Analyzing the Moog VCF with Considerations for Digital Implementation

Tim Stilson(stilti@ccrma.stanford.edu)
Julius Smith(jos@ccrma.stanford.edu)
CCRMA (http://www-ccrma.stanford.edu/)
Music Department, Stanford University

Abstract

Various alternatives are explored for converting the Moog four-pole Voltage Controlled Filter (VCF) to discrete-time form for digital implementation in such a way as to preserve the usefulness of its control signals. The well known bilinear transform method yields a delay-free loop and cannot be used without introducing an ad-hoc delay. Related methods from digital control theory yield realizable forms. New forms motivated by root locus studies give good results.

1 Introduction

The Voltage-Controlled Filter (VCF) designed and implemented by Robert Moog is an influential filter in the history of electronic music. In this paper, the filter is analyzed in continuous time and then several transformations of the filter into discrete time are analyzed for various properties such as efficiency, ease of implementation, and the retention of certain of the original filter's good properties, such as constant-Q, and separability of the Q and tuning controls. The Root-Locus, a particularly useful tool from control systems, is used extensively in the analysis of the VCFs.

The various transformations that turn continuous-time filters into discrete-time filters each have different characteristics that affect how the properties of the continuous-time system map into the discrete domain. Some transforms that will be studied are the backwards-difference transform and the bilinear transform. In a filter such as the Moog VCF, a possible goal in the move to the discrete domain is to preserve constant-Q. Under our definition of constant-Q, a transformation cannot made which is finite-order rational. We will see how well the rational transforms approximate constant-Q.

In this work, Root-Locus techniques were found to be extremely useful. The Root-Locus comes from control-systems analysis and has particular usefulness in the analysis of systems with sweepable control inputs (inputs intended to have signal-rate updates, such as audio-rate modulation or smooth sweeps of parameters susceptible to zippering). Because the amount of processing available to translate these parameters into algorithm parameters is typically in short supply (so the algorithm is typically designed around these parameters), the parameter usually enters into the filter's equations simply, maybe even linearly. The traditional Root-Locus can plot the locations of the system's poles with variations in the parameter if the parameter enters in linearly, and many techniques in control-system synthesis can be applied to the design to keep the complexity down. The rules of how the root locus works also give the designer new tools and hints for sweepable filter design.

2 The Moog VCF

The VCF used in Moog synthesizers employs the filter structure shown in Fig. 1.

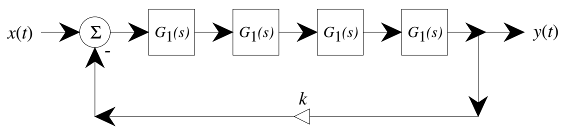

Figure 1: *The Moog VCF.*

The transfer function of each section is

$$G_1(s) = \frac{1}{1 + s/\omega_c}$$

The four real poles at $s = -\omega_c$ combine to provide a lowpass filter with cut-off frequency (-3 dB point) at

$\omega = \omega_c$. The overall transfer function with feedback as shown is

$$H(s) \triangleq \frac{Y(s)}{X(s)} = \frac{G_1^4(s)}{1+kG_1^4(s)} = \frac{1}{k+(1+s/\omega_c)^4}$$

where g is the feedback gain which is varied between 0 and 4. Each real pole section can be implemented as a simple (buffered) RC section. Moog implemented the RC sections using a highly innovative discrete analog circuit known as the "Moog ladder" [Moog 1965, Hutchins 1975].

At $\omega = \omega_c$, the complex gain of each pole section is

$$G_1(j\omega_c) = \frac{1}{1+j} = \frac{1}{\sqrt{2}} e^{j\frac{\pi}{4}}$$

Therefore, the gain and phase of all four sections are

$$G_1^4(j\omega_c) = \frac{1}{4} e^{j\pi} = \frac{1}{4}(-1)$$

I.e., the total gain is $1/4$ and the phase is -180 degrees (inverting). In contrast, at $\omega = 0$, the gain is 1 and the phase is 0 degrees (non-inverting), while at $\omega = \infty$, the gain is 0, and the phase is -360 degrees (also non-inverting). In summary, the four one-pole sections comprise a lowpass filter with cut-off frequency $\omega = \omega_c$, which is *inverting* at cut-off. Therefore, the use of inverting feedback provides *resonance* at the cut-off frequency. This is called "corner peaking" in analog synthesizer VCF design [Hutchins 1975, p. 5d(12)]. As the feedback gain k approaches 4, the total loop gain approaches 1, and the gain at resonance goes to infinity.

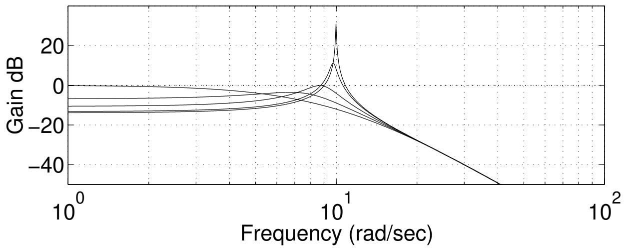

Figure 2: *Amplitude response of the analog Moog VCF for different levels of feedback ($\omega_c = 10rad/sec$). At $k = 0$, the dc gain is 1 and the filter is a lowpass without corner peaking. Also shown are $k = 4[0.3, 0.6, 0.9, 0.99]$. As k increases, corner peaking develops at the cut-off frequency. At $k = 4$, the lowpass filter oscillates at its cut-off frequency.*

Figure 2 shows a family of frequency response functions for the Moog VCF for a variety of feedback levels. As the feedback gain g goes from 0 to 4, the poles of the overall filter expand outward in an "X" pattern from $s = \omega_c$ until the two poles on the right reach the $j\omega$ axis at $\omega = \omega_c$.

Lowpass Nature: Since the one-pole filters are $G_1(s) = \frac{\omega_c}{s+\omega_c}$, we get

$$H(j\omega) = \frac{\omega_c^4}{(j\omega+\omega_c)^4 + k\omega_c^4}$$

so at $\omega \ll \omega_c$, $|H(j\omega)| \approx \frac{1}{1+k}$, and at $\omega \gg \omega_c$, $|H(j\omega)| \approx \frac{1}{\omega^4}$.

Root-Locus Interpretation

We can also analyze the VCF with the root-locus technique. Root Locus is a method popular in the field of control systems analysis that gives various rules for feedback-loop pole location movement in terms of the open-loop transfer function and the variations of the feedback gain. While originally intended for analysis of control systems, there is no reason why it cannot be used to analyze audio filters (indeed, linear control systems *are* filters, just dealing with different frequency ranges).

Introduction to Root Locus

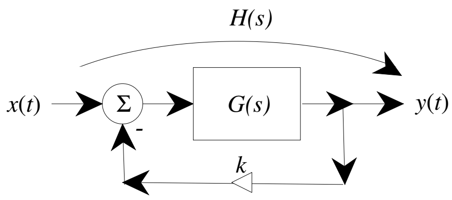

Figure 3: *A simple feedback system.*

Let's assume a system as shown in Figure 3, a simple feedback system with the transfer function $G(s)$ in the forward path. We know from block-diagram algebra that the total (closed-loop) transfer function is:

$$H(s) = \frac{G(s)}{1+kG(s)}$$

Now, if $G(s) = \frac{N(s)}{D(s)}$, then:

$$H(s) = \frac{N(s)}{D(s)+kN(s)}$$

If $G(s)$ is in the feedback path, then:

$$H(s) = \frac{D(s)}{D(s)+kN(s)}$$

Note that in both cases the poles are the same:

$$D(s) + kN(s) = 0 \tag{1}$$

$$k = -\frac{D(s)}{N(s)} = -G^{-1}(s)$$

Since k is real and positive, we see that the total root-locus (the locus of all points in the s-plane that are roots of eq. 1 as k traverses $[0, \infty)$) is all s for which $\angle G(s) = \pi$.

Two rules for root locus are immediately clear from eq. 1: for $k = 0$, the roots of eq. 1 are the roots of $D(s)$ (the poles of $G(s)$); and for $k \to \infty$, the roots of eq. 1 are the roots of $N(s)$ (the zeros of $G(s)$). Thus as k traverses $[0, \infty)$, the closed-loop poles start at the open-loop poles and head towards the open-loop zeros.

The rules for root locus were developed to aid in hand-drawing the loci, and can be found in any introductory book on control systems (such as Franklin & Powell 1994)[1]. Although it is now trivial to use computers to calculate root-loci via brute-force numerical root finders, familiarity with the rules and the common root-locus shapes allows one to use root-locus as a design tool.

The MoogVCF Analyzed

The Moog VCF is in the simple feedback form of Figure 3. Therefore, we can immediately draw the root locus for changes in the feedback gain.

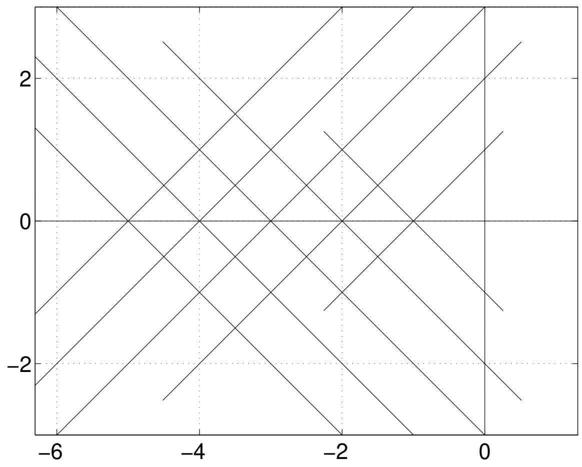

Figure 4: *Root-Loci for the Moog VCF, various ω_c, sweeping $k \in [0, 5]$.*

Root-Locus Rules state that the four coincident poles of the open-loop filters break away from the real axis at 45-degree angles and head to the zeroes at infinity along straight-line asymptotes, which in this case happen to be the same as the break-away lines. Thus, the root locus consists of these 45-degree lines that cross the imaginary axis at $\omega = \omega_c$, the open-loop pole location. Quick calculation also shows that the feedback gain at which this happens is $k = 4$. Thus one has a trivial corner-frequency control via the pole location of the cascaded one-pole filters.

Resonance Control: One can also evaluate a root locus of the VCF with ω_c, the open-loop pole location as the free variable.[2]

Algebraic solution of $(s+\omega_c)^4 + k\omega_c^4 = 0$ gives

$$\begin{aligned} s &= \omega_c \left(-1 \pm \sqrt{\pm j}\, k^{1/4}\right) \\ s &= \omega_c \left(-1 \pm k^{1/4} e^{\pm j\pi/4}\right) \end{aligned}$$

Which shows the 45-degree root-locus lines mentioned earlier. If we keep k constant, and look at the dominant poles (the ones that approach the $j\omega$ axis), we get:

$$\begin{aligned} s &= a e^{\pm j\alpha}, \text{where} \\ \alpha &= \frac{\pi}{2} + \tan^{-1}\left(\frac{\sqrt{2} - k^{\frac{1}{4}}}{k^{\frac{1}{4}}}\right) \end{aligned}$$

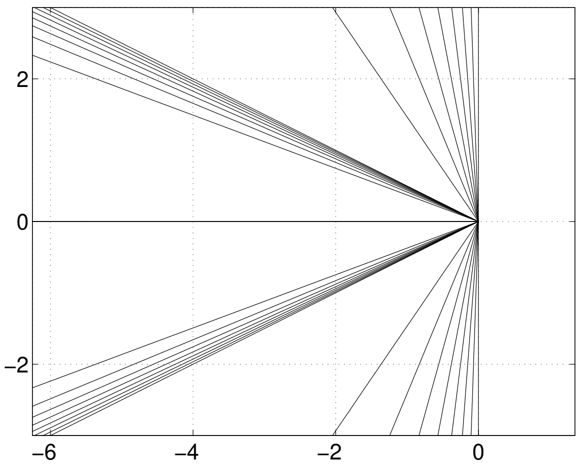

Figure 5: *Root-Locus vs. ω_c, various k.*

So sweeping ω_c while keeping k constant gives root-locus lines that keep a constant angle from the $j\omega$ axis. This gives the filters a constant Q across sweeps in ω_c, so that k becomes a Q control. Thus the Moog VCF has simple, uncoupled controls of corner frequency and resonance.

[1] The same rules apply in discrete-time filters (root-locus in z) as in continuous-time (root-locs in s), the only difference being the pole-location interpretation.

[2] Unfortunately, most of the standard root-locus rules do not apply anymore. The root-locus rules work for any system that can be put into the form $A(s) + cB(s) = 0$, so that the coefficients of s in the expanded polynomial are at most *affine* in c (linear plus an offset, i.e. the highest power of c in the polynomial is 1.). In the VCF, the above equation is 4'th-order in ω_c. Thus the root-locus can only be evaluated numerically, or in simple situations, solved algebraically, since there are no simplified rules for the patterns in this "higher-order" root locus (*yet*—work is being done to this end).

2.0.1 Definition of Q

For this paper, Q will be defined in terms of *pole location* rather than the center/bandwidth definition. In particular, it will be defined according to the impulse response of the poles (assuming dominance): "the number of cycles for the envelope of the impulse response to decay to $1/e^{\pi}$"(Morse, p. 25) . With dominance, pole location and impulse-response decay time are essentially equivalent. When we define Q in terms of the impulse response, we can arrive at a discrete-time definition of Q via the impulse-invariant transform $z = e^{sT}$, which gives us constant-Q pole locations as $z = r\, e^{\pm j\theta}, r = e^{-\alpha\theta}$, which are logarithmic spirals in the z-plane, and $Q = 1/(2\sin(\tan^{-1}(\alpha)))$ (thus the Q of the pole z_p is $Q(z_p) = [2\sin(\tan^{-1}(-\ln|z_p|/\angle z_p))]$).

3 Discretizing the VCF

It is desired to create digital filters with frequency and resonance controls as simple and efficient as those in the analog VCF. In particular, we desire filters whose controls (1) are uncoupled, (2) control useful parameters, such as frequency and Q, and (3) are efficient to control (not requiring expensive conversions, such as transcendentals, to get from the desired parameter to the actual control value). Filters based on the Moog VCF topology are explored here because it is hoped that at least some of the good features of the filter will translate well into the digital realm.

In order to preserve controllability, the continuous-time (CT) VCF equations must be translated to discrete-time (DT) using some transformation of the transfer function. as opposed to doing a impulse-response discretization, or a DT filter design based on the CT frequency response, because these methods typically aren't parameterizable, nor do they preserve any parameterization of the original system.

For similar reasons, the VCF's topology (cascaded one-pole filters with feedback around the whole loop) will be used for the DT filters. This means that the filter equations to be transformed will be those of the one-pole filters rather than the equations of the whole system. This keeps the controls simple, because otherwise, when the equations for the VCF are determined (i.e. by multiplying out the cascaded filters and collapsing the feedback), the resulting coefficients are no longer simple functions of the controls (including higher powers of the controls and divisions involving the coefficients), which destroys the efficiency of the controls.

Some popular transforms are the Backward-Difference Transform, the Bilinear Transform, and the Pole/Zero Mapping. These techniques accomplish the transform by applying some mapping to convert from the CT variable s to the DT variable z. The backward-difference mapping is $s \leftarrow (1+z^{-1})/T = (z-1)/(z\,T)$, ($T$ is the sampling period); the bilinear transform is $s \leftarrow 2(z-1)/(z+1)T$; and the pole/zero mapping is $z \leftarrow e^{sT}$ for the poles[3][Franklin & Powell 1990, Ch. 4].

The pole/zero mapping can't be used as a direct substitution for s into transfer functions, because the resulting equation in z is non-rational and thus not implementable. It can be used, however, to guide the design of an equivalent DT filter, with the poles and zeros in the positions described by the $z \leftarrow e^{sT}$ mapping. This can, still, increase the complexity of a design (and decrease the efficiency) because complex exponentials (or at least transcendentals) can easily crop up in the control equations of the new system.

Implementability:

An unfortunate fact in the discretization of the VCF topology is that most of the above-mentioned transforms will produce one-pole filters that have a delay-free path from input to output. That means when they are placed in the feedback loop, the system is unrealizable[4]. Thus the systems must be modified to make them realizable, typically by adding a unit delay into the loop. Unfortunately, this addition interferes with many of the features of the filters, including, most notably, causing the controls to no longer be uncoupled.

Therefore, a major part of the design process is finding transforms (or directly designing DT systems in the Moog VCF topology) that minimize the distortions required in the realizations.

Bilinear Transform

The bilinearly transformed onepole $\frac{s}{s+a}$ is:

$$G_1(z) = \frac{1}{2}(p+1)\frac{z+1}{z+p} \qquad \texttt{(Bilinear)}$$

where

$$p = \frac{a-2}{a+2}$$

so that

$$G(z) = \left(\frac{0.5\,(p+1)(z+1)}{(z+p)}\right)^4$$

and

[3] And the finite zeros, with all but one of the zeros at infinity being placed at $z=-1$.

[4] unless the feedback loop is collapsed with block-diagram algebra, but as mentioned earlier, this destroys the efficiency of the control.

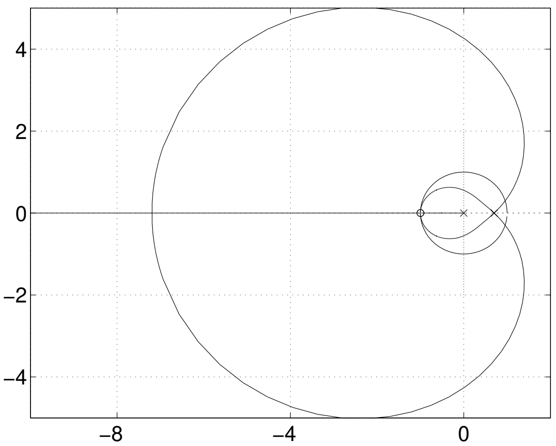

Figure 6: *The Complete Root Locus for a given p, bilinearly-transformed VCF.*

$$H(z) \;=\; \frac{G(z)}{1+kz^{-1}G(z)}$$

$G(z)$ has a delay-free path, so to implement this, a unit delay has been added to the loop. This kills the uncoupled nature of p and k for frequency and resonance control, see Figure 7 (If the controls were uncoupled, the curves would be horizontal).

We can see from the Figure 7 that k must be kept below 1.0 if one wants to sweep the whole range of p while keeping k constant yet stay stable at the high frequencies. Unfortunately, this causes the Q at low frequencies to be quite low. The current fix for the coupled controls is to use a "separation" table to scale the feedback gain as function of the pole location p:

$$k_{\text{actual}} = k_{\text{desired}}\text{Table}(p)$$

Where Table(p) is given by the top trace of Figure 7, and $k_{\text{desired}} \in [0,1)$.

The lower traces in Figure 7 are not simple scalings of the top trace, but the are rather close. This causes the Q along a given sweep of p to rise at very high frequencies, making this not exactly a constant-Q sweepable filter (see Figure 16). This inaccuracy is considered tolerable because it only becomes major for corner frequencies in the top octave, which are typically unused at $f_s = 44.1$ kHz, and if used, typically only for special effects, where total accuracy is not completely necessary.

To get exact constant-Q sweeps, table(p) would also have to be a function of Q, which vastly increases the storage requirements for the table.

Another table lookup must also me done if exact tuning is deemed necessary (Figure 8). Note that at low frequencies, the tuning curve is almost linear, so may be

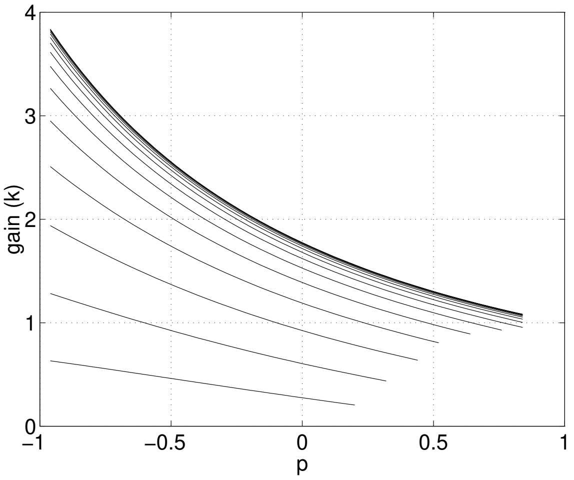

Figure 7: *Feedback gains vs. p to get various Q,* `Bilinear` *case.*

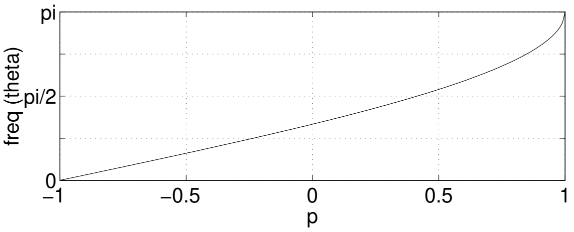

Figure 8: *Tuning curve at infinite Q,*`Bilinear`.

unnecessary. The use of tuning tables is often less of a problem for efficiency because in many cases exact tuning is needed at slower rates, such as only at the beginning and end of sweeps.

PZMap Onepole Placement

This case acts almost the same as the bilinear case, but it achieves realizability by removing one of the zeros at $z = -1$ instead of adding the unit delay (which puts a pole at $z = 0$). Otherwise it acts similarly, so will not be considered further.

Backward Difference Transform

The backward-difference transformed onepole $\frac{s}{s+a}$ is:

$$G_1(z) \;=\; (p+1)\frac{z}{z+p} \qquad \texttt{(Back-Diff)}$$

where

$$p \;=\; a+1$$

so that

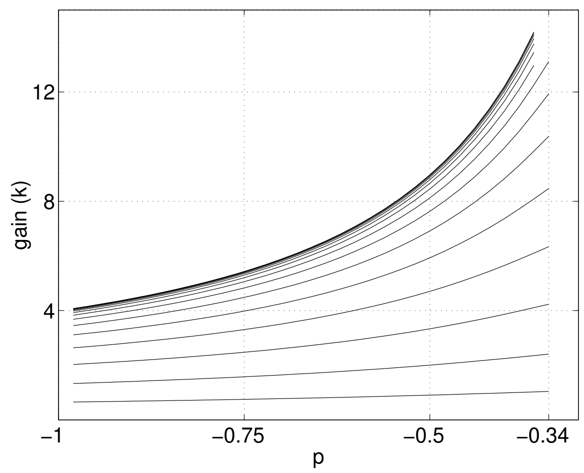

Figure 9: *Feedback gains for various Q,* `Back-Diff`.

$$G(z) \;=\; \left(\frac{(p+1)\,z}{(z+p)}\right)^4$$

Again, this requires an extra delay in the loop to become implementable, And, as in the bilinear case, a table is required for separability. This filter, however, can be used without a separation table with better results than in the bilinear case because, as we can see from Figure 9, the *Q falls* as p increases, for a given k. This allows the user to sweep p without worrying about stability as long as the Q at low frequencies is desirable. For many effects where exact Q isn't necessary, the variation in Q vs. p that this filter presents (Figure 17) may be acceptable.

If more closely constant Q is required, then the techniques described for the bilinear case (the use of a separation table) apply with similar results, although this filter may be able to be implemented slightly more efficiently because of numerator of the onepole is simpler (typically this affects the total system efficiency only slightly, since the bilinear case can be implemented very efficiently.)

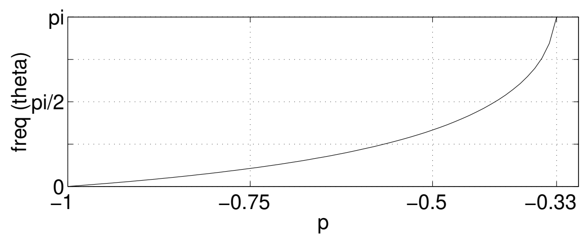

Figure 10: *Tuning curve at infinite Q,* `Back-Diff`.

The tuning curve is more drastic (Figure 10) than in the bilinear case, which makes the use of a tuning table more necessary in practice.

'Compromise' Version

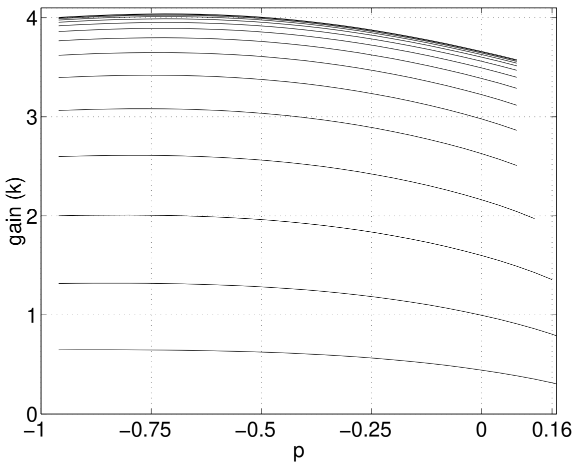

Figure 11: *Feedback gains for various Q,* `Compromise`.

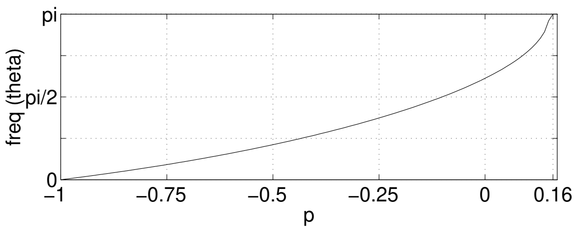

Figure 12: *Tuning curve at infinite Q,* `Compromise`.

The reader may have noticed that the gain curves for the two preceding cases have nearly opposite behaviors (for the bilinear, Q goes up with p when not using a separation table, and for the backwards difference, it goes down). The big difference between these two architecturally is the placement of the one-pole filters' zeros: the bilinear case places them at $z = -1$, and the backwards difference places them at $z = 0$. This suggests finding an intermediate position that may flatten out the gain curves and give p-sweeps more close to constant-Q.

A few eyeballed tries gave a zero position of $z = -0.3$:

$$G_1(z) \;=\; \frac{(p+1)}{1.3}\frac{z+0.3}{z+p} \qquad \texttt{(Compromise)}$$

$$G(z) \;=\; \left(\frac{(p+1)(z+0.3)}{1.3\;(z+p)}\right)^4$$

Again, a delay is put in the loop.

Referring to Figure 11 and Figure 18, we see that in the frequency range $[0, f_s/4]$ and for Q up to about 100, the filter is quite close to constant-Q and the controls p and k are almost uncoupled controls of frequency and Q, *without the use of a separation table*, although a tuning table may be necessary, as in all these cases.

An optimization could be performed to arrive at the "best" zero location, maybe even optimizing the four zeros to different locations.

Thoughts on Exact Constant-Q: It is likely that the auditory system is not *extremely* sensitive to variations in Q (i.e. the JND is probably large). Unfortunately I don't have any references on the subject other than a mention of a study on speech formant-width sensitivty [Smith86, p. 130]. If true, and if a number for JND (such as percentage) were found, then it would tell us how close to constant-Q we need to get in filters that don't exactly follow constant-Q sweeps, such as all the ones mentioned above. It would also help in the design of stopping conditions for optimization procedures that may be used to design or tweak these kinds of filters. It is likely that there is quite a bit of leeway in the variation of Q with corner frequency that we can tolerate.

On the other hand, JNDs for amplitude are quite small, and since messing with Q usually messes with amplitude (or loudness), this might place a tighter condition on Q. What we really need is a JND for resonance amplitude variation across corner frequency sweeps.

Comparisons

Root Loci: The Root Loci for the above-mentioned filters (Figures 13-15) are quite informative. These plots show dominant-pole locations versus sweeps of p and k (in the bilinear case, k is scaled with the separation table). Constant-Q pole locations are shown on the z-plane grid, so we can see how the filters deviate from constant-Q (at least at high frequencies). The loci also show how the tuning acts versus p.

Note that in all cases, $k = 0$ gives the positive real axis. Also note how the use of a separation table (Figure 13) guarantees stability, at the expense of the extra table lookup.

Constancy of Q: These plots (Figures 16-18) show the frequency ranges and Q ranges over which the filters approximate constant-Q (again, this is based on the location of the dominant poles). These show the Q as p varies, with k held constant at various values (except in the bilinear case, where the separation table is used[5]).

[5] Again, separation tables would also work in the other cases to get better curves, but the intention is to find filters for which the use of a separation table is unnecessary. It is necessary in the bilinear case for stability reasons.

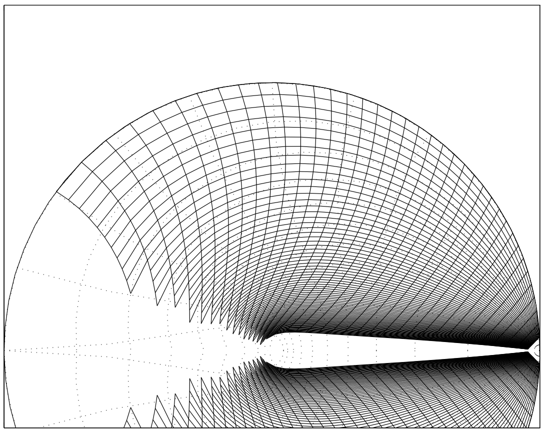

Figure 13: *Dominant pole locations for p and k sweeps,* `Bilinear` *with separation table.*

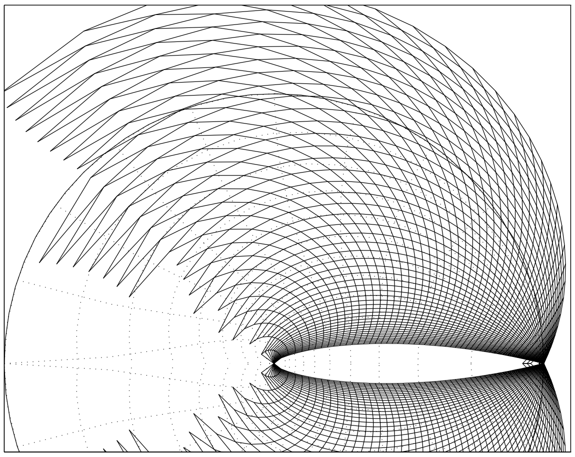

Figure 14: *Dominant pole locations for p and k sweeps,* `Back-Diff`*, no separation table.*

This type of plot is one of the more useful pieces of information when designing VCFs that are intended to be constant-Q.

Bode Plots: A complete set of Bode plots would take up much too much space, so instead a single p sweep is shown for each filter, with k held constant (with separation table in `Bilinear` case) at a value that gives a medium Q (Figures 19-21).

Oversampling: Another way to approach constant-Q is to oversample. Almost all filters of this type can be tweaked to act very well over a small frequency range. Oversampling reduces the range of desired frequencies significantly, thus making the VCF design problem easier. This is also understandable from the viewpoint that

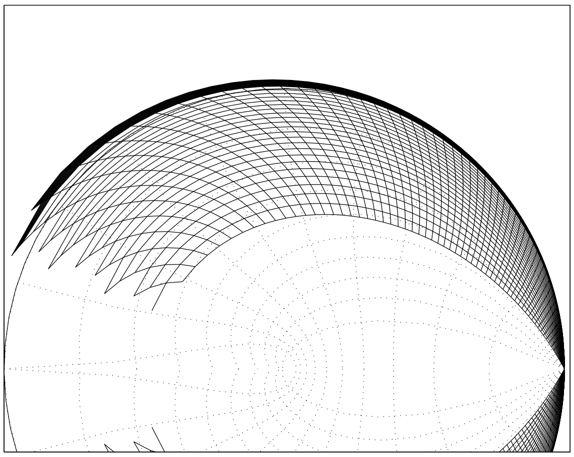

Figure 15: *Dominant pole locations for p and k sweeps,* `Compromise`*, no separation table.*

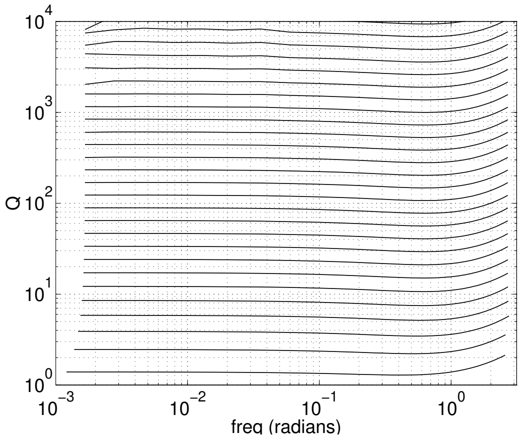

Figure 16: *Q vs. corner freq. for various (pre-scaling) k,* `Bilinear` *with separation table.*

highly oversampled systems are better approximations of continuous-time systems, because the region about $z = 1$ can be linearized down to a rectangular coordinate system (just like the CT coordinates) by the approximations

$$r\sin(\theta) \xrightarrow{\theta\to 0} r\,\theta$$
$$r\cos(\theta) \xrightarrow{\theta\to 0} r$$

furthermore, for $r \approx 1$,

$$r\sin(\theta) \xrightarrow{\theta\to 0} \theta$$
$$r\cos(\theta) \xrightarrow{\theta\to 0} r$$

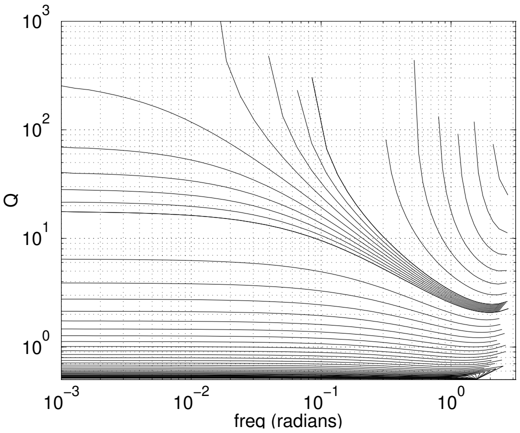

Figure 17: *Q vs. corner freq. for various k,* `Back-Diff`*, no separation table.*

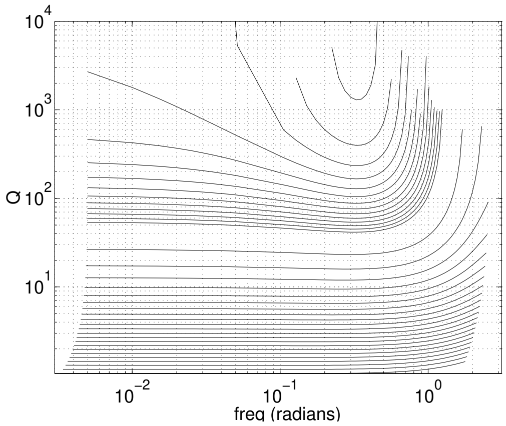

Figure 18: *Q vs. corner freq. for various k,* `Compromise`*, no separation table.*

On the other hand, oversampling is less efficient. For an oversampling factor of M, the oversampled filter is M times more expensive. This may be useful, however, if because of the oversampling, the cost of the filter can be significantly reduced (cf. the above argument that the design is simpler). Oversampling can also aggravate certain numerical errors, such as coefficient round-off, because all the poles become bunched up around $z = 1$, which increases sensitivity to the coefficient errors (Franklin & Powell 1990, p. 339)

Constant-Q Filters, Algebraic Derivation

If efficiency is not a problem, we can directly write the equations for the denominator of the desired filter:

2 poles:

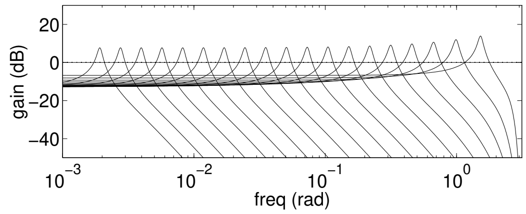

Figure 19: *Bode plots, constant pre-scaling k (medium Q), various p,* `Bilinear` *with separation table.*

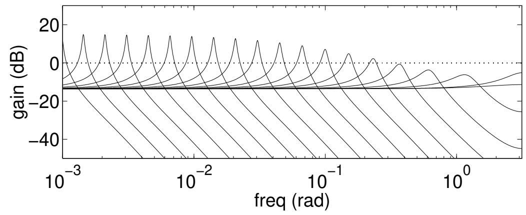

Figure 20: *Bode plots, constant k (medium Q), various p,* `Back-Diff`*, no separation table.*

$$(z - e^{-\alpha\theta}e^{-j\theta})(z - e^{-\alpha\theta}e^{j\theta})$$

4 poles:

$$(z - e^{-\alpha\theta}e^{-j\theta})(z - e^{-\alpha\theta}e^{j\theta})(z-a)(z-b)$$

In the four-pole case, the placement of the other two poles is a matter of design. If we compare with the root-locus of the bilinear case (Figure 6), which has the other two poles somewhere closer to the origin, it may be that some good choices for the other poles are: (1) both at $z = 0$, (2) same angle as the main poles, but with a reduced radius, (3) same angle, lower Q, or (4) at the same positions as the main poles (so they become repeated).

Multiplied out, the two-pole denominator is:

$$z^2 + 2e^{-\alpha\theta}\cos(\theta)z + e^{-2\alpha\theta}$$

As a first pass at making this efficient, we could use table lookups for the exponent and cosine, making for 2 table lookups (probably interpolated) and 2 multiplies for each new pole location.

A further efficiency increase comes in constant-rate frequency sweeps, where the update rate is constant. This satisfies the equation:

$$e^{-\alpha\theta(t)} = e^{-\alpha(a+bt)} = e^{-\alpha a}e^{-\alpha bt}$$

at $t = t_0 + \Delta t$,

$$e^{-\alpha\theta(t_0+\Delta t)} = e^{-\alpha a}e^{-\alpha bt_0}e^{-\alpha b\,\Delta t}$$

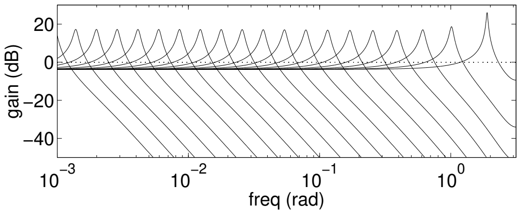

Figure 21: *Bode plots, constant k (medium Q), various p,* `Compromise`*, no separation table.*

So that at the denominator is:

$$z^2 + e_1(t)\cos(\theta(t))z + e_2(t)$$

where

$$e_1(t+\Delta t) = e_1(t)\delta_{e1}$$
$$e_2(t+\Delta t) = e_2(t)\delta_{e2} \text{ or } e_1(t+\Delta t)^2$$

and

$$\delta_{e1} = e^{-\alpha b\,\Delta t}$$
$$\delta_{e2} = \delta_{e1}^2$$

δ_{e1} need only be evaluated at the beginning of the sweep, thus we get rid of one table lookup per Δt. This technique can be used to smooth low-rate θ updates. If necessary, this method can also be used on α sweeps (or both).

4 Root-Locus Filters

Other patterns that show up in root-locus analysis can also be used to create useful sweepable filters. In particular, we can directly look for patterns that are useful for digital filters, rather than finding useful continuous-time patterns and then transforming the filters to discrete-time. We can call filters designed this way "Discrete-Time Root-Locus Filters"[6].

A common pattern in root-loci is a circle surrounding a zero. Circles are particularly interesting from a discrete-time perspective because of the region of stability in z, which is also a circle.

The Two-Pole Constant-Bandwidth Root-Locus Filter

By placing an open-loop zero on the origin, and two poles on the positive real line (so, using the notation of

[6]By analogy, the Moog VCF is a *Continuous-Time Root-Locus Filter*

Figure 3, $G(s) = z/(z-a)(z-b))$, we can get a root-locus (in k) that is a circle centered on the origin with a radius that is the geometric average of the pole locations. A particularly simple choice of pole locations is therefore to put them both on the desired radius: $G(s) = z/(z-r)^2$ (Figure 22).

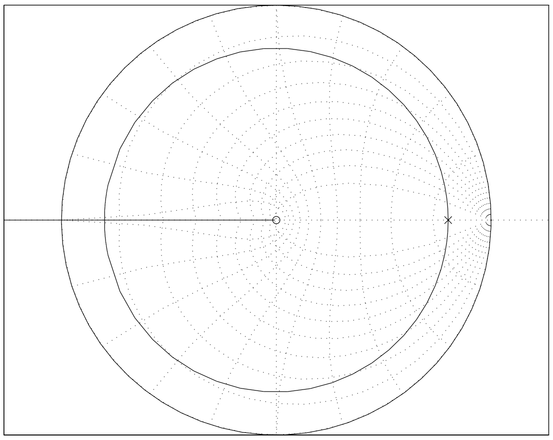

Figure 22: *Root locus of this 2-pole RL Filter, $r = 0.8$.*

We thus have two trivial controls: (1) k controls pole angle (corner frequency), and (2) the open-loop pole location controls the pole radius. Because the root locus is a perfect circle (this can be easily shown), the radius is constant over all frequencies (to the limits of the number system), so stability is not a problem. The controls are also completely uncoupled. frequency is related to k as $k = 2r(1-\cos(\theta))$, and radius as $r =$ (pole location).

This filter is not necessarily any more efficient than a direct-form filter (denom = $z^2 + 2rcz + r^2$), which also has uncoupled radius and angle controls with θ related to c as $c = \cos(\theta)$ — essentially the same control complexity. It may, however, have different numerical properties.

Root-Locus Filters Approximating Constant-Q

It is commonly held that constant-bandwidth filters are less useful than constant-Q filters. We can therefore modify the above root-locus filter to try to approximate constant-Q. The first pass is to note that at large Q, the constant-Q root trajectories look visually like circles, and shift the root- locus circle over to touch the unit circle at $z = 1$, like the constant-Q tracks do. This would give a pseudo-Q control with the open-loop zero location ($G(s) = (z-c)/(z-1)^2$), with zero locations nearer $z = 0$ giving higher average Q (here the open-loop poles would be fixed at $z = 1$). Unfortunately, root-locus rules state that the root-locus tracks must leave the real axis at $\pm 90°$,[7] so that at low frequencies, $Q \to \infty$, no matter where the zero is (see Figure 23).

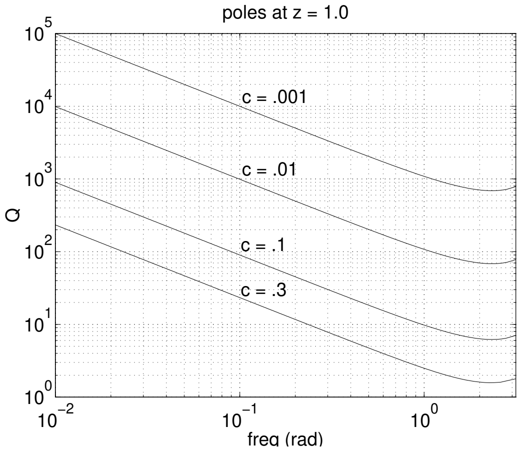

Figure 23: *Q vs. angle, various zero locations.*

The next modification would be to move the open-loop poles in from $z = 1$, so that Q doesn't go to ∞ at DC ($G(s) = (z-c)(z-(1-\varepsilon))^2$). This causes Q to go to zero at DC, rise quickly at low frequencies, and then settle in to the same pattern as above at high frequencies (see Figure 24).

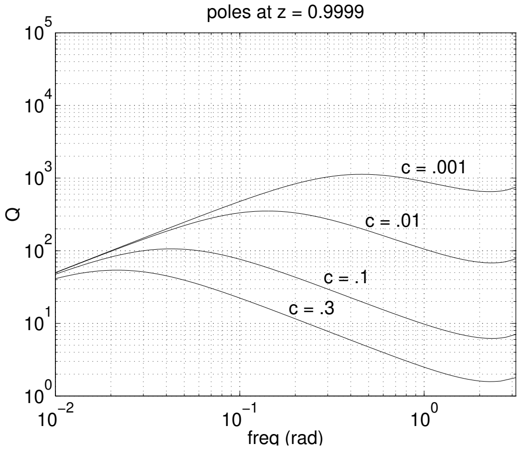

Figure 24: *Q vs. angle, various zero locations.*

Further flattening of Q can be achieved by adding pole/zero pairs inside the unit circle. This technique, well known in control-systems design, is used to locally warp the root locus. A pole/zero pair has a large effect

[7]the actual constant-Q tracks leave $z = 1$ at angles greater than $\pm 90°$

on $\angle G(z)$ near the pair (remember, the root-locus is all z for which $\angle G(z) = \pi$), but away from the pair, they cancel each other and have little effect on the root locus. One can control the effects of the pair by controlling their separation and distance from the locus (close together $\Rightarrow$ more localized effect $\Rightarrow$ most be closer to locus, but has stronger effect because of proximity to locus; further apart $\Rightarrow$ more widespread, but weaker effect due to usually being placed further from the locus).

A quick design using this effect is shown in Figure 25, its root locus is shown in Figure 26. By adding pole/zero pairs and shifting the main open-loop pairs, one can follow an ad-hoc optimization path and minimize the deviation from some desired Q. The filter shown was designed by eyeballed trial-and-error[8], but an optimization procedure could be designed.

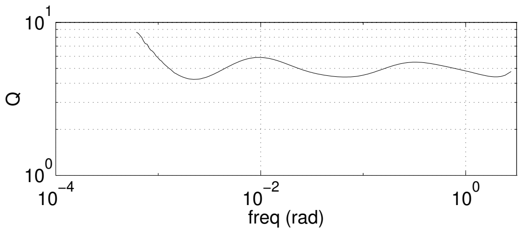

Figure 25: *Q vs. angle, eyeballed minimum-Q-error filter.*

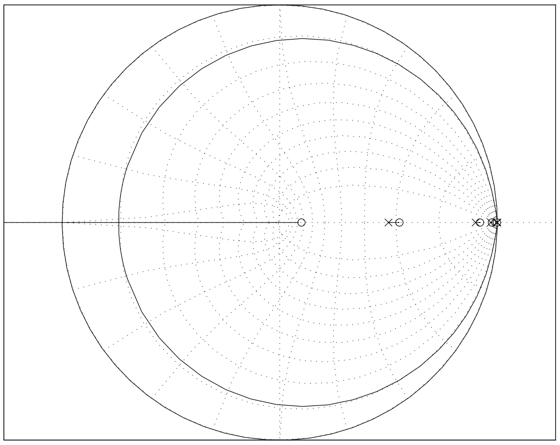

Figure 26: *root locus, eyeballed minimum-Q-error filter.*

Unfortunately, this technique doesn't easily lend itself to parameterizing Q, because a new optimization may need to be done for each Q (although a pattern could develop upon which a parameterization could be based). Also, it is likely not very efficient, due to the number of pole/zero pairs greatly increasing the order of the system.

[8]The pole/zero pairs were on the real axis to make things easier: poles at $z = [.5\ .9\ .97\ .9975\ 1\ 1]$, zeros at $z = [.1\ .55\ .92\ .975\ .9983]$. This particular $Q \approx 5$ is admittedly an easy design compared to a very high Q, but it serves as an example of the idea.

5 Conclusion

Implementability issues make the conversion of the Moog VCF to a digital form nontrivial. Once converted using standard techniques, the filter must be tweaked to recover some of the original features. Some transforms preserve features better than others, but best results come from redesigning the filter directly in the discrete domain. Methods from control-systems theory prove useful in this redesign. These methods also suggest new topologies that prove interesting.

References

[Hutchins 1975] Hutchins, B. 1975. *Musical Engineer's Handbook.* Ithaca, New York: Electronotes.

[Moog 1965] Moog, R. A. 1965. "A Voltage-Controlled Low-Pass High-Pass Filter for Audio Signal Processing." *Audio Eng. Soc. Convention*, Preprint 413(Oct.).

[Zwicker 1990] Zwicker, E. 1990. *Psychoacoustics.* New York: Springer Verlag.

[Franklin & Powell 1990] Franklin, G., J. D. Powell, M. L. Workman, 1990 *Digital Control of Dynamic Systems, 2nd Edition* Reading: Addison Wesley

[Franklin & Powell 1994] Franklin, G., J. D. Powell, A. Emami-Naeini, 1994 *Feedback Control of Dynamic Systems, 3rd Edition* Reading: Addison Wesley

[Morse 1981] Morse, P. 1981 *Vibration and Sound* Acoustical Society of America.

[Smith 1983] Smith, J. O. III 1983 "Techniques for Digital Filter Design and System Identification With Applicatio to the Violin." Ph.D./EE Thesis, Stanford University, CCRMA Report STAN-M-14

This paper can be found online at the web page: `http://www-ccrma.stanford.edu/~stilti/papers`