Time Domain Analysis and Design of Discrete Time Biquadratic Filters

Robin Schmidt (www.rs-met.com)

June 14, 2019

We consider the problem of finding a closed form expression of the impulse response of a discrete time biquadratic filter that realizes the transfer function:

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

and the inverse problem of finding the coefficients for such biquad filters from time-domain specifications of the impulse response.

Pole Locations

The two poles of the filter are the roots of the denominator and their locations, denoted as p_1, p_2, are given by:

$$\boxed{p_{1,2} = -\frac{a_1}{2} \pm \sqrt{\frac{a_1^2}{4} - a_2}} \tag{2}$$

Depending on whether the value under the square-root is positive, zero or negative, we will have either two distinct real poles, two coinciding real poles or a pair of complex conjugate poles respectively. Reversely, the a-coefficients may be computed from the poles as:

$$\boxed{a_1 = -(p_1 + p_2), \quad a_2 = p_1 p_2} \tag{3}$$

Impulse Response

The impulse response that corresponds to our transfer function $H(z)$ in (1) is given by the inverse z-transform of $H(z)$. To find it, we start by multiplying numerator and denominator of (1) by z^2, such that:

$$H(z) = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2} \tag{4}$$

This is an improper rational function in z because the numerator order is not strictly less than the denominator's. Generally, improper rational functions with numerator order N and denominator order M can be expressed as a sum of a polynomial of order $N - M$ and a strictly proper rational function where the denominator's order is less than the numerator's. So, in our case, $N - M = 0$ which means, we may express $H(z)$ as a sum of polynomial of order zero, i.e. a constant, and a strictly proper rational function. This decomposition can be done by means of polynomial long division an gives:

$$H(z) = b_0 + \frac{(b_1 - b_0 a_1)z + (b_2 - b_0 a_2)}{z^2 + a_1 z + a_2} \tag{5}$$

Multiplying numerator and denominator by z^{-2} and factoring out z^{-1} gives:

$$H(z) = b_0 + z^{-1} \frac{(b_1 - b_0 a_1) + (b_2 - b_0 a_2)z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{6}$$

We define the second term (excluding the z^{-1} factor) as the two-pole-one-zero transfer function:

$$G(z) = \frac{c_0 + c_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \quad c_0 = b_1 - b_0 a_1, \quad c_1 = b_2 - b_0 a_2 \tag{7}$$

so we can write:

$$H(z) = b_0 + z^{-1} G(z) \tag{8}$$

To find the inverse z-transform of that expression, we have to find the inverse z-transform of both terms separately and add them up (because the z-transform is linear). From a table of z-transforms, we find the transform pair: $\delta[n] \leftrightarrow 1$. By linearity, we conclude that $b_0 \delta[n] \leftrightarrow b_0$. Furthermore, we note that a multiplication in the z-domain by z^{-1} corresponds to a unit delay in the time domain. So we can write down our full impulse response as:

$$\boxed{h[n] = b_0 \delta[n] + g[n-1]} \tag{9}$$

where $g[n]$ is the impulse response that corresponds to the transfer function $G(z)$ and $g[n-1]$ is the unit-delayed version thereof. The impulse response $g[n]$ will be the subject of the next section. If we are given the two-pole-one-zero coefficients c_0, c_1 and want to go back to our biquad coefficients b_1, b_2, the relations are:

$$b_1 = c_0 + b_0 a_1, \quad b_2 = c_1 + b_0 a_2 \tag{10}$$

Two-Pole-One-Zero Filters

We consider the problem of finding the impulse response $g[n]$ that corresponds to the transfer function:

$$G(z) = \frac{c_0 + c_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{11}$$

Case 1: Two Distinct Poles

If the value under the square-root in (2) is nonzero, the poles are distinct (i.e. $p_1 \neq p_2$), and the transfer function $G(z)$ in (11) can be expanded into partial fractions as:

$$G(z) = \frac{c_0 + c_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{r_1}{1 - p_1 z^{-1}} + \frac{r_2}{1 - p_2 z^{-1}} \tag{12}$$

which leads us to the system of 2 simultaneous equations:

$$c_0 = r_1 + r_2, \quad c_1 = -(r_1 p_2 + r_2 p_1) \tag{13}$$

which we may solve for the residues r_1, r_2 as:

$$r_1 = \frac{c_1 + c_0 p_1}{p_1 - p_2}, \quad r_2 = c_0 - r_1 \tag{14}$$

To find the impulse response, we need the inverse z-transforms of the two partial fractions in (12). From a table of z-transforms, we find the transform pair: $a^n u[n] \leftrightarrow \frac{1}{1-az^{-1}}$, where $u[n]$ is the unit step function which is 1 for $n \geq 0$ and zero otherwise. The multiplication by the unit step function just zeros the function for negative sample indices n, which is to say, that our filter's impulse response is causal. We just keep this in mind and will consider only $n \geq 0$, so we may subsequently drop the unit step function from the equations for notational convenience. Due to linearity of the z-transform, we can apply this to both partial fractions separately multiply in the residues and add the results to obtain the impulse response $g[n]$ that corresponds to the transfer function $G(z)$ as:

$$\boxed{g[n] = r_1 p_1^n + r_2 p_2^n} \tag{15}$$

Subcase 1: Both Poles Real

If the value under the square-root in (2) is positive, we have two distinct real poles and the residues will also be real. In this case, (15) represents a weighted sum of two exponential functions, the growth- or decay rate of which is determined by the poles p_1, p_2. In most practical applications, we will probably have $|p_1| < 1, |p_2| < 1$, corresponding to exponentially decaying functions, i.e. stable filters. If the value of one of the poles is negative, the impulse response will have alternating signs of successive samples for the respective exponential term. If both poles are positive, using $a^x = e^{\ln(a)x}$, we can also write the impulse response as:

$$g[n] = r_1 e^{-\alpha_1 n} + r_2 e^{-\alpha_2 n}, \quad \alpha_1 = -\ln(p_1), \alpha_2 = -\ln(p_2) \tag{16}$$

where the α values are decay rates and their reciprocals are normalized time constants, i.e. the number of samples required for the exponential to decay down to $1/e$. Such time constants are often used in the description of exponentially decaying functions. If a pole p_i is negative, a corresponding sign alternation factor $(\text{sign}(p_i))^{n+1}$ would have to be included in the respective term [TODO: verify this experimentally].

Applications Such sums of two weighted exponential functions may be useful for envelope generators. For example, in some plucked or struck acoustic instruments, we see a fast early decay and slower late decay, which could be modeled straightforwardly by this kind of envelope shape. A particularly interesting case is the one where $r_2 = -r_1$. Assuming that $\alpha_2 > \alpha_1$, i.e. the second term decays faster than the first, we see a kind of attack-decay curve starting at 0, going through some peak and then decaying away with an asymptotic decay rate determined by α_1. Given a fixed decay α_1, the location of the peak will be determined by α_2 and its height by the mutipliers. It's possible to calculate the location and height of the peak and vice versa, it's possible to calculate an appropriate α_2 given a desired peak location. The height can then be easily adjusted by choosing the multiplier r_1. Putting this together, we may create an attack-decay envelope with attack time defined by the peak location and an asymptotic decay rate given by α_1.

Subcase 2: A Pair of Complex Conjugate Poles

If the value under the square-root in (2) is negative, the two poles form a pair of complex conjugates such that $p_1 = p, p_2 = \bar{p}$. Let p be expressed in polar nation as $p = Pe^{j\omega}$. Then (3) can be expressed as:

$$\boxed{a_1 = -2\Re(p) = -2P\cos(\omega), \quad a_2 = P^2} \tag{17}$$

If the poles are complex conjugates and the numerator coefficients are real, it turns out that the residues r_1, r_2 will also form a complex conjugate pair, so we may write the impulse response (15) as:

$$g[n] = rp^n + \bar{r}\bar{p}^n \tag{18}$$

Let the residues also be expressed in polar notation as $r = Re^{j\phi}$. Then:

$$g[n] = Re^{j\phi}(Pe^{j\omega})^n + Re^{-j\phi}(Pe^{-j\omega})^n \tag{19}$$

which, after some simplification, becomes:

$$g[n] = 2RP^n\cos(\omega n + \phi) \tag{20}$$

This represents an exponentially enveloped cosine function with normalized radian frequency ω, some phase offset ϕ, an overall amplitude of $2R$ and an exponential envelope that is determined by P. In practice, we would choose a pole radius $P < 1$, such that we see an exponential decay rather than growth, corresponding to a stable filter. We may also write this impulse response in the more familiar form of a damped sinusoid:

$$\boxed{g[n] = Ae^{-\alpha n}\sin(\omega n + \varphi)} \tag{21}$$

where:

$$\boxed{A = 2R, \quad \alpha = -\ln(P), \quad \varphi = \phi + \frac{\pi}{2}} \tag{22}$$

Analysis Algorithm Putting it all together, we arrive at the following algorithm to compute the damped sine parameters $A, \alpha, \omega, \varphi$ in (21) from the filter coefficients c_0, c_1, a_1, a_2: First, we compute the pole radius P, the normalized radian frequency ω and the actual complex pole location p from the feedback coefficients by means of (17):

$$\boxed{P = \sqrt{a_2}, \quad \omega = \arccos\left(-\frac{a_1}{2P}\right), \quad p = Pe^{j\omega}} \tag{23}$$

Next, we use (14) to compute the residue r - its radius R and angle ϕ are related to the amplitude A and startphase φ via (22), so:

$$\boxed{r = \frac{c_1 + c_0 p}{p - \bar{p}}, \quad A = 2|r|, \quad \varphi = \angle r + \frac{\pi}{2}} \tag{24}$$

Design Algorithm If we have a design specification for a filter in terms of $A, \alpha, \omega, \varphi$ and want to compute the filter coefficients c_0, c_1, a_1, a_2, we proceed as follows: Equations (22) and (17) are used to compute the pole radius P and the feedback coefficients a_1, a_2:

$$\boxed{P = e^{-\alpha}, \quad a_1 = -2P\cos(\omega), \quad a_2 = P^2} \tag{25}$$

For the feedforward coefficients c_0, c_1, we use (13) with $r_1 = r = Re^{j\phi}, r_2 = \bar{r} = Re^{-j\phi}$ and $A = 2R, \varphi = \phi + \frac{\pi}{2}$ from (22). After some simplification, this leads to:

$$\boxed{c_0 = A\sin(\varphi), \quad c_1 = AP\sin(\omega - \varphi)} \tag{26}$$

As it stands, the algorithm requires 1 evaluation of the exponential function, 1 cosine evaluation and 2 sine evaluations. Some programming languages provide a function that computes sine and cosine of the same argument simultaneously without additional cost. If we have such a function and are concerned about computational efficiency, we may get rid of one of the sine evaluations, by evaluating sine and cosine of φ, ω and apply an addition theorem for the computation of $\sin(\omega - \varphi)$. This leads to the optimized computations for the feedforward coefficients:

$$s_\varphi = \sin(\varphi), c_\varphi = \cos(\varphi), s_\omega = \sin(\omega), c_\omega = \cos(\omega), \quad c_0 = As_\varphi, c_1 = AP(s_\omega c_\varphi - c_\omega s_\varphi) \tag{27}$$

which replaces the 1 cosine and 2 sine evaluations by 2 sine-and-cosine evaluations.

Spiraling Phasor Implementation An alternative implementation structure for such damped sinusoid filters is possible which allows for an intuitive interpretation of the filter's internal state variables. To see this, we recall that a multiplication of a complex number z by some factor p results in a new complex number $z' = pz$ that can be seen as z, rotated by $\angle p = \omega$ and scaled by $|p| = P$. Thus, iterating this multiplication by p on a complex number z like $z[n] = pz[n-1]$ results in a complex phasor that rotates with a speed determined by ω and thereby possibly shrinks (or grows) in exponential manner at a rate determined by P. So, the overall trajectory of our complex number z is an exponential spiral. In order to extract a decaying sinusoid with arbitrary amplitude and phase from that spiraling phasor, we could set the initial value of $z[0] = 1$, iterate the multiplication by p and at each iteration extract a real number as output value by forming an appropriate linear combination of z's real and imaginary parts. Noting that any sinusoid with frequency ω and arbitrary amplitude A and phase φ may be expressed as:

$$A\sin(\omega n + \varphi) = A_s \sin(\omega n) + A_c \cos(\omega n) \tag{28}$$

where

$$\boxed{A_s = A\sin(\varphi), \quad A_c = A\cos(\varphi)} \tag{29}$$

we conclude that A_s is the appropriate multiplier for z's imaginary part and A_c for z's real part, if we choose an initial value of unity for z. But in a digital filter, we don't choose an initial value and then just iterate on internal states - instead we have input and output signals. If we replace our iteration $z[n] = pz[n-1]$ by incorporating an input signal $x[n]$, such that $z[n] = pz[n-1] + x[n]$, we will see the same exponential spiral, if the input signal $x[n]$ is given by the unit impulse. Putting all of this together, we may implement our decaying sinusoid filter as:

$$\boxed{z[n] = pz[n-1] + x[n], \quad y[n] = A_c \Re(z[n]) + A_s \Im(z[n])} \tag{30}$$

The advantage of such an implementation is, that at any time instant n, we may interpret the internal state z of our filter as a complex number which is going to be rotated and scaled by our complex feedback coefficient p. If we now suddenly change the value of p during recording the filter's impulse response, we will just see the output signal switch to a new frequency and/or decay rate. In other implementation structures, we might expect to see artifacts from sudden parameter changes, in particular, if such parameter changes affect feedback coefficients. Thus, our spiraling phasor implementation structure should have advantages under time-varying conditions.
[TODO: verify this experimentally]

Applications The algorithm to design filters that produce damped sinusoids according to specifications can be useful because damped sinusoids occur naturally in many vibrating systems. So, they can be valuable building blocks in the simulation of such systems. Subtracting the outputs of two of such filters with different decay rates while all other parameters match, we may create a sinusoid with an attack-decay shaped envelope as impulse response.

Case 2: Two Equal Poles

If the value under the square-root in (2) equals zero, the two poles are equal (i.e. $p_1 = p_2 = p$). In this case, our partial fraction expansion in (12) is no longer valid. Instead, the transfer function $G(z)$ in (11) has to be expanded into partial fractions as:

$$G(z) = \frac{c_0 + c_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{r_1}{1 - p z^{-1}} + \frac{r_2}{(1 - p z^{-1})^2} \tag{31}$$

from which we may compute the c coefficients as:

$$c_0 = r_1 + r_2, \quad c_1 = -r_1 \tag{32}$$

or - vice versa - compute the residues from the coefficients as:

$$r_1 = -c_1, \quad r_2 = c_0 + c_1 \tag{33}$$

The first term in our impulse response is $r_1 p^n$ as before. For the second term, we find the z-transform pair $n a^n u[n] \leftrightarrow \frac{a z^{-1}}{(1-a z^{-1})^2}$ from which we conclude: $r_2 (n-1) p^{n-2} u[n-1] \leftrightarrow \frac{r_2}{(1-p z^{-1})^2}$. So, our full impulse response may be written as:

$$g[n] = \begin{cases} 0 & n < 0 \\ r_1 & n = 0 \\ r_1 p^n + r_2 (n-1) p^{n-2} & n > 0 \end{cases} \tag{34}$$

For $n > 0$, a second term kicks in which represents an exponential decay (or growth) multiplied by a linear growth starting at 0 for $n = 1$ [TODO: verify this experimentally]. I currently don't see any obvious applications for this kind of impulse response but the case was included for completeness.

Conclusion

We have seen that the impulse response $h[n]$ of a biquad filter with transfer function $H(z)$ can be expressed a sum of a scaled unit impulse $b_0 \delta[n]$ and a unit-delayed impulse response $g[n-1]$ of a two-pole-one-zero filter: $h[n] = b_0 \delta[n] + g[n-1]$. The impulse response $g[n]$ can have the form of a a weighted sum of exponentials with different decay times, a damped sinusoid or a weighted sum of an exponential and the same exponential delayed with linear growth multiplied in. These types of two-pole-one-zero impulse responses can be useful in their own right, in particular, the damped sine type. Design formulas have been given to compute the damped sine filter coefficients from a given set of specifications.