Complex Digital Filters Using Isolated Poles and Zeroes

Donald Daniel

January 18, 2008
Revised Jan 15, 2012

Abstract

The simplest possible explanation is given of how to construct software digital filters with arbitrary pole-zero configuration and specified gain or loss. These filters simulate the simple behavior of analog filters. This kind of digital filter is the most appropriate for computer programs that simulate the behavior of lumped physical systems, whether electrical, mechanical, or acoustical. It is also the most appropriate for some kinds of audio signal processing.

Digital filters commonly accept a stream of real numbers as input and produce a stream of real numbers as output. In the case of simulation of electrical filters, this stream of real numbers represents samples of voltage going into and coming out of a hardware filter.

Real linear lumped element analog hardware filters containing no distributed elements like delay lines respond to signals in a way that can be described completely by an assortment of poles and zeroes and a scalar multiplier. A single filter can have many poles and zeroes.

Digital filters model lumped element linear systems in the time domain. If you start with the physical configuration of the system, it is sometimes difficult to derive the poles and zeroes from the physical configuration. If modelling in the time domain is not necessary, it is much easier to model in the frequency domain using nodal simulation or two port simulation. If modelling in the time domain is necessary, a digital filter based on poles and zeroes is the best solution. We will describe here modelling in the time domain with poles and zeroes.

It is a mathematical identity that

$$cos(\omega t) = 0.5(e^{j\omega t} + e^{-j\omega t}) \tag{1}$$

The term j is imaginary, the square root of minus one. Because of the mathematical properties of filters it is more appropriate to describe them theoretically in terms of positive and negative imaginary radian frequencies $j\omega$ than real radian frequencies ω. Complex phasors $e^{j\omega t}$ are used instead of real sinusoids $cos(\omega t)$. However the data actually fed into the digital filters we will create does not need to be in the form of complex phasors.

Types of signals that might be applied to the filter are described with reference to the plane of complex numbers. The vertical, imaginary axis on the complex plane represents positive frequencies above the origin and negative frequencies below the origin. The horizontal, real axis on the complex plane is positive to the right and negative to the left. It represents decay rates to the left and expansion rates to the right. Both axes are zero at the origin where the axes cross. Most points on the complex plane represent expanding or decaying sine waves of various frequencies. The origin represents a steady unchanging voltage. Frequency response plots are determined by applying sinewaves that are steady, with amplitudes that are neither expanding nor decaying. Therefore frequency response plots represent input signals that lie on the vertical, imaginary axis.

Poles represent internal resonances in the filter. Zeroes represent signals the filter would block and not pass. Stable filters can have zeroes on both the left and the right side of the complex plane, but poles only on the left, with negative real part. Poles and zeroes may appear singly on the real axis, but a pole or a zero must appear as complex conjugate pairs if they are not on the real axis. The real component of a pair of poles or a pair of zeroes will be the same, but the imaginary component of a pair must have opposite sign. Thus one component of a pair is at positive frequency and the other is at a negative frequency of the same absolute value.

Digital filters have traditionally used clever tricks to represent complex poles and zeroes using real arithmetic. A single digital filter may represent many poles and zeroes. But some of these clever tricks are complicated to learn and limited in scope. An alternative approach presented here constructs a complicated filter as elemental filters in series, each representing an isolated pole or zero. The technique is simple and broad in scope.

Even though voltages in nature are always real, the poles and zeroes of real filters can be complex. The isolated poles and zeroes discussed here can be complex. Each elemental filter operates on a stream of complex numbers. This is discouraging to people whose data is a stream of real numbers. It need not be. If your age is 25 it is also 25+j0, with zero imaginary part. j is the square root of minus one. A stream of real numbers can have j0 added to each real number to make it a stream of complex numbers. Between successive poles and zeroes the imaginary part will be large. But if all complex poles or complex zeroes appear as complex conjugate pairs of poles or complex conjugate pairs of zeroes, as they must in nature, the stream of complex numbers coming out of the complete filter will have zero imaginary part except for roundoff error, and the imaginary part may be safely discarded to convert back to a stream of real numbers.

The signal stream passes one at a time through each pole and zero of the complete filter. It does not matter what order the poles and zeroes are arranged, the final result will be the same. However, to avoid roundoff error it is best to avoid having two zeroes in series. There should be a pole after each zero. Two poles in series does not caouse any roundoff problem. Because the signal passes through the elements in series, one might imagine that the filter would function as a delay line. It does not behave like a delay line because the signal passes through all the elements before the simulation advances to the next input sample.

An example of Oberon-2 computer code that connects poles and zeroes in series with a pole after every zero is given here. The input is pzxk, the output pzyk. The number of poles is fr.n and the number of zeroes is fr.nz.

```
PROCEDURE pzfltr*(t:REAL;pzxk:cx.complex;
VAR pzyk:cx.complex; VAR fr:fltrrec);
(*feed data through poles and zeroes in series*)
VAR k:LONGINT;tempxk:cx.complex;
BEGIN
tempxk:=pzxk;
FOR k:=1 TO fr.n DO
IF k < fr.nz+1 THEN
fr.z[k].xk:=tempxk;zero(t,fr.f1,fr.z[k]);
tempxk:=fr.z[k].yk;END(*IF*);
fr.p[k].xk:=tempxk;pole(t,fr.f1,fr.p[k]);
tempxk:=fr.p[k].yk;
END(*FOR*);
```

```
pzyk:=tempxk;END pzfltr;
```

The formulas presented here to represent poles and zeroes are only approximations of actual poles and zeroes. They are circular functions of the imaginary coordinate. They are not, however, circular functions of the real coordinate. The highest positive or negative frequency of the imaginary coordinate that can be represented is given by the Nyquist frequency, half the sample rate. The circular functions are continuous across the Nyquist frequency. The greatest positive frequency and the greatest negative frequency are the same frequency, with the same value of a pole or zero. The imaginary frequency coordinate could be thought of as being a frequency circle. The poles and zeroes are, respectively monotonically decreasing or increasing functions as you move away from the imaginary component of the pole or zero until you get to the opposite side of the frequency circle, where the functions are flared out so that they are flat, not increasing or decreasing. This flaring of the simulated poles and zeroes is an error relative to actual poles and zeroes. Therefore some kinds of filter will show distortion in the passband if they have poles or zeroes far from the center of the filter. The center frequency of a lowpass filter is zero. If the lowpass filter is wide enough to have poles one fourth of the way to the Nyquist frequency, distortion in the passband will be visible. Since a bandpass filter has passbands at both positive and negative frequencies, the center frequency of the entire bandpass filter is zero. Zeroes at the origin are an important part of any bandpass filter. If the passband is one fourth of the way to the Nyquist frequency, distortion will be visible, unless the passband is very narrow. This distortion is caused by flaring of the zeroes at the origin, and flaring of the conjugate poles at negative frequencies. Raising the sample rate can fix the problem. In the case of bandpass filters, another technique to fix the problem will be given later.

Let t be the time step, a constant equal to the time interval between samples. The sample number is k, the input stream is x_k, the output stream is y_k. The symbol ω represents $2\pi f$, where f is frequency. π is 3.14159... . The location of a pole or zero is represented by the real coordinate σ and the imaginary coordinate $j\omega$. This location on the complex plane can also be represented as the complex number $\sigma + j\omega$.

A pole at $\sigma_p + j\omega_p$ radians/sec is given by:

$$y_k = e^{\sigma_p t} e^{j\omega_p t} y_{k-1} + (1 - e^{\sigma_p t}) x_k \tag{2}$$

Arbitrary normalization has been chosen to make the gain unity at the nearest point on the frequency circle.

In the steady state $y_k = e^{j\omega t} y_{k-1}$. Rearranging, $y_{k-1} = y_k e^{-j\omega t}$. With this substitution, solving for the steady state gain of the pole as a function of frequency ω we have:

$$y_k/x_k = (1 - e^{\sigma_p t})/(1 - e^{\sigma_p t} e^{j(\omega_p - \omega)t}) \tag{3}$$

A zero at $\sigma_z + j\omega_z$ radians/sec is given by:

$$y_k = (x_k - e^{\sigma_z t} e^{j\omega_z t} x_{k-1})/(1 + e^{\sigma_z t}) \tag{4}$$

Arbitrary normalization has been chosen to make the gain unity at the farthest point on the frequency circle.

In the steady state $x_k = e^{j\omega t} x_{k-1}$. Rearranging $x_{k-1} = x_k e^{-j\omega t}$. With this substitution, solving for the steady state gain of the zero as a function of radian frequency ω we have:

$$y_k/x_k = (1 - e^{\sigma_z t} e^{j(\omega_z - w)t})/(1 + e^{\sigma_z t}) \tag{5}$$

The poles and zeroes are connected in series to create a filter. We choose a radian frequency ω_1 where we wish the filter gain to be unity. We take the reciprocal of the gains of the poles and zeroes at ω_1. The product of these reciprocals is the gain term for the filter. If we were to lump the gain terms at one point in the calculation, we would risk numerical difficulties. It is perhaps safer to apply each reciprocal gain term to each pole and zero respectively. By far the most important way to avoid numerical difficulties is to make sure that there is always a pole between two zeroes. This is possible because there are never more zeroes than poles in a realizable filter.

We can test the frequency response of the filter by feeding it a one, $(1+j0)$, followed by zeroes, then applying a fast Fourier transform (FFT) to the resulting output. This input signal will produce equal positive frequency and negative frequency components. The left and right halves of the output of the FFT will have to be swapped to put zero frequency in the middle, with negative frequencies on the left and positive frequencies on the right.

If it is important to test frequency response over a wide frequency range using a logarithmic frequency scale, a more direct approach can be used. Test sine waves at each frequency of the logarithmic frequency scale can be applied one at a time. Sufficient time must be allowed for the output to settle to a constant value before each measurement is made. However, it is not really necessary to test the filter for a log plot of the response. The pole and zero gain formulas are accurate, so the normalized product of the gain formulas will accurately predict the frequency response without actually measuring it.

We will want to compare the FFT plot of our filter with the ideal transfer function. The transfer function is the ratio of the output of the complete filter to its input. It is in the form of a the product of all of the zero terms divided by the product of all of the pole terms. The variable s represents a complex number that could be anywhere on the complex plane. For zeroes z and poles p the transfer function can be factored into terms $(s - z)$ in the numerator and $(s - p)$ in the denominator. As an example, this is the transfer function for a system with 2 zeroes and 3 poles:

$$T.F. = K\frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)(s - p_3)} \tag{6}$$

If there are no zero terms, the numerator will be the number one. The factor K is a scalar multiplier needed to normalize the gain to unity in the passband. We will want to express the transfer function as a frequency response plot, so it will be evaluated along the $s = j\omega$ axis.

We do not know the factor K in advance. We can eliminate K and use another way to normalize the transfer function. The transfer function must be normalized in the same way as the digital filter to compare them. We replace each zero factor in the numerator by $(j\omega - z)/(j\omega_1 - z)$, and each pole factor in the denominator by $(j\omega - p)/(j\omega_1 - p)$. The normalization we have chosen for both the digital filter and the transfer function results in unity gain at $j\omega_1$. We can multiply by an arbitrary number to achieve any other desired gain or loss.

Now, final comments about lumped filters. Real analog lowpass all pole filters with 10 poles will have noticeable delay in their response to a step function input. Our simulation will have the same delay. This does not mean that our simulation is behaving like a delay line because the poles are in series.

We have noted that if the pole/zero pattern is symmetrical about the real axis, a real input will result in a real output, with zero imaginary part. However, the frequency spectrum will not be real. The spectrum will have a real part that is an even function of frequency, with equal values at positive and negative frequencies. The spectrum will have an imaginary part that is an odd function of frequency, with equal values of opposite sign at positive and negative frequencies.

Thus far we have discussed a single filter with as many poles and zeroes as desired. Some applications, such as a filter with separated narrow spike passbands might be better implemented as separate filters with the same input applied to all, and the outputs summed.

While real signals can be represented as complex signals with zero imaginary part, there are applications involving approximations of reality which require complex signals with non-zero imaginary parts [1].

Now, an application for which we are willing to tolerate a non-zero imaginary part in the time domain. The computational load of a simulation is high if a high sample rate is required to simulate high frequencies. If the bandwidth of interest is small, but centered at a high frequency, it can be moved and centered at zero, requiring a much lower sample rate and lower computational load. If a high frequency signal is represented by a stream of complex numbers with zero imaginary part, equation (1) shows that it can be represented by phasors at positive and negative frequencies. But if we use only the positive frequency phasors, and shift the center frequency of the stream of phasors to zero, we can greatly reduce the sample rate. It is easier to get a stream of positive phasors without negative phasors if the data is computed theoretically rather than measured experimentally. If the center frequency of the positive phasors is $j\omega_c$ with amplitude and frequency modulation about that center frequency, then multiply the stream of phasors by a stream of phasors at a fixed amplitude of one and a fixed frequency of $-j\omega_c$ to reduce the center frequency to zero. Reduce the sample rate after this. Any amplitude modulation or frequency modulation of the stream about its center frequency will be preserved. Working with phasors can be facilitated by Euler's formula:

$$e^{j\omega t} = cos(\omega t) + j\, sin(\omega t) \tag{7}$$

This technique is applicable to very narrow bandpass filters where we are only interested in a small bandwidth around the pass band. Bandpass filters are typically designed by starting with a low pass prototype design consisting of an assortment of poles and zeroes symmetrical about zero frequency. A mathematical procedure known as a lowpass to bandpass transformation creates a bandpass filter by moving the lowpass assortment to be centered at the bandpass frequency, a conjugate assortment centered at minus the bandpass frequency and additional zeroes at the origin. But since all these have an effect at the passband, the assortments at the two passbands are not identical copies of the lowpass prototype. The assortments of poles and zeroes at the passbands must be distorted in order to create a frequency response at the passband that approximates the response of the lowpass prototype. We have shifted the positive phasor stream to zero, and would like to shift the positive assortment to zero, but in the absence of the conjugate assortment, and the zeroes that were originally at the origin, it would not create the response of the lowpass prototype, so instead we use the original lowpass prototype centered at zero.

For example filters with a link to all the details of the calculations see [2]. For the derivation of the pole see [3]. For the derivation of the zero from the pole see [4].

Acknowledgments

The author wishes to acknowledge assistance received from Chris Chopper, Grant LaBarre and Ken Wetzel. The work was performed at the Naval Air Warfare Center, China Lake, CA just before the author retired in 1998.

References

[1] T. H. Crystal and L. Ehrman, "The Design and Applications of Digital Filters with Complex Coefficients", *IEEE Trans. Audio and Electroacoustics,* vol. AU-16 pp. 315-320, September 1968.

[2] D.Daniel, "Example Digital Filters" www.waltzballs.org/ other/ engr/ pzex.html, Jan 2008.

[3] D.Daniel, "Derivation of Complex IIR Pole" www.waltzballs.org/ other/ engr/ fltr4.pdf, Aug 2008.

[4] D.Daniel, "The Relation between IIR pole and FIR Zero", www.waltzballs.org/ other/engr/fltr3.pdf, Feb 2008.