = \frac{F\left(2 - DF - F^2\right)(z-1)}{\Delta} \qquad H_{x\to b} = \frac{F^2\left(z + 3 - 2DF - F^2\right)}{\Delta}$$

Further:

$$H_{x\to a'} = \frac{F\left(2 - DF - F^2\right)\left(z^2 - z\right)}{\Delta} \qquad H_{x\to ai} = \frac{F\left(z^2 - DFz + (DF - 1)\right)}{\Delta}$$

$$H_{x\to b'} = \frac{F^2 z\left(z + 3 - 2DF - F^2\right)}{\Delta} \qquad H_{x\to bi} = \frac{F^2\left(\left(3 - DF - F^2\right)z + (1 - DF)\right)}{\Delta}$$

$$H_{x\to ci} = \frac{z^2 + \left(F^2 - 2\right)z + \left(1 - F^2\right)}{\Delta} \qquad H_{x\to c'} = \frac{\left(1 - DF - F^2\right)z^2 + \left(F^2 + 2DF - 2\right)z + (1 - DF)}{\Delta}$$

Selecting different output combinations for smooth frequency sweeps, we find:

Type	Output	Transfer Function
Lowpass	b_i	$\frac{F^2\left(\left(3 - DF - F^2\right)z + (1 - DF)\right)}{\Delta}$
Bandpass 1	2a’	$\frac{2F\left(2 - DF - F^2\right)\left(z^2 - z\right)}{\Delta}$
Bandpass 2	a’ + a_i	$\frac{F\left[\left(3 - DF - F^2\right)z^2 + \left(F^2 - 2\right)z + (DF - 1)\right]}{\Delta}$
Highpass	(c’ + c_i)/2	$\frac{\left(2 - DF - F^2\right)(z-1)^2}{2\Delta}$
Peaking	b’ - c_i	$\frac{\left(F^2 - 1\right)z^2 + \left(2 - 2DF^3 + 2F^2 - F^4\right)z + \left(F^2 - 1\right)}{\Delta}$
Notch	b’ + c’	$\frac{(1 - DF)z^2 + \left(4F^2 - 2DF^3 + 2DF - F^4 - 2\right)z + (1 - DF)}{\Delta}$

These filters have desirable properties for the neutral setting F = D = 1. The lowpass, peaking, and notch types become transparent and the highpass and bandpass 1 types perfectly mute. If complete muting is deprecated for musical reasons, bandpass 2 is a good alternative. To get a minimum-phase lowpass, we may calculate and output the new b_i at the end of the update sequence. This would not add to computational costs as it eliminates the calculation of b_i in the next sample. The peaking filter inverts components above the center frequency creating musically interesting cancellation effects when running in parallel with other filters.

The limiting scheme remains D = min(D_c, 2-F_c), but the frequency correction is modified to:

$$F = F_c(1.22 - 0.22DF_c) \qquad ; 0 < F_c \le 1$$

This yields $f_{c(max)} \approx 20$ kHz for $f_s = 48$ kHz. The magnitude responses are shown in Fig. 15-19.

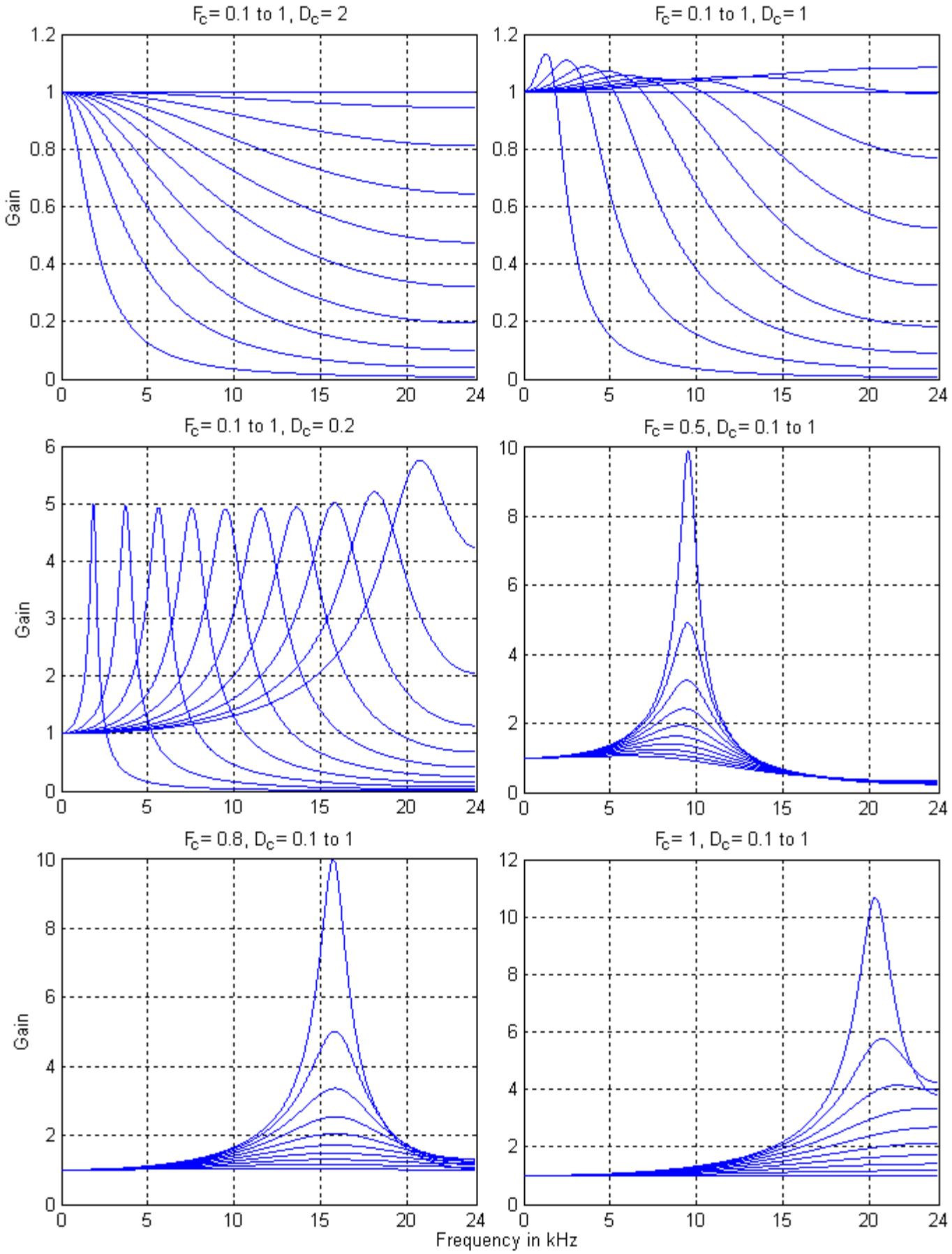


Fig. 15: Magnitude Response of the Corrected Two-Fold Oversampled Chamberlin Lowpass Filter

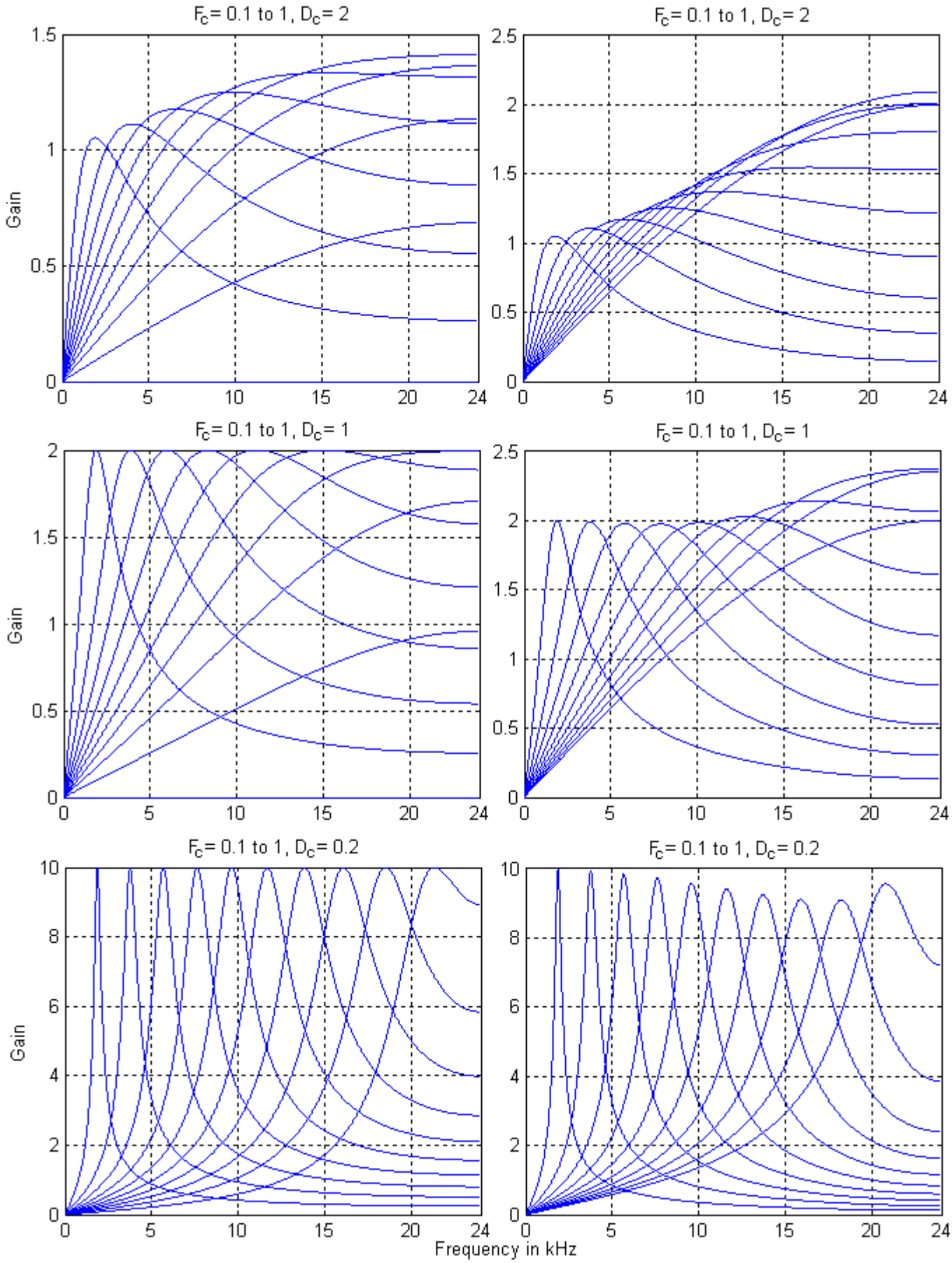


Fig. 16: Magnitude Response of the Corrected Two-Fold Oversampled Chamberlin Bandpass Filter (Left: Version 1, Right: Version 2)

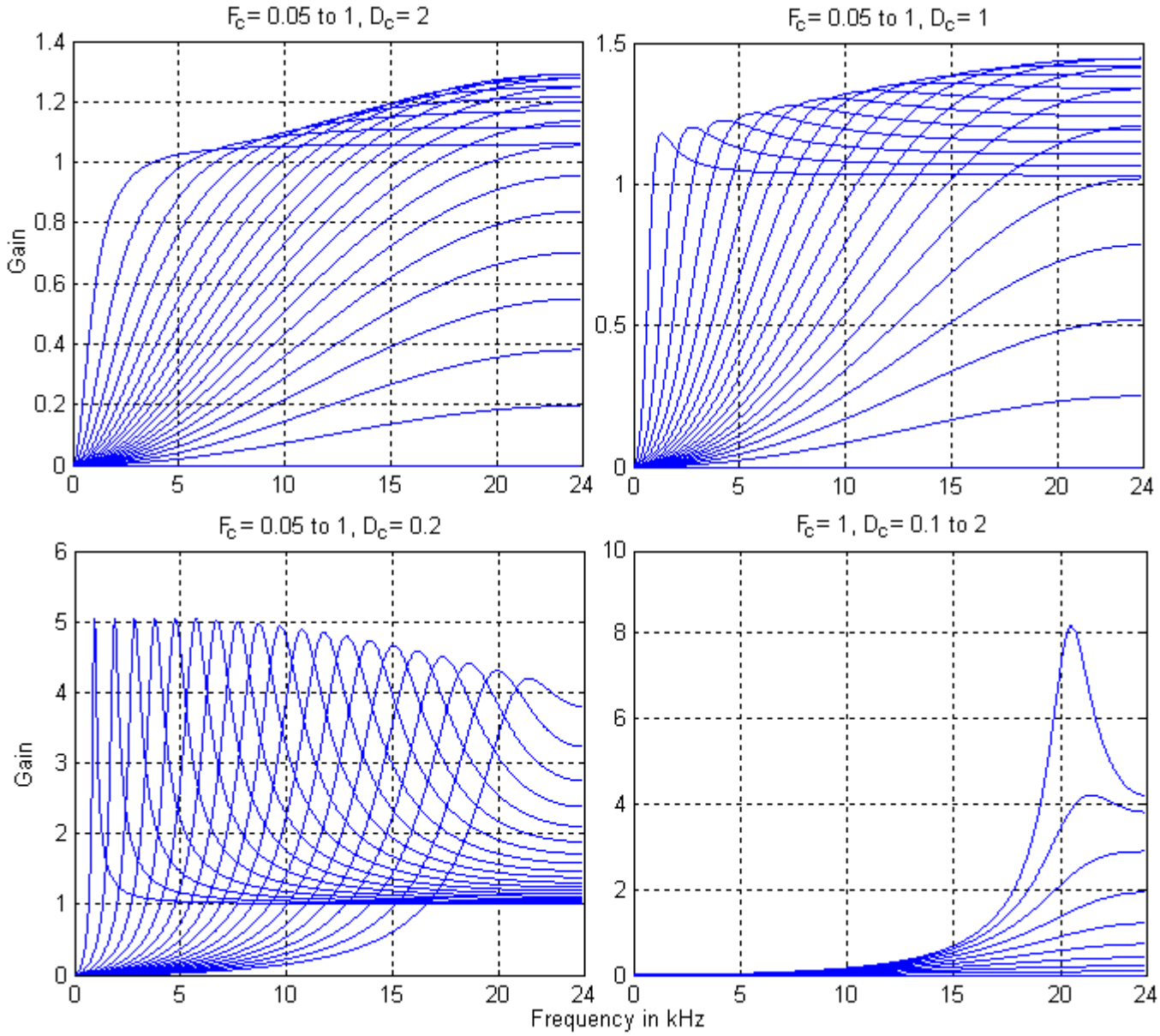


Fig. 17: Magnitude Response of the Corrected Two-Fold Oversampled Chamberlin Highpass Filter

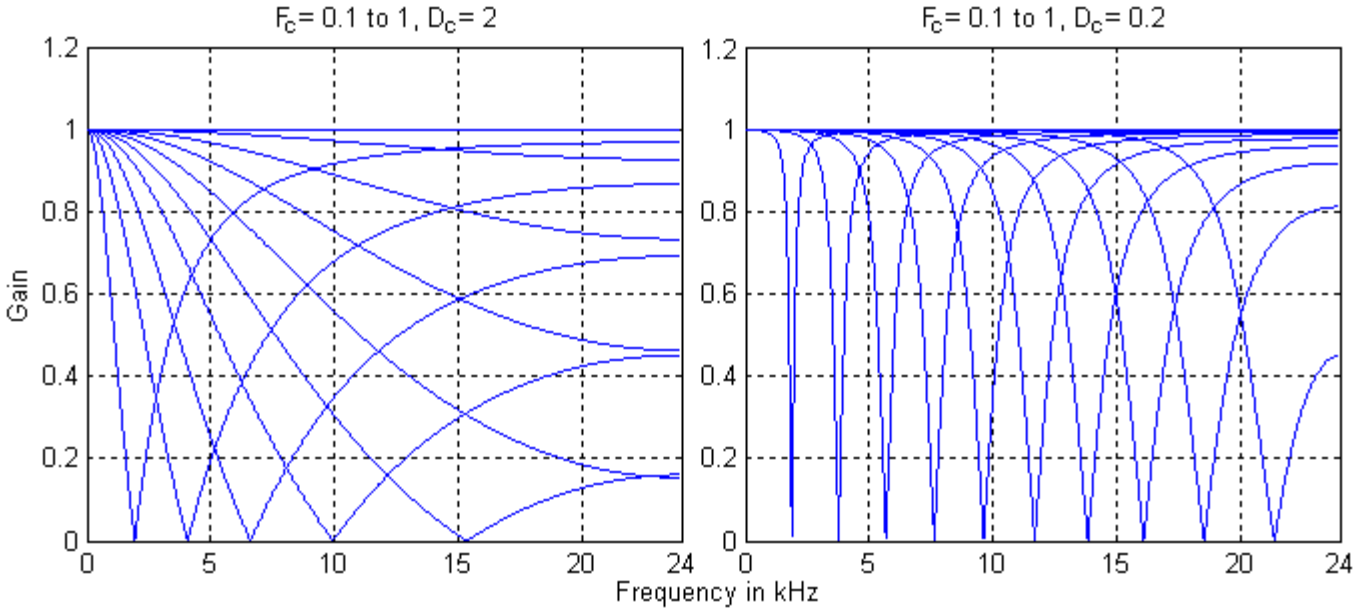


Fig. 18: Magnitude Response of the Corrected Two-Fold Oversampled Chamberlin Notch Filter

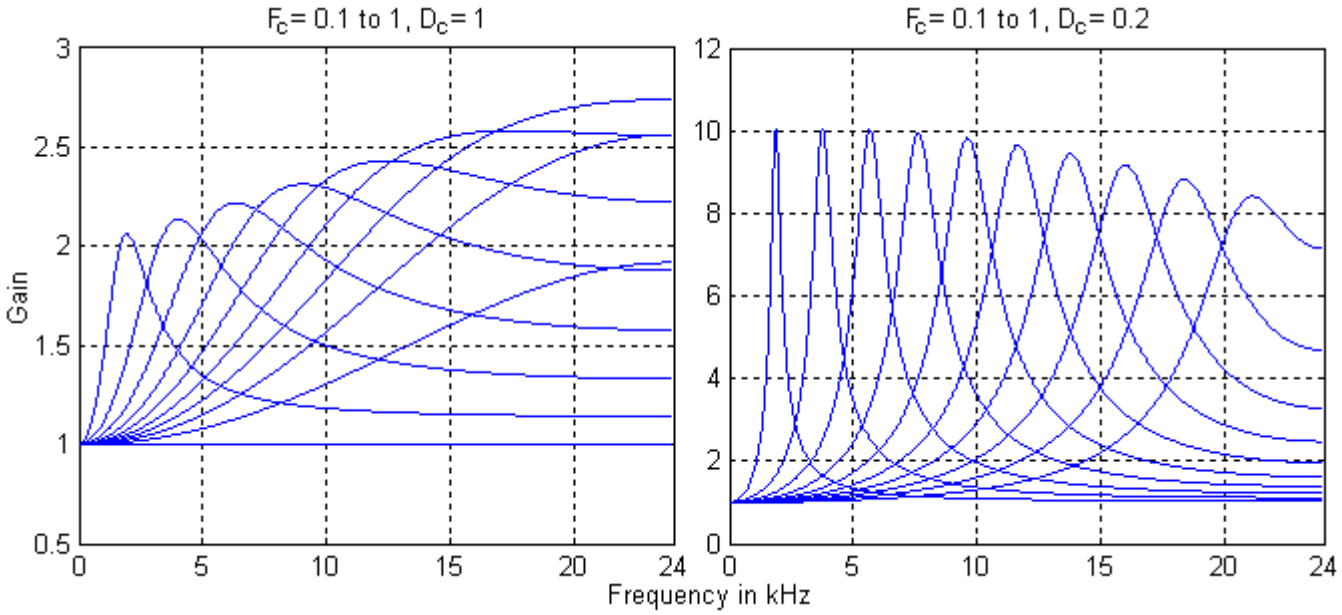


Fig. 19: Magnitude Response of the Corrected Two-Fold Oversampled Chamberlin Peaking Filter

3.4 Bandlimited Saturation

Many electronic timbres involve frequency sweeps at high Q factors. With a linear filter, they would sound very irregular due to the amplitude peaking each time the center frequency hits a harmonic. This may even lead to hard clipping inside the filter and subsequent stages. Analog synthesizer filters exploit the soft saturation of certain stages within their circuitry to solve the problem, for example the hyperbolic tangent transfer characteristic of a differential amplifier based on bipolar transistors. Unfortunately, this approach does not translate easily into the digital domain as the required nonlinearity extends the bandwidth of the signal raising the risk of audible aliasing.

In a first attempt, we adopt the analog filter concept and replace the integrator at the bandpass node by a soft-saturating version (Fig. 20). Experiments with alternative locations of the nonlinearity, most notably the outer feedback paths, have been carried out in [2].

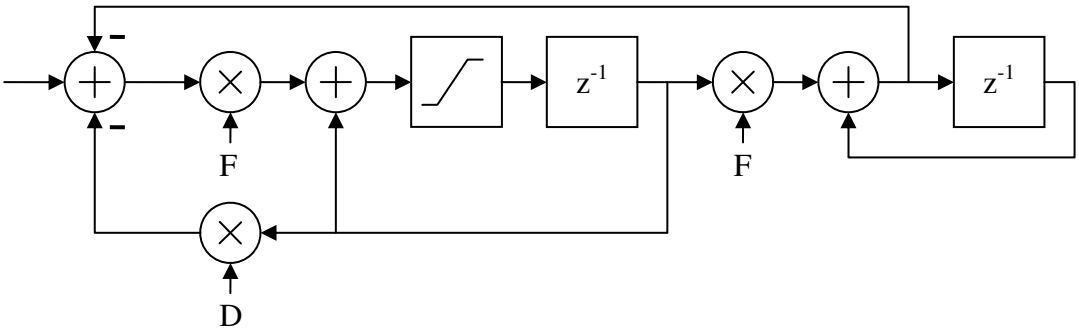


Fig. 20: Simple Saturating Chamberlin Filter

The soft-saturating function should be continuous, asymptotically linear with a gain of 1 around zero input, and have zero first derivatives at the points of saturation. If the original signal x contains frequency components up to f_{max}, a polynomial f(x) of degree N will produce an output spectrum limited to Nf_{max} as long as no saturation occurs. An unbiased input generates an unbiased output if the polynomial consists of odd terms only. The simplest function that satisfies all of the above conditions is:

$$f(x) = \begin{cases} 2/3 & ;\ x > 1 \\ x - x^3/3 & ;\ |x| \le 1 \\ -2/3 & ;\ x < -1 \end{cases}$$

In practice, we observe that a two-fold oversampled filter according to Fig. 20 is only useful for a weak input over the entire audio range, because the bandpass signal is not sufficiently bandlimited at high center frequencies. To obtain smooth sweeps on par with analog filters, we had to drive the circuit so hard that a lot of objectionable aliasing would occur.

In order to avoid excessive oversampling, we have to find a way to decouple the saturation mechanism from the high frequency audio content. This is achieved in the structure of Fig. 21 wherein a limiting power detector drives D towards the safe value of 1 with increasing amplitude. The time constant is chosen in a way that the system responds quickly enough to amplitude changes without introducing audible aliasing. A value of 0.5 to 1 millisecond is recommended, which corresponds to $\lambda = 1000..2000/f_s$. The limiter can be of a brickwall type.

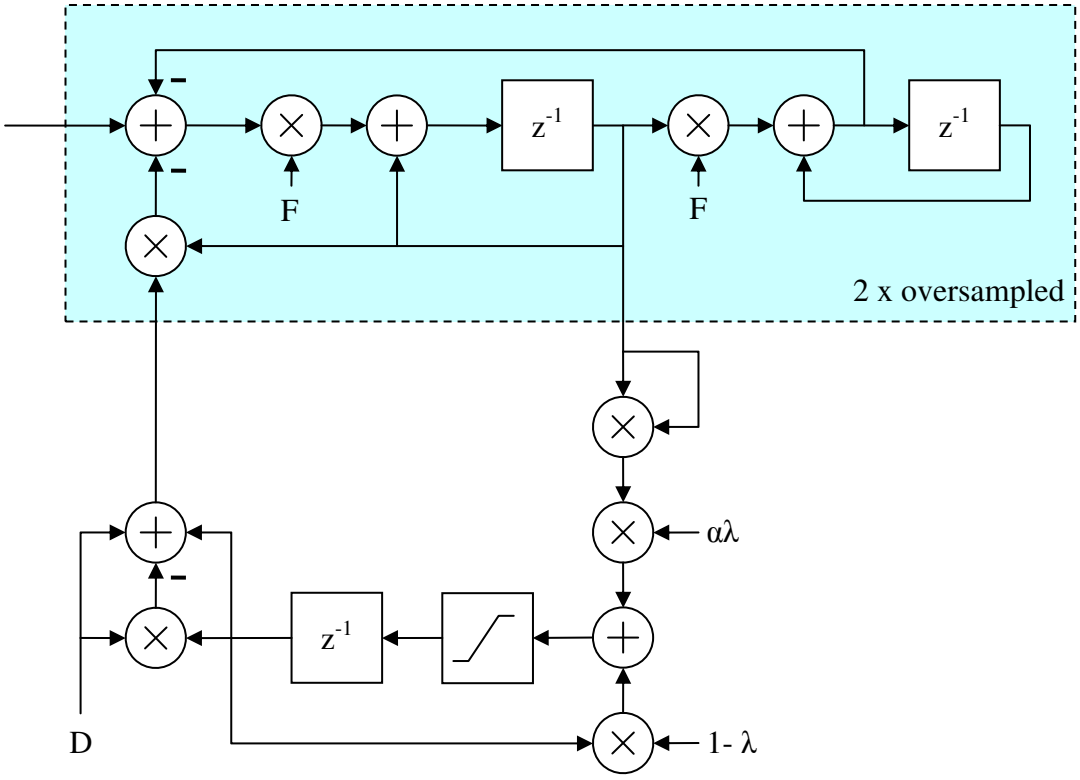


Fig. 21: Oversampled Chamberlin Filter with Bandlimited Saturation

Only the audio part is oversampled by a factor of two according to Fig. 14, whereas the auxiliary circuit runs at the normal rate and is fed with a_n.

The gain α of the auxiliary path requires some adjustment to yield optimally smooth sweeps without triggering transient instabilities. For a system with a values range of ±1, a good starting point is α = 3 with a maximum input amplitude of 0.3 (peaking filter) to 0.45 (lowpass filter). Fast sweeps on a full-scale input waveform with a high crest factor, for example a band-filtered sawtooth, have proven reliable as test signals in this optimization step. As a guideline, the output level is about twice as high at $D \approx 0$ compared to D = 1.

This filter is capable of self oscillation for $D < 0$.

If two of these filters are wired in series, D should exceed 0.2 in the first stage to prevent interference in the auxiliary circuit. A good sounding and efficient multi-type filter can be made of a linear Chamberlin filter with $D > 0.5$ followed by a saturating Chamberlin filter.

The filter in Fig. 21 sounds smooth. For a more expressive saturation, a consecutive nonlinear element may be attached. Since the filter effectively limits the output amplitude, a simple scheme suffices in the low and band pass cases: A sigmoidal function f(x) is chosen and f(γx)/γ generated, whereby x is the filter output and γ a factor that is gradually reduced when the cutoff frequency approaches the high end. If we intend to drive the nonlinearity harder, βf(γx) is preferable, where γ starts to decrease earlier and β is a parameter that tracks the cutoff frequency in a user-adjustable way. Thus, an unpleasant amplitude boost at high cutoff frequencies is avoided.

Controlling a gain parameter by a power detector to introduce a bandlimited nonlinearity has a long history in audio and RF design. Despite its universality and some subtleties, only few publications discuss this technique in the context of synthesizers (e.g. by Stilson/Thornburg), hence applying it usually involves some experimentation and acid tests.

3.5 Parameter Update

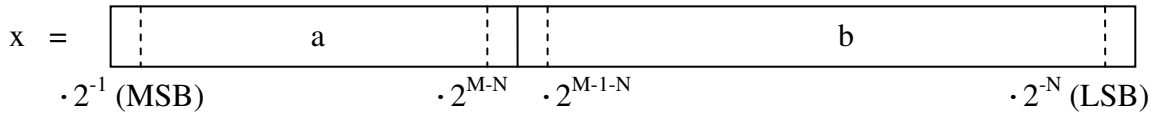
Whereas setting the Q factor is straightforward, musical constraints force us to map the user-side frequency control parameter exponentially to a typical cutoff frequency range of 5 to 20000 Hz. Furthermore, a precision of at least 0.1% should be maintained up to 4 kHz at the point of self oscillation if we intend to use the filter as a tuned oscillator. Digital filter coefficients are trigonometric expressions of the cutoff frequency that can be approximated well by simple polynomials in the lower range. One approach to efficient control is to combine the trigonometric and the exponential part, store the exact mapping in a table, and interpolate. Alternatively, we may exponentiate first and feed the result into a polynomial that maps a linear scale to the final filter coefficients. An additional polynomial is occasionally required with either method to adjust Q according to the cutoff frequency in the upper range.

3.5.1 Fast Exponentials

If fast multiplication is available, the interpolation method performs well. It also works with non-exponential functions. For the function $f(x) = \alpha\exp(\lambda x)$ with $0 \le x \le 1$, a table with $N >> \lambda$ entries, and linear interpolation, the maximum relative error becomes $\varepsilon_{rel} \approx \frac{\lambda^2}{8N^2}$.

For the conditions mentioned in the introduction, we obtain $N \ge 93$.

The fractional binary representation of the input x = [0, 1) opens some interesting alternatives for exponentiation on programmable logic and microcontrollers:



$$x = a \cdot 2^{M-N} + b \cdot 2^{-N}$$

$$y = y_{\min}\left[\frac{y_{\max}}{y_{\min}}\right]^{x} = y_{\min}\left[\frac{y_{\max}}{y_{\min}}\right]^{a \cdot 2^{M-N} + b \cdot 2^{-N}} = \left[\frac{y_{\max}}{y_{\min}}\right]^{a \cdot 2^{M-N}} \cdot y_{\min}\left[\frac{y_{\max}}{y_{\min}}\right]^{b \cdot 2^{-N}}$$

Case 1: $\left[\frac{y_{\max}}{y_{\min}}\right]^{2^{M-N}} = 2 \quad \rightarrow \quad y = 2^{a} \cdot y_{\min}\left[\frac{y_{\max}}{y_{\min}}\right]^{b \cdot 2^{-N}}$

The right part is read from a table and multiplication by the left part is done with a barrel shifter. No multipliers are required.

Example: M = 9, N = 13.
$y_{max}/y_{min} = 2^{16} = 65536$. Accuracy: ±0.067%. Table size: 512.

Case 2: N = 2M. Both parts are read from a table and conventionally multiplied.

Example: M = 6, N = 12. $y_{max}/y_{min} = 4000$.
Accuracy: ±0.1%. Table size: 64.

The accuracy is limited in that the input must be rounded to fit the low-resolution binary representation. Hence, b deviates up to 0.5 from the exact value. Another consequence is discrete output steps with a continuous input.

3.5.2 Interpolating Exponentials

For efficiency reasons, especially when table lookups are expensive, exponentiation at audio rate f_s may be impractical. Instead, it is done at a slower control rate $f_c = f_s/N$ with N integer whereas the values are just interpolated at f_s. In practice, we find that linear interpolation sounds good up to an octave between data points. As virtual analog filters should be able to sweep through several decades within a millisecond, N will be low with consequent high control overhead. Thus, we'd better look for more specific interpolators.

A natural approach comes from the constant multiplication rate property of the exponential. Given an input that proceeds from a to b in N steps, the desired output y travels from e^a to e^b according to the following algorithm:

Initial values: $y[0] = e^a$, $\lambda = e^{(b-a)/N}$, calculated at f_c.
Update: $y[n+1] = \lambda y[n]$, calculated at f_s.

The output is a perfect exponential, but λ has to be precise to avoid a perceivable step when the new initial output value is set. Since the error accumulates in the same direction, the accuracy of λ should be N times higher compared to an audio rate update. This method usually performs well for modified exponentials $y = f(e^x)$ with $f(z) \approx z$ too. In this case, $\ln(f(e^x))$ is tabulated and the initial values are calculated as:

$$y[0] = f\left(e^a\right) \qquad \gamma = \frac{1}{N}\left[\ln\left(f\left(e^b\right)\right) - \ln\left(f\left(e^a\right)\right)\right] \qquad \lambda = e^{\gamma}$$

A second scheme originates from the limes definition of the exponential $e^x = \lim_{n\to\infty}\left(1+\frac{x}{n}\right)^n$.

The expression suggests that the n-th power of a linearly progressing variable x exhibits an exponential-like shape for $x << n$. Therefore, we tabulate the function $e^{x/K}$ and act as follows to move nearly exponentially from where we are ($y[0] = x[0]^K$) to the next data point $y[N] = e^b$ in N steps:

Initial value: $\lambda = (e^{b/K} - x[0])/N$, calculated at f_c.
Update: $y[n] = x[n]^K$, $x[n+1] = x[n] + \lambda$, calculated at f_s.

The algorithm directly maps a given linear input to a reasonably good exponential, a property that comes in handy in various applications. Moreover, only the exact end point has to be known. This eliminates any discontinuities from rounding errors when the new initial value is set and makes the method perfectly suitable for modified exponentials. Another big plus is efficient and bandlimited audio modulation of the output by simply adding the modulating signal to x[n] before raising it to the power in the calculation of y[n].

Optimum efficiency is achieved for $K = 2^M$ with M integer, because in that case the power calculation reduces to M squaring operations. Fig. 22 shows the performance for various K.

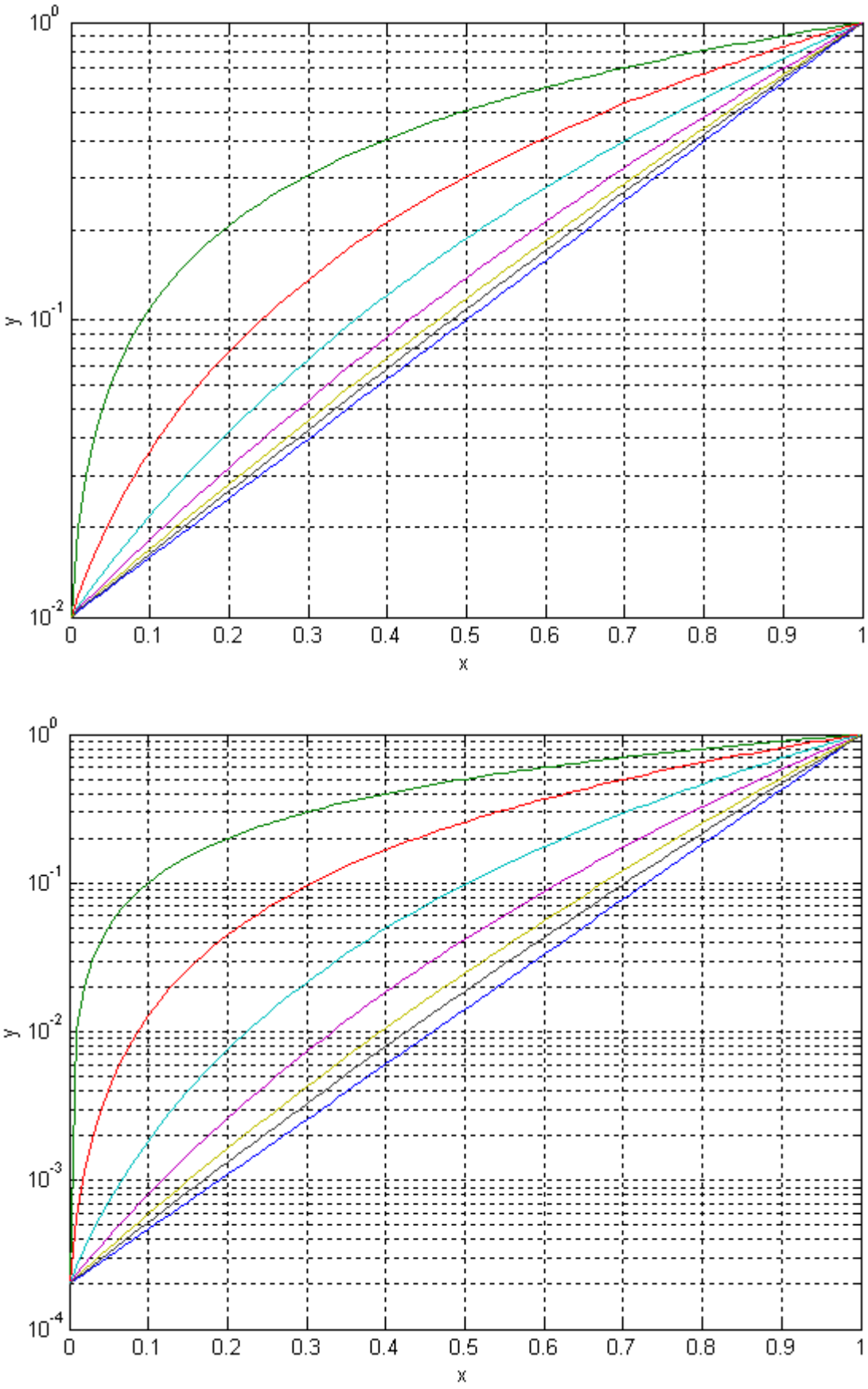
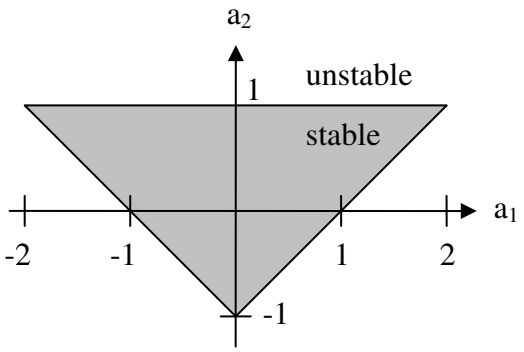


Fig. 22: Interpolated Exponential Function using Exact Values for x = 0 and x = 1 (Linear Interpolation, 2nd Method with K = 2,4,8,16,32, Ideal)

3.5.3 Stability of Time-Varying Filters

A causal discrete time LTI system is BIBO stable if all poles of its transfer function H(z) lie inside the unit circle. In consequence, a digital biquad filter is stable if the coefficients of its denominator polynomial lie inside a triangle (Fig. 23).



$$H(z) = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2}$$

Fig. 23: Stability Triangle of the Digital Biquad

When we move from one stable set to another by linearly interpolating the coefficients and updating them at once in each step, the trajectory becomes a straight line and the biquad never leaves the region of stability. An often preferred exponential or similar nonlinear sweep is accessible through the following procedure: The control parameters are updated according to the desired nonlinear law and mapped to the biquad coefficients at a lower control rate, whereas the coefficients are interpolated linearly at audio rate. This yields a piecewise linear trajectory that approximates the nonlinear one we would have obtained without interpolation.

Filters with high-level control parameters that determine the actual coefficients of the transfer function should be designed stable for any possible parameter set. A stability criterion is found with aid of the triangle or the Schur-Cohn test. Additional calculations at audio rate may be mandatory to modify the incoming parameter set in order to meet the criterion and to decouple the controls. If audio rate modulation of the parameters is not required, we may move these calculations to the control rate stage and interpolate between two stable parameter sets at audio rate. In this case, the stability of the intermediate filters has to be examined and ensured with respect to the applied interpolation scheme (maybe nonlinear, see section 3.5.2).

Until now, we assumed a time-varying filter to be stable if the time-invariant intermediate filters were stable, which is not true. Given a sinusoidal input, the coefficients determine the amplitude and phase relations between the state variables. For the same input but different coefficients, the current state would usually not occur. As a result, a decaying transient with time constants according to the new coefficient set is generated when they are changed. In case of a sweep towards lower center frequencies, the amplitude may build up successively and finally induce clipping. The same holds for continuous modulation of the center frequency (“Filter FM”) [3]. The intensity of the effect depends on the state-space description of the filter structure. The spectral norm of its transition matrix has the meaning of a worst case growth factor of the current state vector. Hence, a value below 1 ensures BIBO stability under arbitrary parameter modulation. Direct form realizations are so susceptible that they are hard to use as synthesizer filters, whereas the oversampled Chamberlin filter is robust. The Gold-Rader topology and first order filters with a single delay element are guaranteed stable. Virtual analog filters behave especially well in this regard due to their built-in amplitude compression, which is a versatile technique to keep fast modulated filters under control at the price of a more or less pronounced nonlinearity.

3.6 Noise Analysis

The quantized nature of digital signals leads to various errors in real-world filters. Important error sources are requantization steps in the processing hardware and coefficient quantization. The latter is not an issue in any of the synthesizer filters discussed in this book and will therefore be neglected.

For signal levels well above the quantization interval, the quantization process Q_n can be modeled as an independent source of uniformly distributed white noise. Assuming a full scale signal range of ±1 and a fixed point output with a word length of n bits, the variance of the noise signal e_n is $\sigma_n^2 = \frac{2^{-2n}}{3}$. A constant input to the quantizer does not produce any noise.

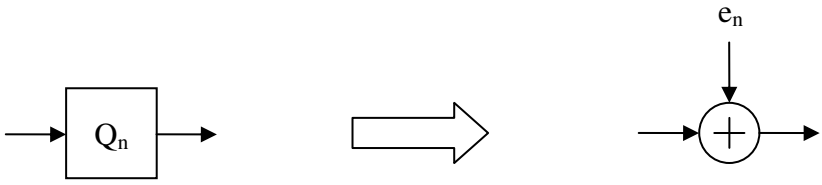


Fig. 24: Quantization Noise Model

To calculate the noise a given source injects into a certain node of the system, we treat the noise just like any other signal that gets filtered along the path from the source to the node. Thus, it is weighted by the transfer function $H_{src\rightarrow node}$ and the variance at the node results in:

$$\sigma_{node}^2 = \sigma_n^2 \frac{1}{\pi}\int_0^{\pi} \left|H_{src\rightarrow node}\left(e^{i\Omega}\right)\right|^2 d\Omega = \sigma_n^2 \mathrm{I}$$

It's always a good idea to plot the magnitude response of $H_{src\rightarrow node}$ in order to know in which frequency band the noise energy concentrates. Fortunately, the integral I has already been solved for stable first and second order systems:

$$H(z) = \frac{b_0 z + b_1}{z + a_1} \qquad I = \frac{b_0^2 + b_1^2 - 2b_0 b_1 a_1}{1 - a_1^2}$$

$$H(z) = b_0 z + b_1 \qquad I = b_0^2 + b_1^2$$

$$H(z) = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2} \qquad I = \frac{\left(b_0^2 + b_1^2 + b_2^2\right)(1 + a_2) - 2b_1(b_0 + b_2)a_1 + 2b_0 b_2\left(a_1^2 - a_2(1 + a_2)\right)}{(1 - a_2)\left((1 + a_2)^2 - a_1^2\right)}$$

$$H(z) = \frac{b_0 z^2 + b_1 z + b_2}{z + a_1} \qquad I = \frac{\left(b_0^2 + b_1^2 + b_2^2\right) - 2(b_0 b_1 + b_1 b_2)a_1 + 2b_0 b_2 a_1^2}{1 - a_1^2}$$

$$H(z) = b_0 z^2 + b_1 z + b_2 \qquad I = b_0^2 + b_1^2 + b_2^2$$

Alternatively, the integration can be done numerically with aid of a technical computing tool or a rough guess is made by eyeballing the magnitude response.

Since all sources are unbiased, the noise power equals the variance, and the total noise power is obtained by summing the noise power every source m generates at the node of interest.

$$\sigma_{node(tot)}{}^2 = \sum_{m=0}^{M-1} \sigma_{node(m)}{}^2$$

Finally, the signal-to-noise-ratio in dB referenced to a full scale sinusoidal becomes:

$$SNR = 10\log_{10}\left[\frac{0.5}{\sigma_{node(tot)}{}^2}\right]$$

Example:
A first order lowpass filter with constant coefficients is realized in hardware using 18 x 18 bit multipliers with 36 bit accumulation and 24 bit data memory.

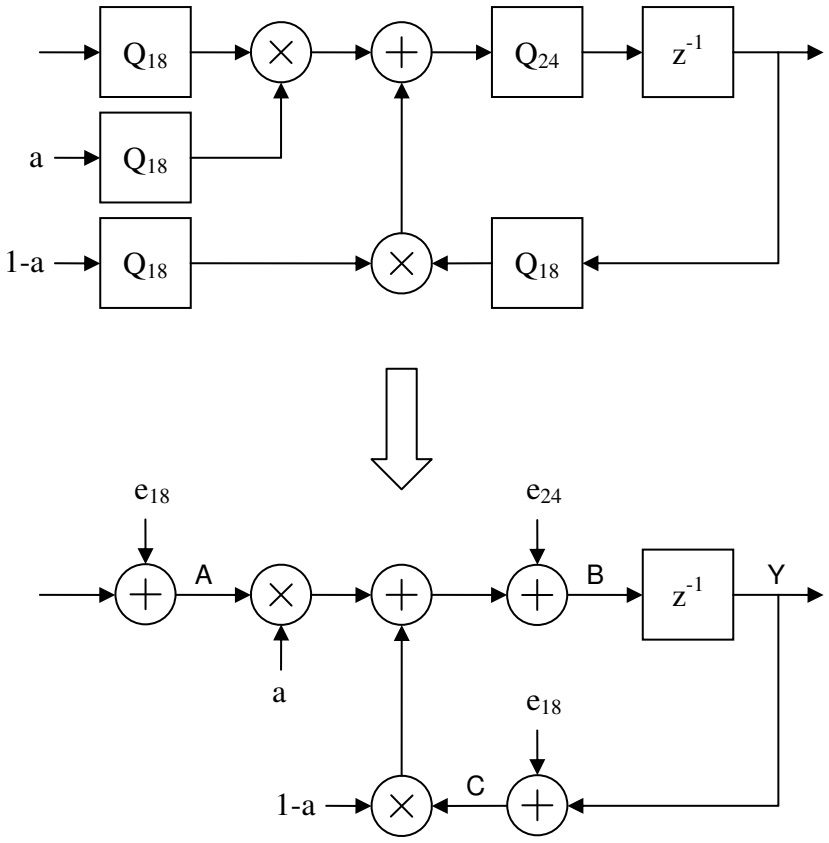


Fig. 25: First Order Lowpass Filter including Requantization Noise

$$H_{A\to Y} = \frac{a}{z-1+a} \qquad H_{B\to Y} = \frac{1}{z-1+a} \qquad H_{C\to Y} = \frac{1-a}{z-1+a}$$

$$\sigma_{Y(A)}{}^2 = \frac{2^{-36}}{3}\cdot\frac{a}{2-a} \qquad \sigma_{Y(B)}{}^2 = \frac{2^{-48}}{3}\cdot\frac{1}{a(2-a)} \qquad \sigma_{Y(C)}{}^2 = \frac{2^{-36}}{3}\cdot\frac{(1-a)^2}{a(2-a)}$$

The worst case occurs for low values of a, whereby the noise term of source C dominates. For f_s = 48 kHz and f_o = 20 Hz, we get a ≈ 0.0026 and SNR ≈ 87 dB. An improved structure is found by making all products proportional to the factor a. Consequently, input requantization noise induced in the multiplication now scales down with decreasing a. (Fig. 26)

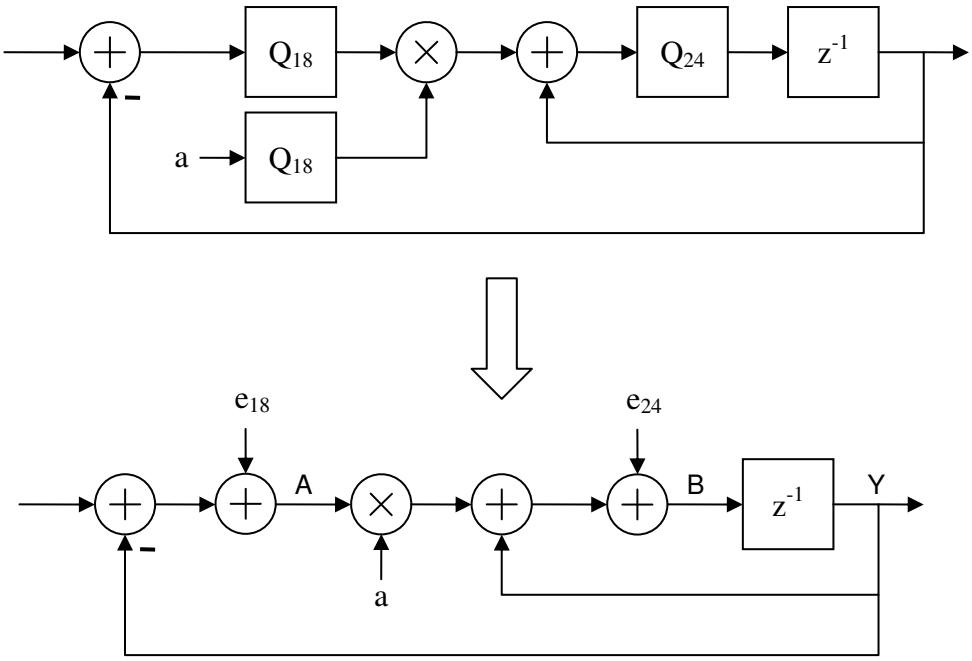


Fig. 26: Improved First Order Lowpass Filter including Requantization Noise

The new structure shows up exactly the same noise terms, but source C has vanished.

Hence, we find $$\sigma_{Y(tot)}{}^2 = \frac{2^{-36}}{3} \cdot \frac{a^2 + 2^{-12}}{a(2-a)}$$

and observe that the noise from state variable quantization starts to dominate below a ≈ 0.015. This becomes evident in the SNR plot (Fig. 27) and substantiates the practice of providing considerably higher resolution for data memory and accumulators than for multiplicands.

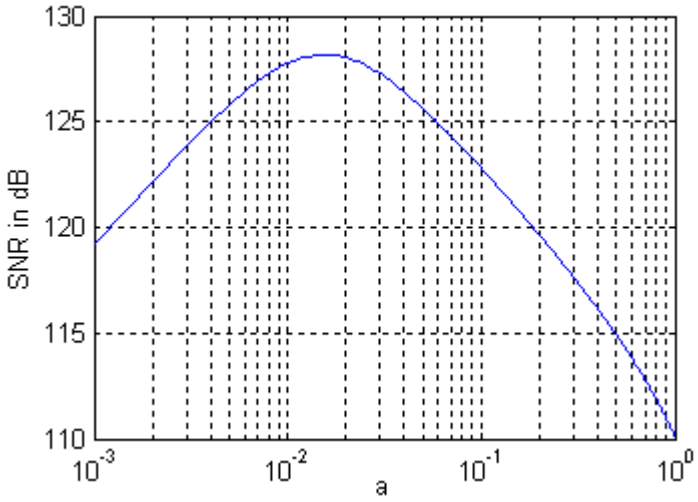
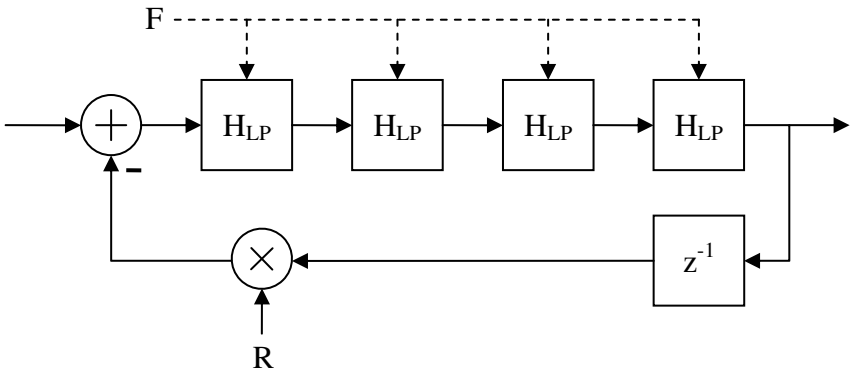


Fig. 27: SNR of the Improved First Order Lowpass Filter

We also found out that a near unity gain feedback loop is prone to noise: A multiplication by 1-ε should be split into a multiplication by ε and an addition. The case ε-1 is equally critical. Alternatives like error feedback may help in stubborn cases at the expense of a slightly increased computational effort. Some applications of this technique are discussed in [4]. We finish the section mentioning that the Chamberlin structure does not pose any noise problems in sound synthesis applications on 24 bit fixed and 32 bit floating point architectures.

3.7 Discrete Moog Lowpass Filter

A discretized version of this famous analog filter is discussed in [1]. We will refine the design to obtain smooth frequency sweeps comparable to the original and to render the characteristic saturation without audible aliasing. Experiments indicate that we have to oversample at least by a factor of two to handle the indispensable wideband nonlinearity. Fig. 28 shows the basic linear system.



$$H_{LP} = \frac{F}{1.3} \cdot \frac{z + 0.3}{z - 1 + F}$$

$$f = f_c\left(1 + 0.03617 f_c (4 - r)^2\right)$$
$$F = 1.25 f\left(1 - 0.595 f + 0.24 f^2\right)$$
$$R = r(1 + 0.077F - 0.117F^2 - 0.049F^3)$$

f_c = normalized cutoff frequency, $0 < f_c \leq 1$ r = resonance, $0 \leq r < 4$

Fig. 28: Basic Discrete Moog Lowpass Filter, f_s = 96 kHz

The first equation makes the filter transparent for r = 0 and f_c = 1. If f_c were used directly, frequencies around 20 kHz would be attenuated up to 12 dB for low r. The second equation maps the cutoff frequency to a linear scale with an accuracy of ±3 cents up to 7 kHz at the point of self-oscillation (r = 4). The third equation decouples the peak gain from the cutoff frequency reasonably well up to a factor of several hundreds. Since amplitude compression will be added later, the regularity of even higher peak gains versus frequency is not critical. Fig. 29 and 30 depict the filter performance for various parameter settings.

Nonlinear compression of resonant peaks is inherent to the bipolar differential stages of the analog Moog cascade. In a first attempt to emulate the behavior, we may use the hyperbolic tangent or a similar sigmoidal function somewhere in the loop. The results are however not satisfactory with regard to aliasing for strong input signals with significant high frequency energy. Weak signals, on the other hand, lead to irregular sweeps caused by pronounced amplitude maxima when the cutoff frequency is close to a low order harmonic. Listening tests suggest adopting the technique from section 3.4 in a modified way for overall bandlimited compression while an additional nonlinearity, preferably after the second or third inner lowpass block, generates the pleasant sounding saturation. The proposed structure in Fig. 31 also includes input scaling to compensate for the amplitude drop with increasing resonance. Moreover, it is capable of self-oscillation.

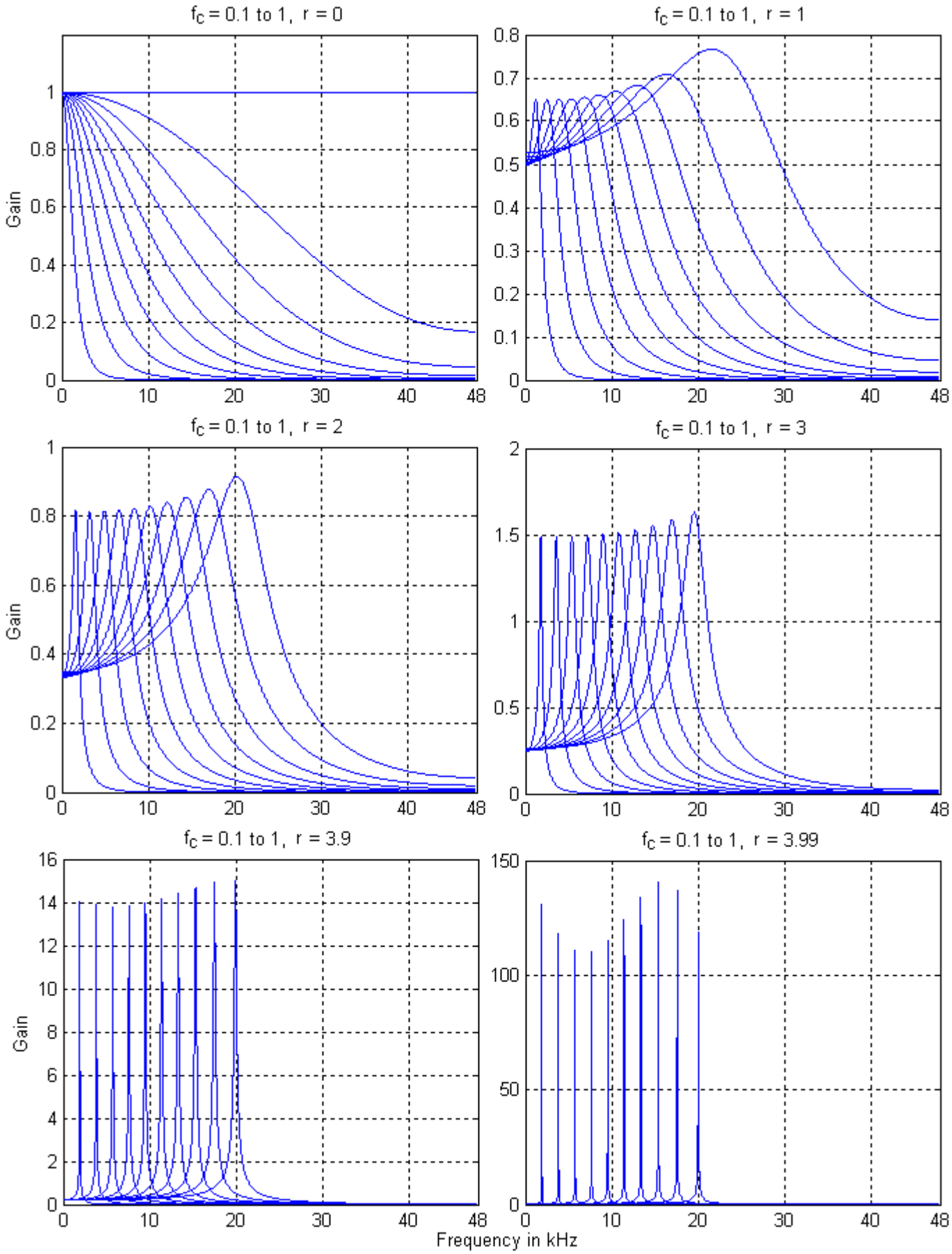


Fig. 29: Magnitude Response of the Basic Discrete Moog Lowpass Filter, f_s = 96 kHz

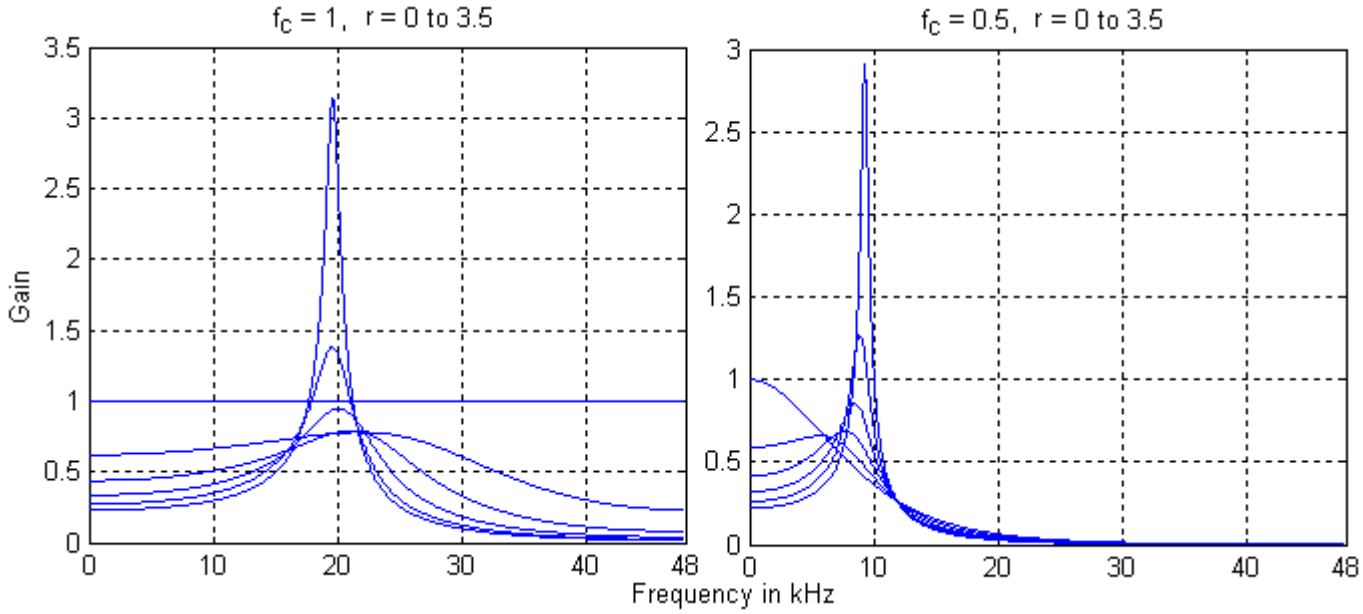
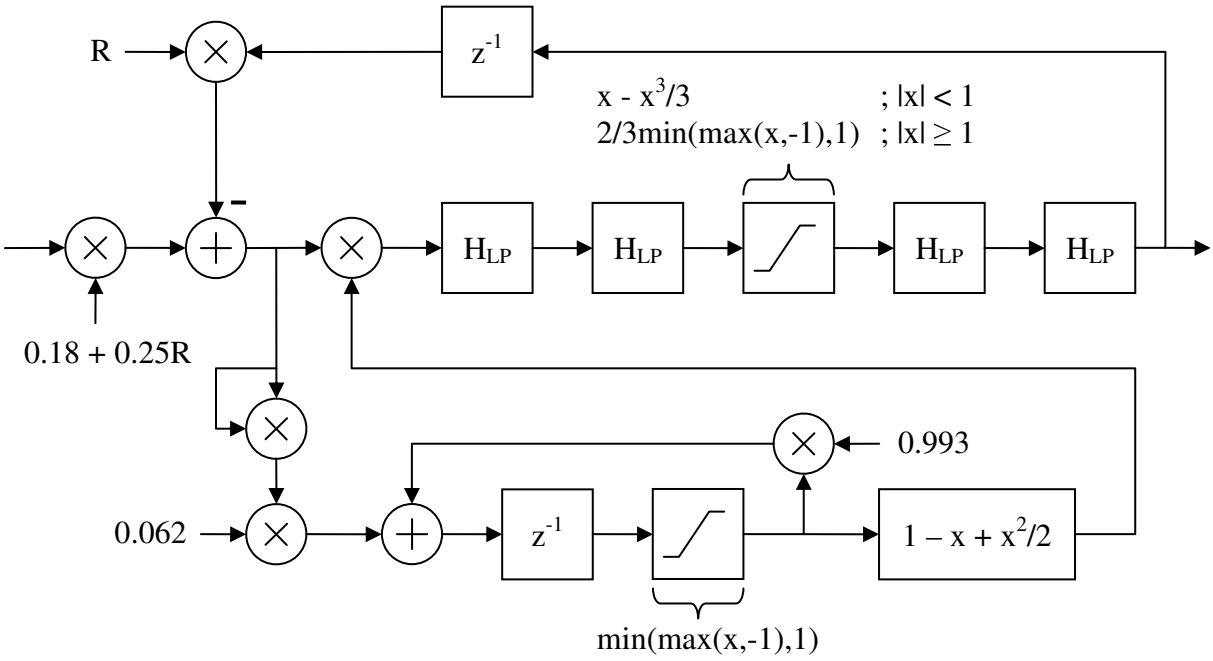


Fig. 30: Magnitude Response of the Basic Discrete Moog Lowpass Filter, f_s = 96 kHz (cont.)



$$H_{LP} = F\frac{z+0.3}{z-1+F}$$

$$f = f_c\left(1+0.5787 f_c (1-r)^2\right)$$
$$F = 1.25 f\left(1-0.595 f + 0.24 f^2\right)$$
$$R = r(1.4 + 0.108F - 0.164F^2 - 0.069F^3)$$

f_c = normalized cutoff frequency, $0 < f_c \leq 1$
r = resonance, $0 \leq r < 1$ (for self-oscillation: < 1.05)

Fig. 31: Low-Alias Nonlinear Discrete Moog Lowpass Filter, f_s = 96 kHz

Last but not least, we aim at an oversampled standard rate version of the nonlinear filter. Unlike the Chamberlin type, it comprises a wideband nonlinearity that potentially creates frequency components around f_s, which lead to audible aliasing if not suppressed before the decimation. This problem is addressed in the proposed decimator. It consists of a FIR part with a triple zero at f_s that causes the suppression and an IIR part to compensate the collateral high frequency attenuation in the audio band after the decimation. The provisions taken to minimize aliasing induced by the nonlinearities can be summarized as follows:

- Upsampling by repeating the input value acts like a two-point averaging FIR filter effectively creating a zero at f_s. Thus, subsequent stages in the filter will see very little energy around f_s.

- At higher amounts of resonance, spectral components above the audio band are attenuated in the inner lowpass stages before they reach the wideband nonlinearity.

- Before the signal is resampled at f_s to form the output, the decimator attenuates components around f_s that may have been generated by the wideband nonlinearity.

The complete system is shown in Fig. 32. Note that the previous input sample is used in the first pass and the decimator output is taken after the second pass. Again, the indication of the actual values in the two passes is preferred to multirate blocks for additional clarity with regard to implementation.

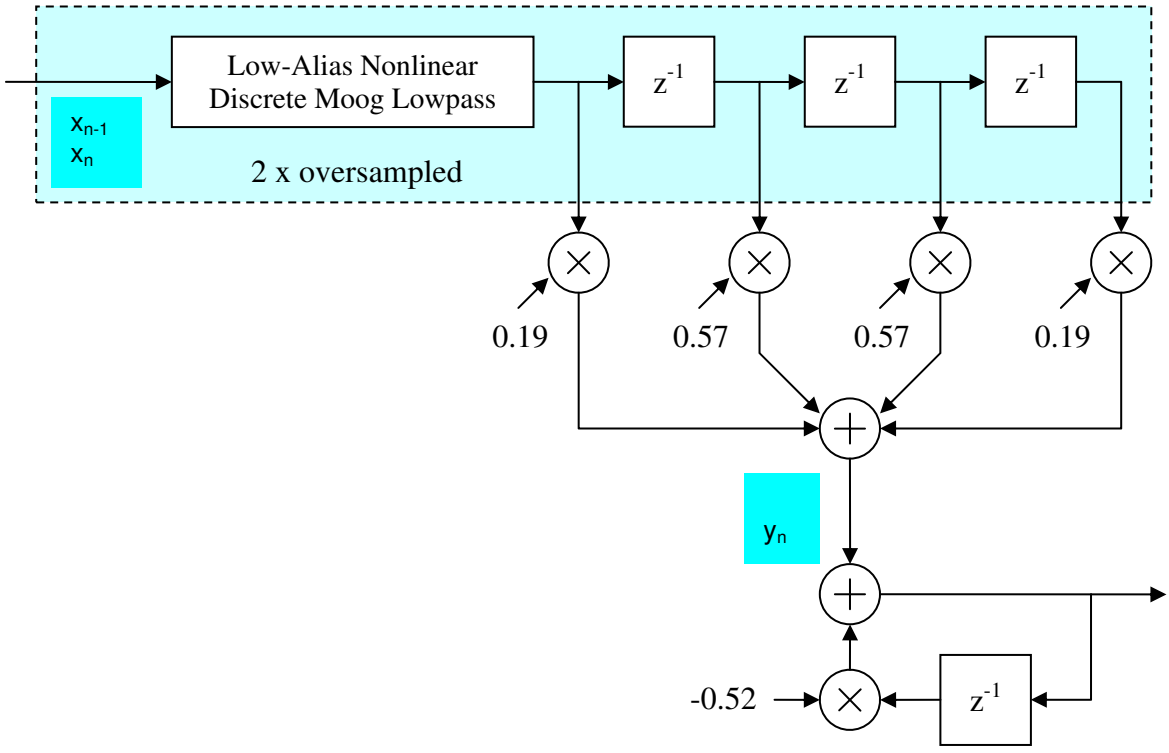


Fig. 32: Two-Fold Oversampled Low-Alias Nonlinear Discrete Moog Lowpass, f_s = 48 kHz

Most polyphonic synthesizers feed the filter output to a linear gain stage and then sum up the signals of all voices. In this case, some computation is saved if we insert the gain stage before the decimation and handle the sum in a single decimator.

The oversampling process introduces minor changes to the frequency response (Fig. 33), mainly for very low values of r. High-frequency shelving is caused solely by the decimator and does not manifest itself at the nonlinearities of the filter. While the difference is insignificant from an auditory perspective in sound synthesis applications with $f_s \geq 48$ kHz, lower sample rates call for changing the coefficient in the first equation of Fig. 31 from 0.5787 to 0.5. This will reduce shelving at the price of a slightly more pronounced dip.

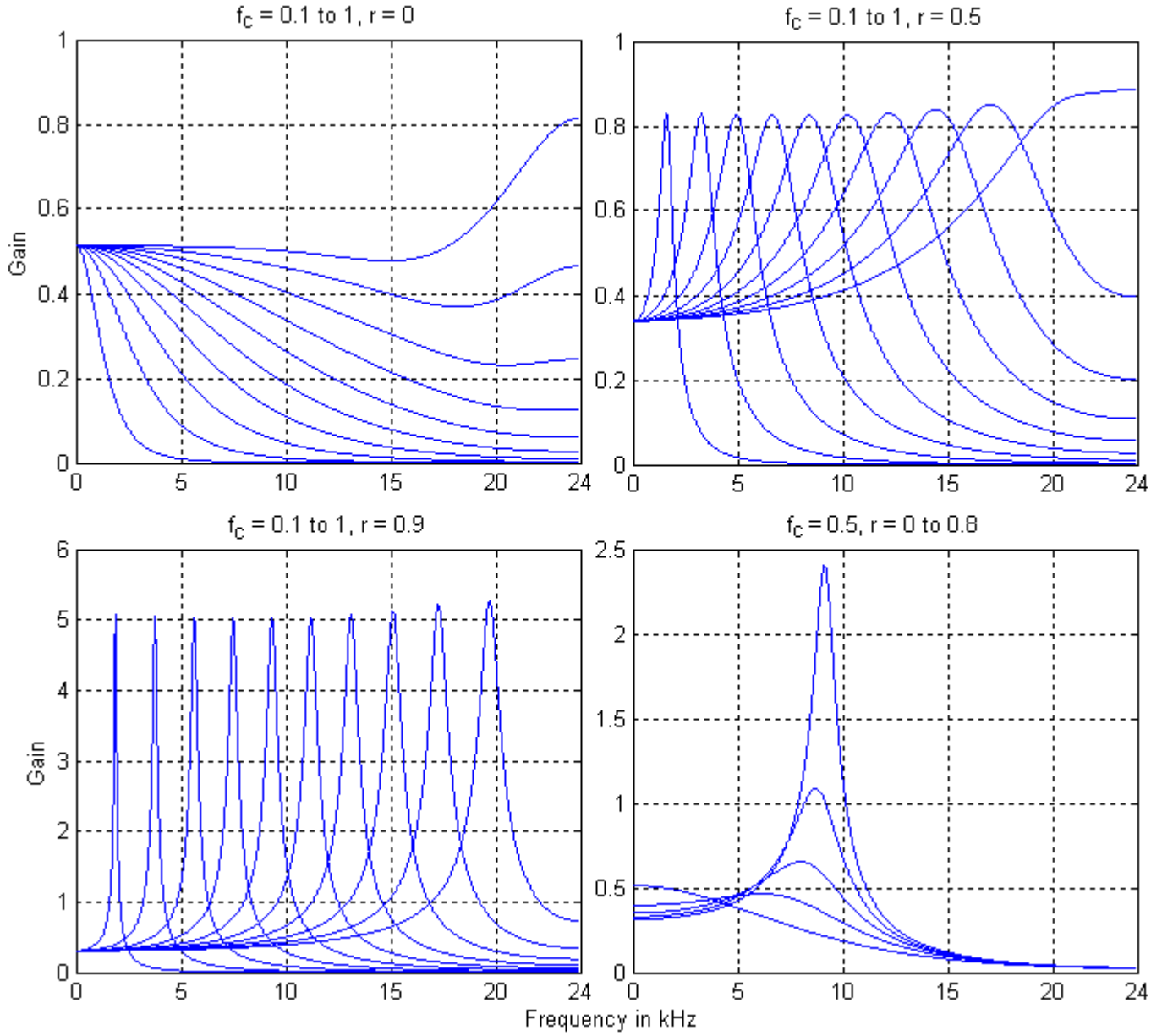


Fig. 33: Small-Signal Magnitude Response of the Two-Fold Oversampled Nonlinear Moog Lowpass Filter

4 Specialized and Auxiliary Filters

4.1 First Order Low and High Pass Filters

Simple lowpass filters are ubiquitous in musical synthesizers. A compact first order structure is shown in Fig. 34.

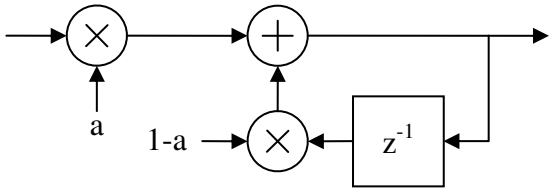


Fig. 34: First Order All-Pole Lowpass Filter

The transfer function and magnitude response are:

$$H(z) = \frac{az}{z - 1 + a} \qquad \left|H\left(e^{j\Omega}\right)\right| = \frac{1}{\sqrt{1 + 4\left(\frac{1-a}{a^2}\right)\sin^2\left(\frac{\pi f}{f_s}\right)}} \quad ; 0 < a \le 1$$

With f_c denoting the -3db corner frequency, we get:

$$a = 2\left(\sqrt{\lambda^2 + \lambda} - \lambda\right) \quad \text{with} \quad \lambda = \sin^2\left(\frac{\pi f_c}{f_s}\right) \quad ; 0 < f_c \le f_s/2$$

$$\text{and} \quad f_c = \frac{f_s}{\pi}\arcsin\left[\frac{a}{2\sqrt{1-a}}\right] \quad ; 0 < a \le \sqrt{8} - 2$$

For $f_c << f_s/(2\pi)$, the following approximation holds: $a \approx 2\pi f_c / f_s$

It is often desirable to warp the corner frequency parameter such that its maximum value yields a = 1 and the filter becomes transparent. A suggested mapping is:

$a = \beta - 0.25\beta^2$ with $\beta = 2\pi f_{cm} / f_s$ and $f_{cm(max)} = f_s/\pi$.(Fig. 35)

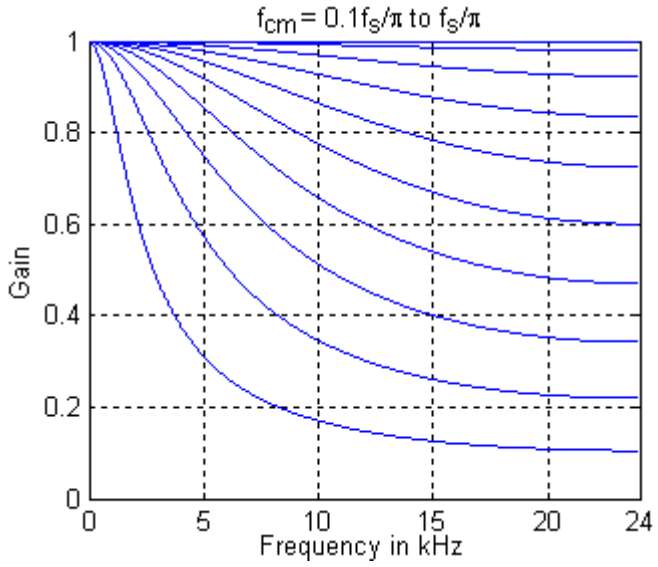


Fig. 35: Magnitude Response of the First Order All-Pole Lowpass Filter with Warped f_c

A slightly more complex structure with an additional zero at $f_s/2$ may help if the compromised behavior of the all-pole filter in the top octave is unacceptable. (Fig. 36)

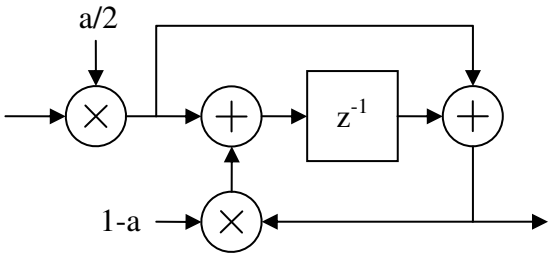


Fig. 36: First Order Pole-Zero Lowpass Filter

The transfer function and magnitude response are:

$$H(z) = \frac{a}{2} \cdot \frac{z+1}{z-1+a} \qquad \left|H\left(e^{j\Omega}\right)\right| = \frac{\left|\cos\left(\frac{\pi f}{f_s}\right)\right|}{\sqrt{1+4\left(\frac{1-a}{a^2}\right)\sin^2\left(\frac{\pi f}{f_s}\right)}} \qquad ; 0 < a < 2$$

With f_c denoting the -3db corner frequency, we get:

$$a = \frac{2\sin\lambda}{\cos\lambda + \sin\lambda} \qquad \text{with} \qquad \lambda = \frac{\pi f_c}{f_s} \qquad ; 0 < f_c < f_s/2$$

$$\text{and} \qquad f_c = \frac{f_s}{2\pi}\arccos\left[\frac{1-a}{1-a+a^2/2}\right] \qquad ; 0 < a < 2$$

For $f_c << f_s/(2\pi)$, the following approximation holds: $a \approx 2\pi f_c / f_s$

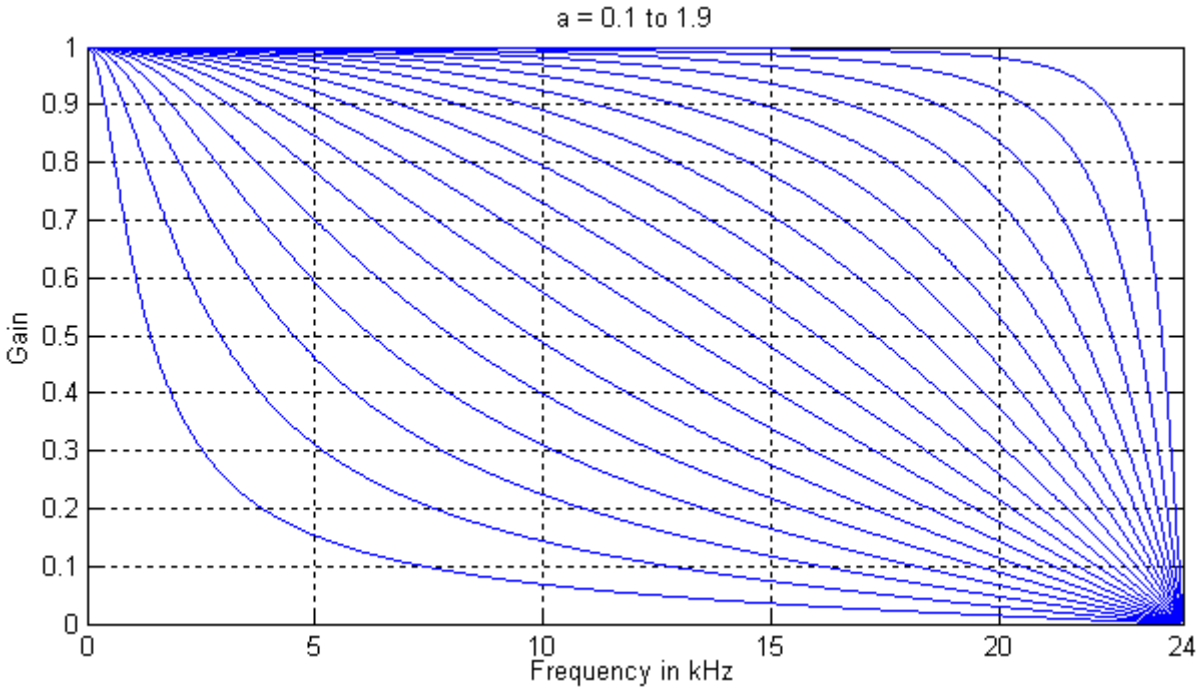


Fig. 37: Magnitude Response of the First Order Pole-Zero Lowpass Filter

The complementary highpass filter has the transfer function $H(z)=\left(1-\frac{a}{2}\right)\frac{z-1}{z-1+a}$ and the magnitude response $\left|H\left(e^{j\Omega}\right)\right|=\frac{(2-a)\left|\sin\left(\frac{\pi f}{f_s}\right)\right|}{\sqrt{a^2+4(1-a)\sin^2\left(\frac{\pi f}{f_s}\right)}}$; 0 < a < 2.

The formulae for the corner frequency are identical to those of the lowpass type. A practical realization is depicted in Fig. 38.

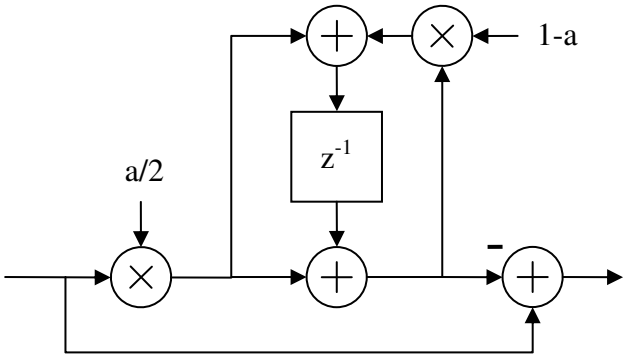


Fig. 38: First Order Pole-Zero Highpass Filter

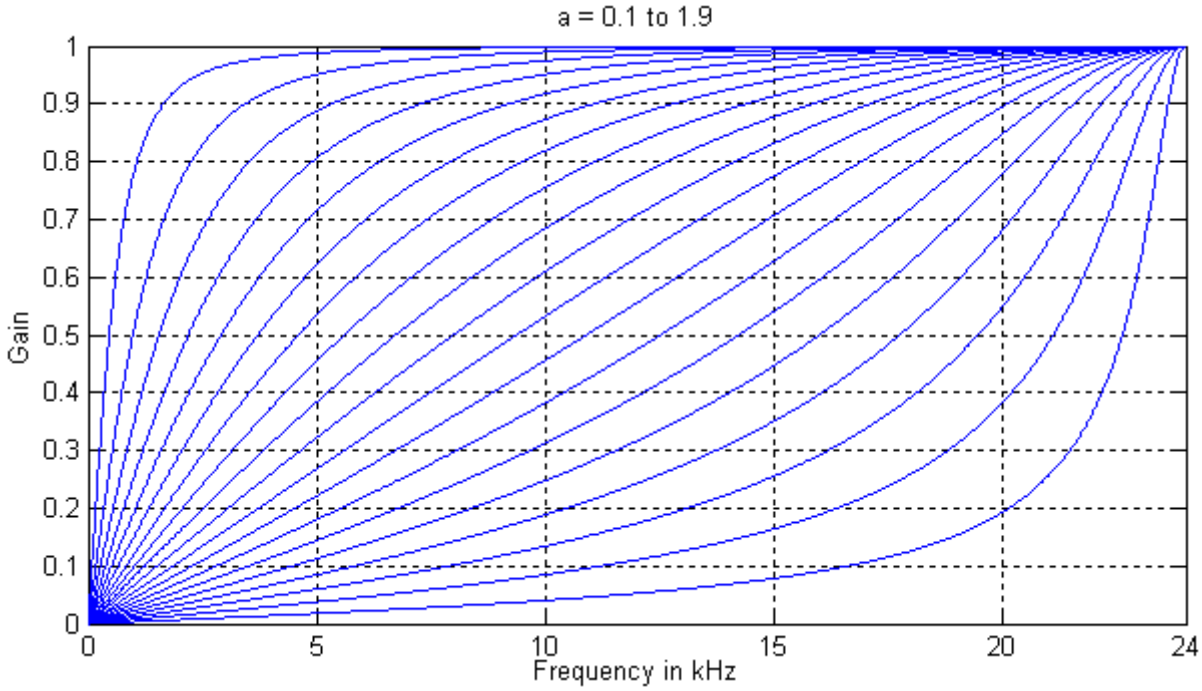


Fig. 39: Magnitude Response of the First Order Pole-Zero Highpass Filter

All filters of section 4.1 have the property that the steady state amplitude at each summation node and in each memory element does not exceed the input amplitude. The pole-zero filters become noisy for a ≈ 2. A recommended limit on 24-bit fixed and 32-bit floating point architectures is a < 1.98.

Certain hardware platforms, like FPGAs and microprocessors, support at least 32-bit fixed point arithmetic except for the inputs of the multipliers. In this case, modifications have to be made to avoid excessive noise and often also to reduce the number of multiplications and limit the factors to an absolute value of 1. Some examples are shown below. The reader may refer to section 3.6 for further insight into the underlying principles.

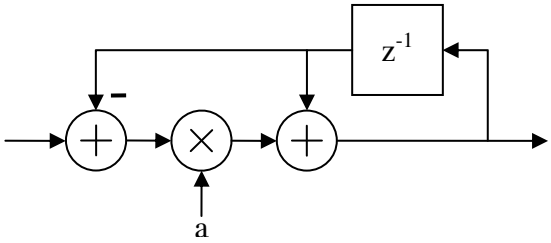


Fig. 40: First Order All-Pole Lowpass Filter for Low Factor Resolution

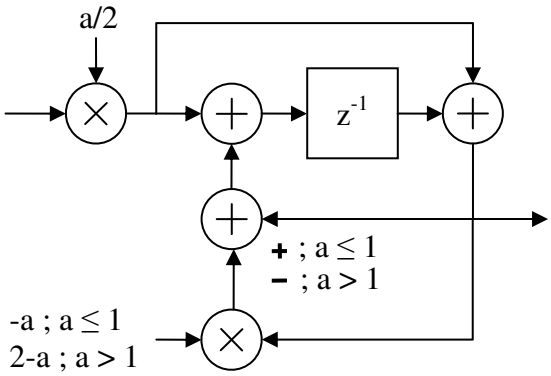


Fig. 41: First Order Pole-Zero Lowpass Filter for Low Factor Resolution

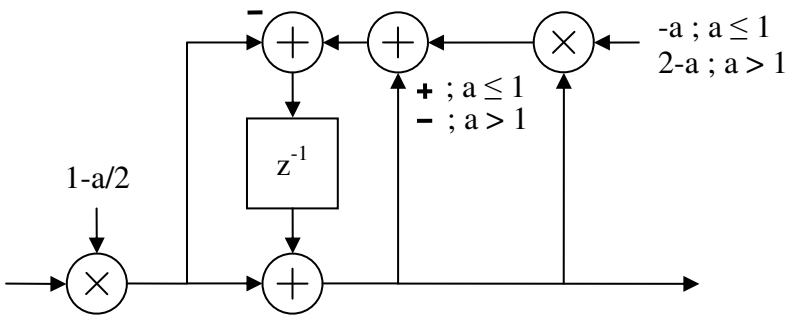


Fig. 42: First Order Pole-Zero Highpass Filter for Low Factor Resolution

4.2 First Order Allpass Filter

Allpass filters exhibit a frequency-dependent phase shift while maintaining unity gain. Although this seems unspectacular, they are versatile building blocks, used for example in reverberators, interpolators, fractional length delays, physical models, frequency-warped filters, phaser effects, equalizers, and many more. A typical realization is shown in Fig. 43.

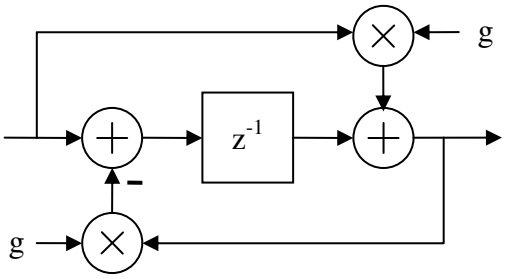


Fig. 43: First Order Allpass Filter

The transfer function is $H(z) = \frac{gz+1}{z+g}$. Hence, the magnitude response becomes $\left|H(e^{i\Omega})\right| = 1$

and the phase response $\varphi_H = \arg\{H(e^{i\Omega})\} = -\arctan\left[\frac{(1-g^2)\sin(2\pi f / f_s)}{2g+(1+g^2)\cos(2\pi f / f_s)}\right]$. (Fig. 44)

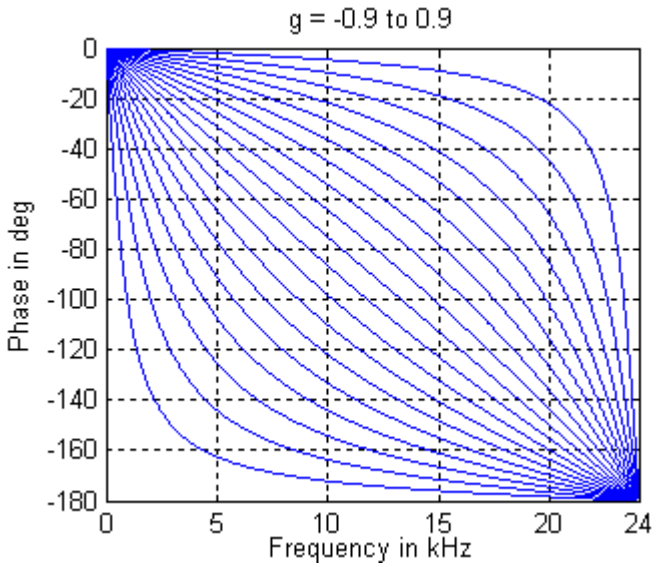


Fig. 44: Phase Response of the First Order Allpass Filter

Group delay is calculated as $\tau = -\frac{\partial \varphi_H}{\partial \omega}$. We start with the polar form of the transfer function $H(e^{i\omega T}) = \left|H\right| e^{i\varphi_H}$. Taking the logarithm followed by differentiation and separating the real and imaginary parts yields: $\tau = -\frac{\partial \varphi_H}{\partial \omega} = -\mathrm{Re}\left\{\frac{zT}{H(z)}\frac{\partial H(z)}{\partial z}\bigg|_{z=e^{i\omega T}}\right\}$.

In the specific case of the first order allpass, we obtain

$$\tau = \frac{1-g^2}{1+g^2+2g\cos(2\pi f / f_s)}T$$

with the approximation $\tau \approx \frac{1-g}{1+g}T$ for $f << f_s/(2\pi)$ and $-0.1 < g < 1$.

Fig. 45 depicts the group delay in sampling intervals T for different values of g suggesting the suitability of an allpass filter as a variable delay element (see section 4.3).

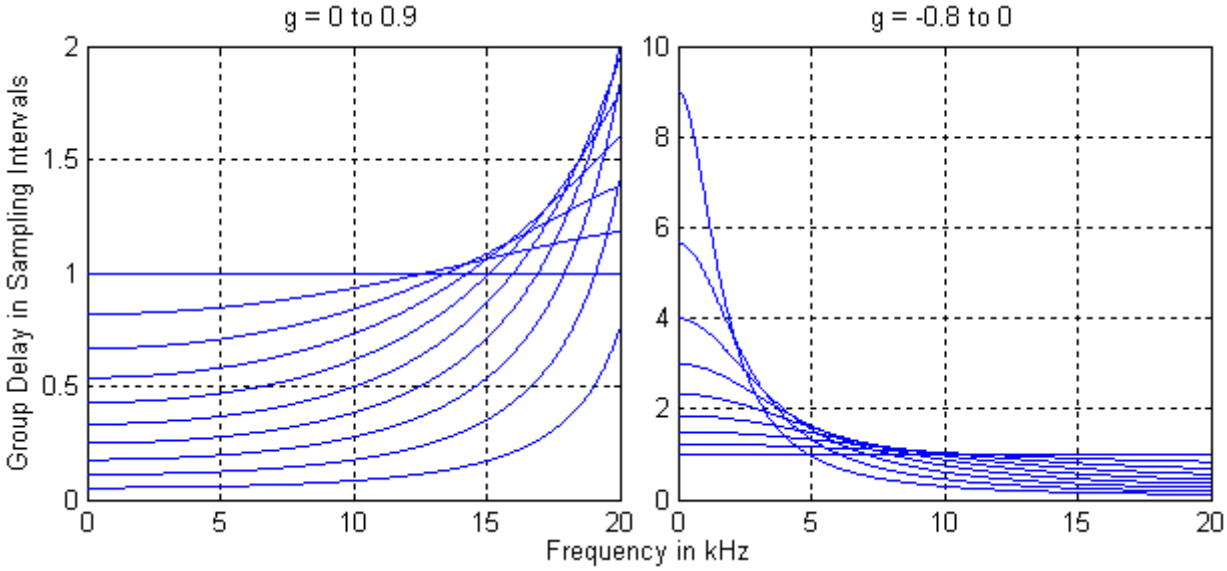


Fig. 45: Group Delay of the First Order Allpass Filter, f_s = 48 kHz

Group delay is a measure of the lag that a system induces onto a compact wave packet with its energy concentrated around a certain frequency. Likewise, it indicates how fast an amplitude change of a sinusoid propagates from input to output. Sometimes, we are more interested in the lag of a steady-state sinusoid. It is given by the phase delay $\tau_p = -\varphi_H/\omega$, which closely approximates the group delay for low frequencies but differs at higher ones.

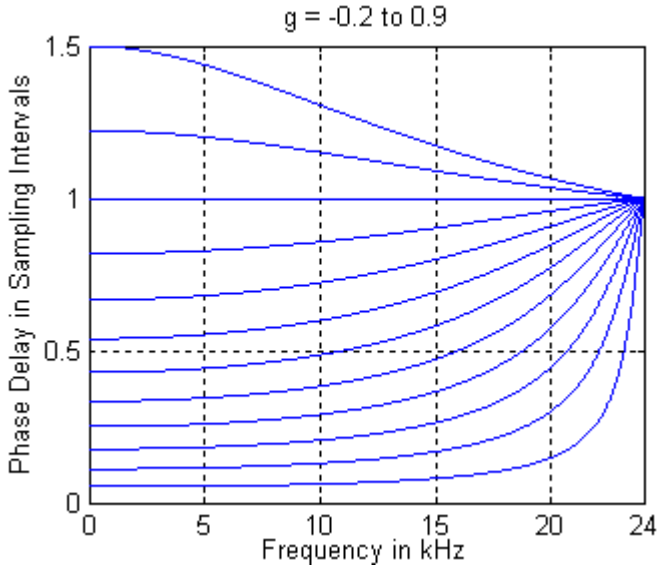


Fig. 46: Phase Delay of the First Order Allpass Filter

There's also a single-multiply realization of the first order allpass. (Fig. 47)

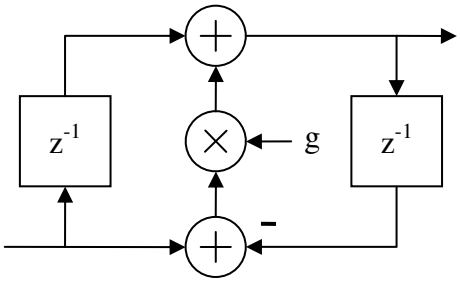


Fig. 47: Single-Multiply First Order Allpass Filter

All of the aforementioned allpass structures become noisy and exhibit ringing around Nyquist at $|g| \approx 1$ due to pole-zero cancellation. This condition should only be maintained for a short period and is avoided easily by using a range that includes negative values of g, for example g = [-0.13, 0.54], which however results in an additional delay of 0.3T compared to g = [0, 1]. If $|g| \approx 1$ seems inevitable, refer to section 3.6 for low-noise design.

4.3 Comb Filters

Comb filters show periodic extrema in their magnitude response that lead to a characteristic coloration perceived as resonating, hollow, or even vowelish. They are realized by mixing a delayed version of the signal with the original using forward and feedback paths. Comb filters are key ingredients of tube and string models. Furthermore, if their delay length is modulated, chorus and flanger type effects are obtained.

We start by examining the feedback only version.

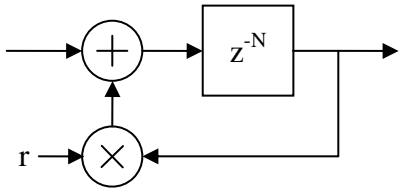


Fig. 48: Feedback Delay Comb Filter

The transfer function and magnitude response are:

$$H(z) = \frac{1}{z^N - r} \qquad \left|H\left(e^{j\Omega}\right)\right| = \frac{1}{\sqrt{1 - 2r\cos\left(2\pi N \frac{f}{f_s}\right) + r^2}} \qquad ; |r| < 1$$

The magnitude peaks at integer multiples of f_s/N for $0 < r < 1$ with a maximum of $\frac{1}{1-r}$.

For negative r, the peaks are shifted by $f_s/(2N)$.

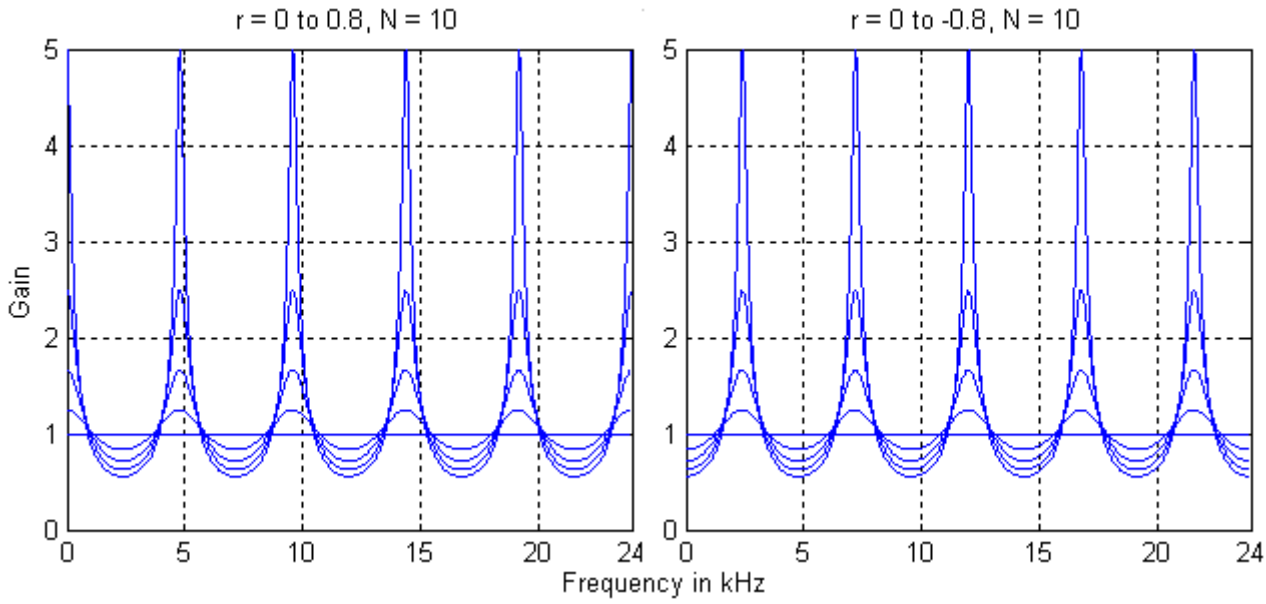


Fig. 49: Magnitude Response of the Feedback Delay Comb Filter

A different kind of comb filter is created by adding the original signal to the output of a first order allpass filter whose unit delay has been extended.

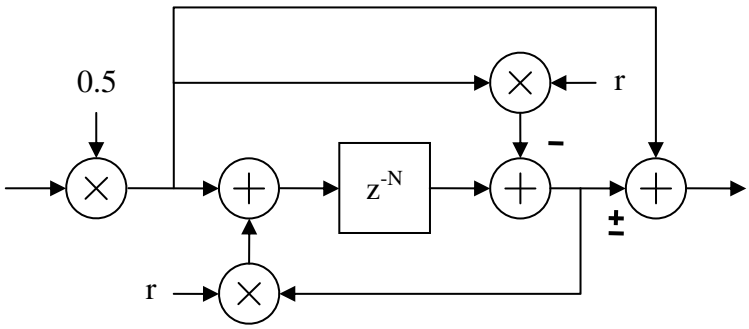


Fig. 50: Allpass-based Comb Filter

The transfer function and magnitude response are:

$$H(z) = \frac{(1 \mp r)}{2} \cdot \frac{z^N \pm 1}{z^N - r} \qquad \left|H\left(e^{j\Omega}\right)\right| = (1 \mp r) \frac{\left|\begin{matrix}\cos \\ \sin\end{matrix}\left(\pi N \frac{f}{f_s}\right)\right|}{\sqrt{1 - 2r\cos\left(2\pi N \frac{f}{f_s}\right) + r^2}} \qquad ; |r| < 1$$

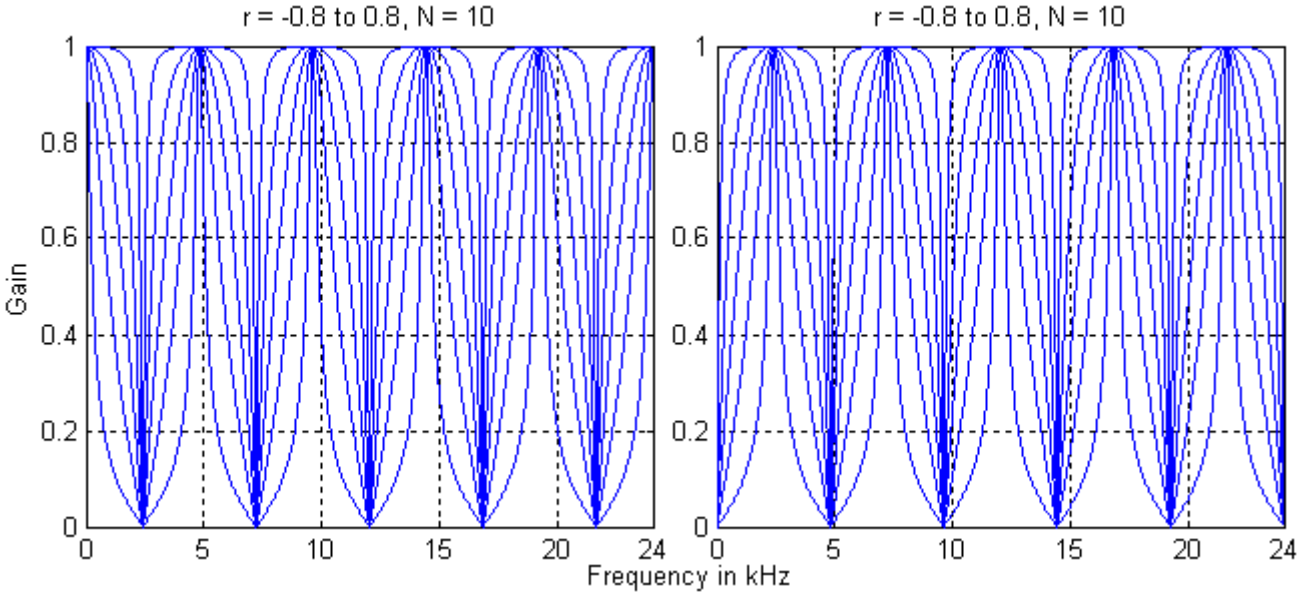


Fig. 51: Magnitude Response of the Allpass-based Comb Filter
(Left: Sum type, Right: Difference type)

Since the feedback delay as well as the allpass-based type have musical applications, it's suggesting to look for a structure that gives us the best of both worlds. Listening tests favor a characteristic ranging from a pure allpass at r = 0 to a pure feedback delay type for |r| = 1. An efficient realization of what we may call a hybrid comb filter is shown in Fig. 52.

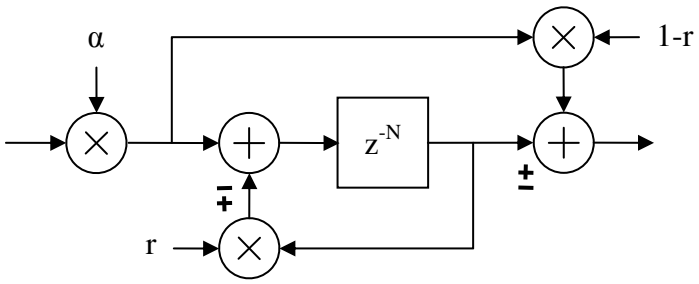


Fig. 52: Hybrid Comb Filter

The transfer function is: $$H(z) = \alpha \frac{(1-r)z^N \pm (1+r-r^2)}{z^N \pm r} \quad ; 0 \le r < 1$$

It becomes a unity gain allpass at $r = r_0 \approx 0.445$ and $\alpha \approx 0.802$.

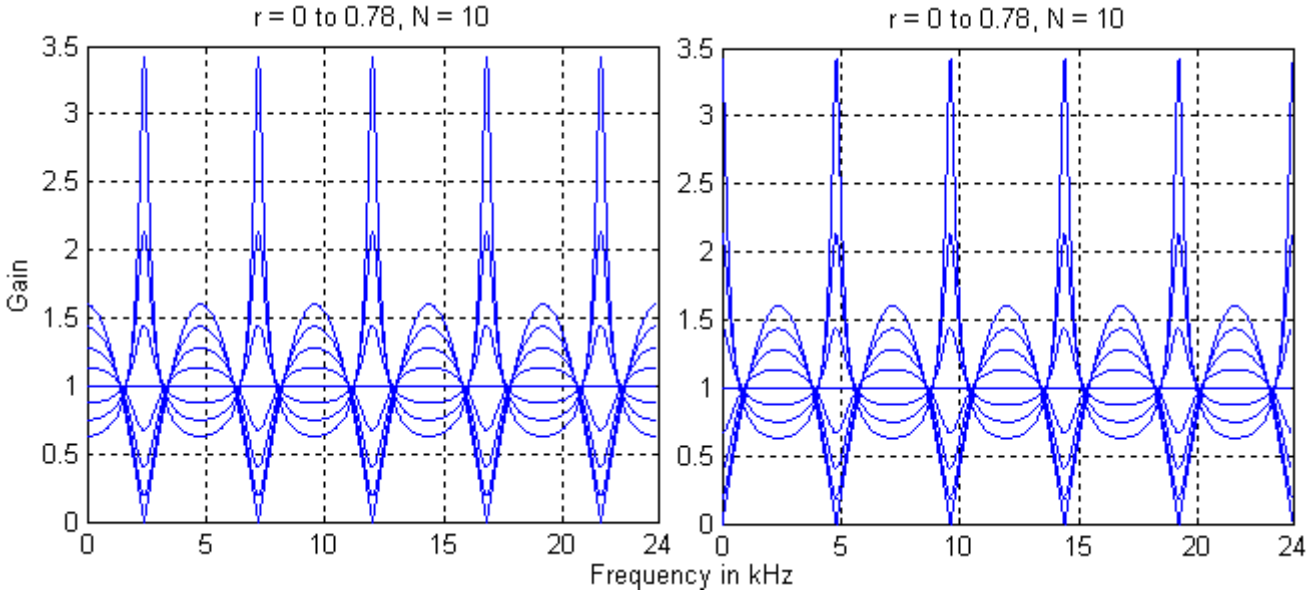


Fig. 53: Magnitude Response of the Hybrid Comb Filter
(Left: Sum type, Right: Difference type)

Switching the filter type when r passes through r_o enables us to control the balance between even and odd harmonics of the input signal up to the cancellation of either part if the delay length is matched to the fundamental. An accompanying change of α may also be performed to compensate for the expected lower energy of the even harmonics.

If the additional gain at low r is undesirable, a modification of the output stage will virtually eliminate it. Some examples are given in Fig. 54 and 55.

Experimental modifications of the hybrid comb filter structure are so rewarding that the reader is likely to always find a variation that fits his specific requirements.

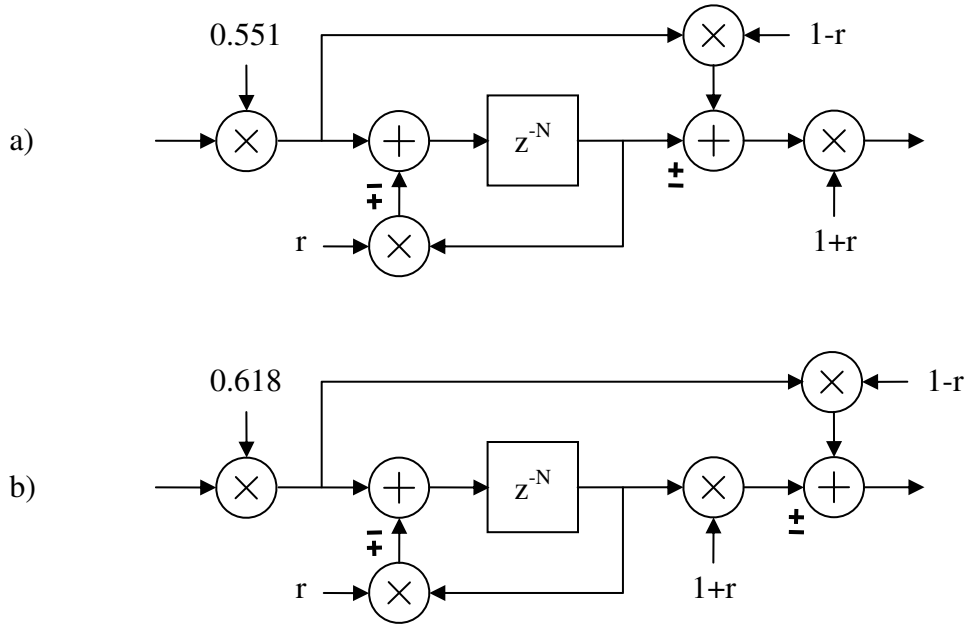


Fig. 54: Modified Hybrid Comb Filters (a: $r_o \approx 0.445$, b: $r_o \approx 0.382$)

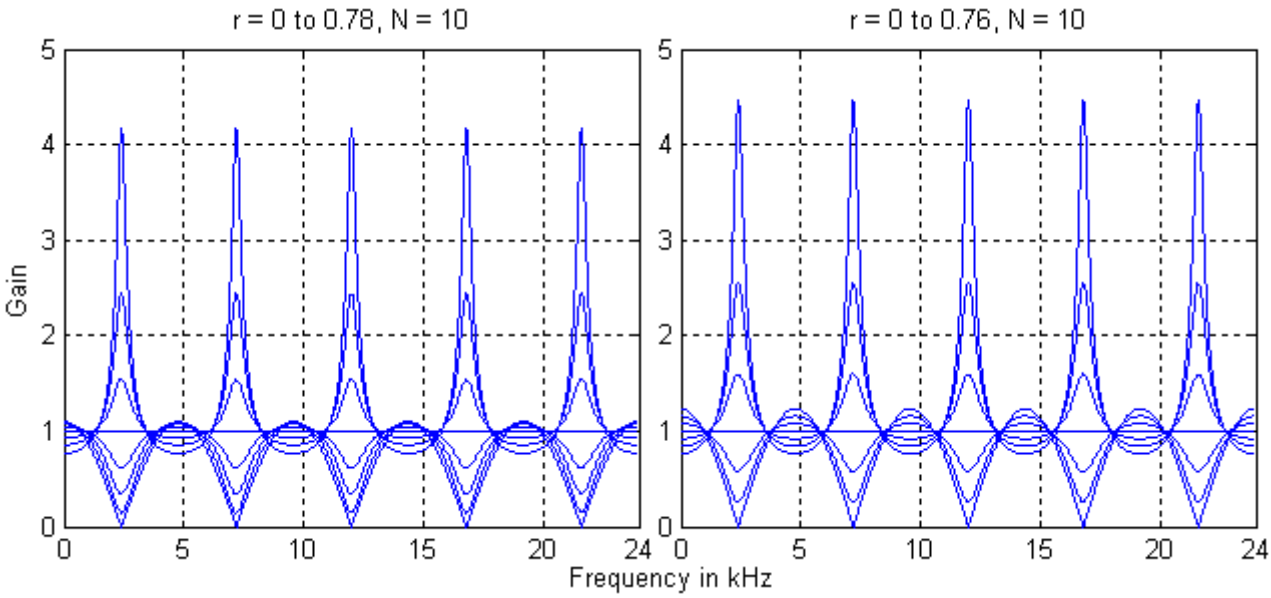


Fig. 55: Magnitude Response of the Modified Hybrid Comb Filters (Sum type, Left: a, Right: b)

So far, the delay length has been limited to integer multiples of the sampling interval. In the next step, we will extend it to fractional values in order to create a continuously tunable delay line (Fig. 56). Among the many ways to implement a fractional length delay [5][6], allpass interpolation has the primary advantage of letting high frequency components pass without attenuation, which is a fundamental requirement if we intend to apply the comb filter to physical models of low absorbent systems. It also sounds good as long as fast wide range length modulation and random access are avoided. The frequency-dependent phase delay of the allpass causes high frequency peaks to be out of tune. Fortunately, this often turns out to be merely a minor disadvantage, even with first order interpolation, because pitch perception degrades accordingly as the deviation becomes larger.

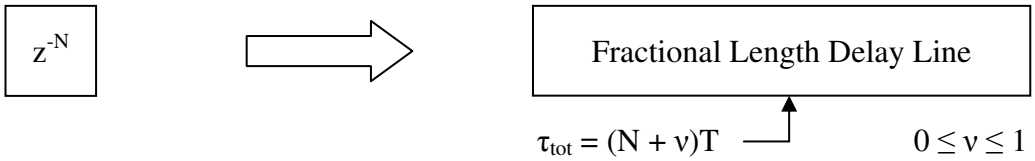


Fig. 56: Making the Comb Filter Tunable with a Fractional Length Delay Line

First, we analyze a fixed fractional length delay line based on an allpass filter (Fig. 57). Depending on the application, either the group or phase delay of the allpass is responsible for the fractional part νT of the total delay time τ_{tot}.

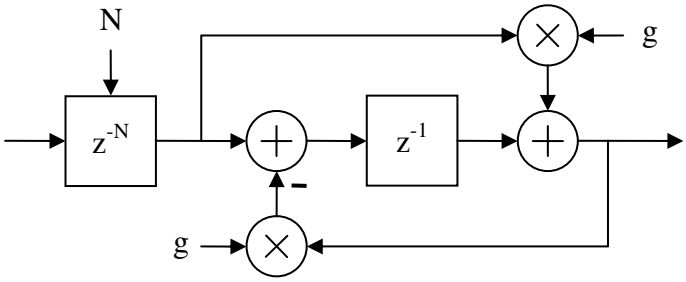


Fig. 57: Allpass-based Fixed Fractional Length Delay Line

Following section 4.2, the group delay of the allpass is $\tau = \frac{1-g^2}{1+g^2+2g\cos(2\pi f / f_s)}T$ and can be approximated, just like the phase delay, by $\tau \approx \frac{1-g}{1+g}T$ for $f << f_s/(2\pi)$ and $-0.1 < g < 1$.

Although τ is a function of the signal frequency, the coupling quickly becomes insignificant towards lower frequencies, where also the effect of a given delay error on the tuning of the comb filter diminishes. The relation between τ and g is normally too nonlinear for instant use, but a simple quadratic mapping fits a variety of applications where an accuracy of $|\tau - \nu T - 0.01T| < 0.029T$ up to $f = 0.05 f_s$ is satisfactory:

$$g = 0.98 - 1.612\nu + 0.627\nu^2 \qquad ; 0 \le \nu \le 1$$

A small offset that increases the delay by 0.01T has been built into the above formula to avoid $g \approx 1$. If near-zero delay capability is traded for a compact impulse response and improved accuracy, extending the offset to 0.3T yields $|\tau - \nu T - 0.3T| < 0.015T$ up to $f = 0.05 f_s$:

$$g = 0.539 - 1.037\nu + 0.369\nu^2 \qquad ; 0 \le \nu \le 1$$

Alternatively, higher order allpass filters provide a more constant τ versus frequency [5].

A time-varying delay line is made by simply modulating τ_{tot}. In this case, we modify the allpass in a way that the loop is always fed with two adjacent samples, a procedure known as first order allpass interpolation. It has the same transfer function for constant length but performs superior under time-varying conditions. (Fig. 58)

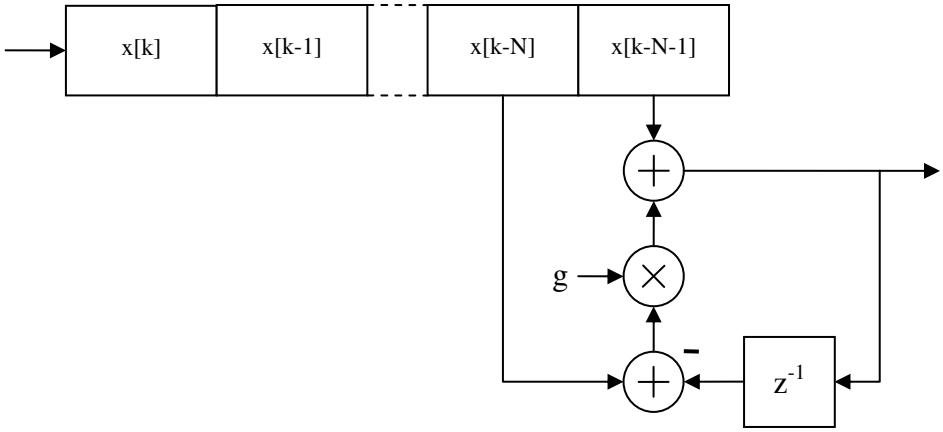


Fig. 58: Allpass Interpolated Time-Varying Fractional Length Delay Line

According to section 3.4, we may add circuitry to reduce amplitude peaking in the presence of strong feedback. (Fig. 59)

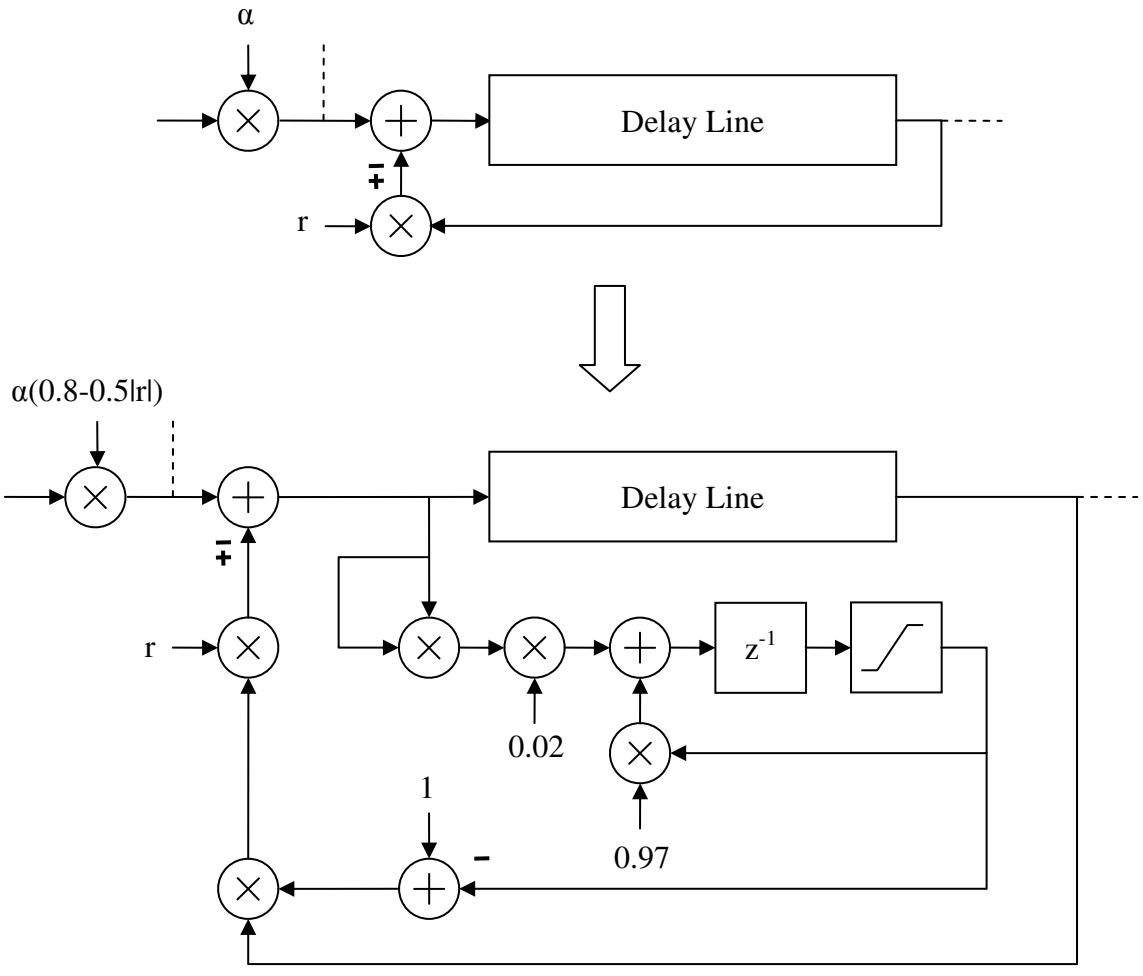


Fig. 59: Adding Bandlimited Saturation to Comb Filters

An inexpensive brickwall limiter suffices. The input scaling factor depends on the internals of the comb filter and is intended as a guideline. Listening tests are suggested for adjustment.

Another interesting addition would be a first order lowpass in the feedback loop to model the absorption of natural resonators without overly affecting transparency at r_o.

Appendix A: References

[1] T. Stilson, "Efficiently-Variable Non-Oversampled Algorithms in Virtual-Analog Music Synthesis", PhD Thesis, http://www-ccrma.stanford.edu/~stilti/papers/ (don't miss the substantial list of references)
[2] H. Chamberlin, "Musical Applications of Microprocessors", ISBN 978-0810457683
[3] J. Laroche, "Using Resonant Filters for the Synthesis of Time-Varying Sinusoids", Proc. 105th AES Convention, 1998, Preprint 4782
[4] U. Zölzer, "Digital Audio Signal Processing", ISBN 978-0471972266 (also available in German, ISBN 978-3519161806)
[5] T.I. Laasko, V. Välimäki, M. Karjalainen, U.K. Laine, "Splitting the Unit Delay – Tools for Fractional Delay Filter Design", IEEE Signal Processing Magazine, Jan. 1996, Vol. 13, pp. 30
[6] J. Dattorro, "Effect Design Part 2: Delay-Line Modulation and Chorus", JAES, Oct. 1997, Vol. 45, pp. 764, http://www.stanford.edu/~dattorro/
[7] P. Dutilleux, "Vers la machine à sculpter le son, modification en temps réel des caractéristiques fréquentielles et temporelles des sons", PhD Thesis

Recommended Additional Reading and Links

1. C. Roads, "The Computer Music Tutorial", 1996, ISBN 978-0-262-68082-0
2. J.O. Smith's Homepage, http://ccrma.stanford.edu/~jos/
3. M. Puckette's Book Project, http://crca.ucsd.edu/~msp/techniques/latest/book-html
4. E. Weisstein's World of Mathematics, http://mathworld.wolfram.com
5. GSL (GNU Scientific Library), http://www.gnu.org/software/gsl
6. No Matlab? Go here: http://www.scilab.org