Lesson 4: Feedback Filters

Introduction

Required Reading:
Ch. 5, 13 (§7 & 8)

In this lesson, I will continue to introduce filters, this lesson focusing on *feedback filters*, in which previous outputs are combined with new inputs to produce new outputs. I will talk about their structure, function, and the differences between feedforward and feedback filters. We will see that sometimes it is better to combine these two kinds of filters. I will also talk a bit about digital filter design. After this lesson, you should understand important concepts like the *poles* (as compared to *zeros*) of a transfer function, *impulse response*, and *bandwidth*. You should know about features of feedback filters and special types of such filters, such as *resons*. You should be able to design simple digital filters, implement them on a computer and use them to solve some simple signal processing problems.

Poles

A simple feedback filter's signal flowgraph is shown in text book chapter 5, figure 1.1. Compared to the feedforward one in figure 2.1 (chapter 4), we notice that instead of combining the input signal with a delayed version, here the output signal is delayed and "fed back" to be combined with the input. This feedback processing is expressed by

$$y_t = x_t + a_1 y_{t-1} \tag{4-1}$$

which is the equation for a simple feedback filter with one delayed component. $a_1 y_{t-1}$ is the feedback term. You can see that the output at $t-1$ is used to compute the output at time t.

From equation (4-1) and the delay operator z^{-1}, we can get the filter's transfer function as follows:

$$\begin{aligned} Y &= X + a_1 z^{-1} Y \\ Y[1 - a_1 z^{-1}] &= X \\ Y &= \frac{1}{1 - a_1 z^{-1}} X \end{aligned} \tag{4-2}$$

Finally we have the transfer function

$$H(z) = \frac{1}{1 - a_1 z^{-1}} \tag{4-3}$$

$$= \frac{z}{z - a_1} \tag{4-4}$$

The magnitude response is

$$|H(z)| = \left| \frac{z}{z - a_1} \right| = \left| \frac{1}{z - a_1} \right| \tag{4-5}$$

($|z| = 1$ because it is a phasor, and so its magnitude is always one).

The values of z that make the denominator of the transfer function zero (where the transfer function becomes infinite) are called its *poles*. In (4-5), there is one pole at $z = a_1$. In general, a_1

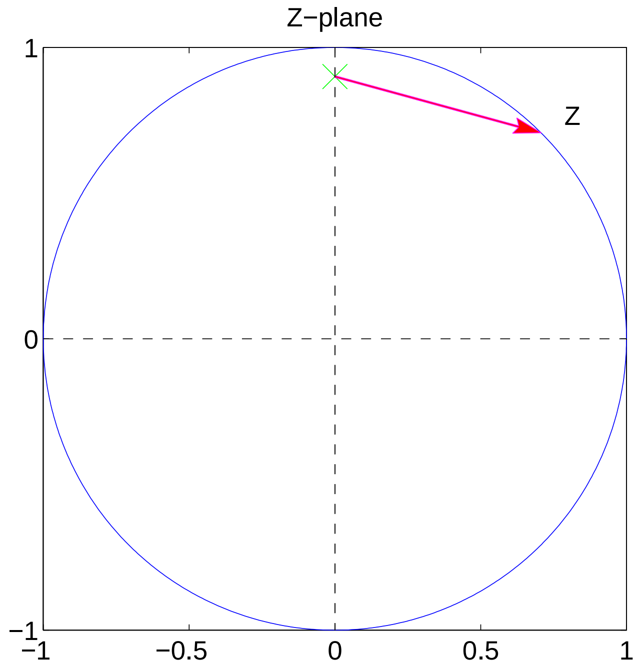

Figure 4.1: One pole in the z-plane, located at $r = 0.9$ and $\omega_0 = \pi/2$.

is not on the unit circle, and so the phasor z approaches it but is never equal to it. Just as we did for zeros, we can draw a line from the pole to the unit circle to indicate the distance between z and the pole. However, now this distance is in the denominator. Therefore, instead of $H(z)$ having a notch when z nears a zero, it has a *peak* when z nears a pole.

Example 1 For the one pole filter (4-1), $a_1 = re^{j\omega_0}$. Figure 4.1 shows this pole when $r = 0.9$ and $\omega_0 = \pi/2$. Figure 4.2 shows its magnitude response. As expected, $|H(\omega)|$ has a "hill" near $\omega_0 = \pi/2$. As r moves closer to one, the hill becomes steeper.

Stability

Let's examine equation (4-1) again. Note that y_t doesn't only depend on x_t. This means that, even if the input signal is zero after some initial value, this feedback term can continue to have a nonzero value and so the filter can still produce some output. This is a major difference from feedforward filters, where the output depends only on the input. An example is shown in text book equation (1.2). In this case, the input signal is a *unit impulse*,

$$x_t = \delta_t = \begin{cases} 1 & t = 0 \\ 0 & t > 0 \end{cases} \tag{4-6}$$

which means it only has a value of one at the beginning, and then turns off. The output of any filter for a unit impulse is called its *impulse response*. For the filter in (4-1) with coefficient $a_1 = 0.5$,

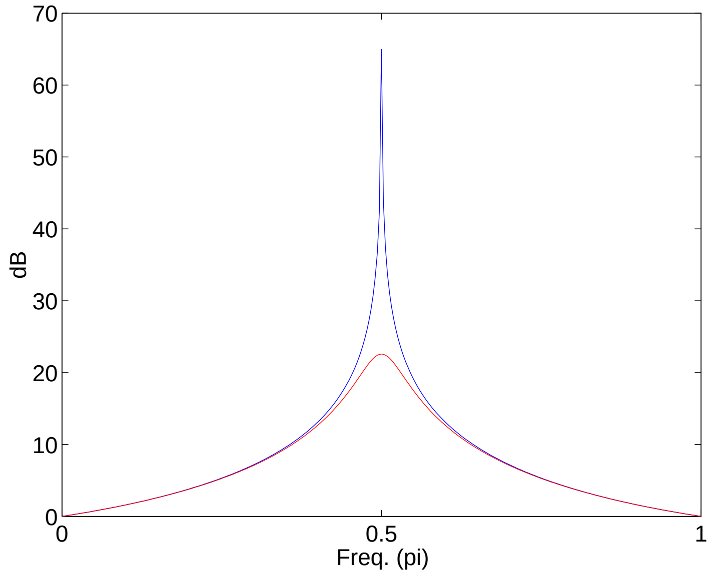

Figure 4.2: Magnitude response of two filters with poles located at $\omega_0 = \pi/2$ and $r = 0.9$ (red) or $r = 1$ (blue).

the impulse response is:

$$
\begin{aligned}
y_0 &= x_0 = 1 \\
y_1 &= x_1 + 0.5y_0 = 0 + 0.5 \times 1 = 0.5 \\
y_2 &= x_2 + 0.5y_1 = 0 + 0.5 \times 0.5 = 0.25 = 0.5^2 \\
y_3 &= x_3 + 0.5y_3 = 0 + 0.5 \times 0.25 = 0.125 = 0.5^3 \\
&\vdots \\
y_t &= 0.5^t \qquad (4\text{-}7) \\
&\vdots
\end{aligned}
$$

You can see that the output y_t continues after $t = 0$, but the value becomes smaller each time, and will decay to zero. On the other hand, if $a_1 > 1$, the output becomes bigger and bigger, and goes to infinity. This behavior is called *unstable*. From equation (4-5), we know that a_1 is a pole of the filter. In fact, a feedback filter with poles p_i is stable if and only if

$$|p_i| < 1, \text{ for all } i \qquad (4\text{-}8)$$

In other words, all the poles must be inside the unit circle. Here is the basic idea for a proof of this:

1. **A feedback filter with N feedback terms can be decomposed into N one-pole feedback filters.**

   The equation for a feedback filter with N feedback terms can be written as

$$y_t = x_t - \sum_{k=1}^{N} b_k y_{t-k} \qquad (4\text{-}9)$$

Its transfer function is

$$
\begin{aligned}
H(z) &= \frac{1}{1+\sum_{k=1}^{N} b_k z^{-k}} \\
&= \frac{z^N}{z^N+\sum_{k=1}^{N} b_k z^{N-k}}
\end{aligned}
\tag{4-10}
$$

Let $p_i, i = 1, 2, ..., N$ be its poles. Let's not worry about how we can get them (we know how if we can factor the denominator polynomial, so let's assume we have already done that). So, (4-10) can be rewritten in the form

$$
H(z) = \frac{z^N}{(z-p_1)(z-p_2)\ldots(z-p_N)} \tag{4-11}
$$

$$
= \frac{z^N}{\prod_{k=1}^{N}(z-p_k)} \tag{4-12}
$$

Recalling our introductory calculus, a fraction with a product of terms as in (4-12) can be rewritten using a partial fraction expansion as

$$
H(z) = \frac{A_1}{(1-p_1 z^{-1})} + \frac{A_2}{(1-p_2 z^{-2})} + \ldots + \frac{A_N}{(1-p_N z^{-N})} \tag{4-13}
$$

$$
= \sum_{k=1}^{N} \frac{A_k}{(1-p_k z^{-k})} \tag{4-14}
$$

Each of the terms

$$
\frac{A_k}{(1-p_k z^{-k})} \tag{4-15}
$$

is a one-pole feedback filter. So, the complete N-pole filter's output is the sum of N one-pole filters' outputs. This is called a parallel one-pole filter, and is shown for three poles in text book figure 2.1.

2. **A one-pole feedback filter is stable if and only if its impulse response is stable.**

   Consider the unit impulse δ_t,

$$
\delta_t = \begin{cases} 1 & t = 0 \\ 0 & t > 0 \end{cases} \tag{4-16}
$$

   Any discrete input signal can be thought of as a weighted sum of delayed unit impulses,

$$
\begin{aligned}
x &= \{x_0, x_1, x_2, ...\} \\
&= x_0\delta_t + x_1\delta_{t-1} + x_2\delta_{t-2} + \ldots
\end{aligned}
\tag{4-17}
$$

   We can see that the filter's response to any signal will be stable if and only if its response to each of these impulses is stable.

3. **An impulse response is stable if and only if its pole $|p| < 1$.**

   Equation (4-7) tell us that the ith one-pole feedback filter's impulse response is the signal with samples

$$1, p_i, p_i^2, p_i^3, \ldots \tag{4-18}$$

   Obviously, the impulse response is stable only when the pole $|p_i| < 1$

4. **An impulse response is neither stable nor unstable if $|p| = 1$.**

Resonance and Bandwidth

Resonance is the increase in a filter's magnitude response in the region near a pole. An example is shown in figure 4.2. The pole is at frequency $\omega = \pi/2$, which is the peak of the magnitude response. The magnitude is small when away from the pole.

One other thing I have mentioned before is that the peak becomes steep when r nears one. The steepness is measured by the filter's *bandwidth*. Bandwidth, B, is defined as the width of the filter's response at half its maximum power output. The filter's power is the square of its amplitude, so the bandwidth is also located at $1/\sqrt{2}$ of its peak amplitude value. These half power points are denoted by $H(\omega)_B^2$. If the peak value of power is $|H(\omega)|_p^2$ and the peak amplitude is $|H(\omega)|_p$, the bandwidth points are located at

$$|H(\omega)|_B = \frac{1}{2}|H(\omega)|_p^2 = \frac{1}{\sqrt{2}}|H(\omega)|_p \tag{4-19}$$

Since $20\log_{10} 1/\sqrt{2} = -3dB$

$$\frac{|H(\omega)|_B}{|H(\omega)|_p} = -3dB \tag{4-20}$$

and so the *cutoff frequencies* for a filter are sometimes called its "minus three deebee points". This definition is also shown in text book figure 3.1.

Let's take a look at the case in which a pole is on the real axis and no other poles are nearby; the pole is $a = re^{j\omega}$, $r = R$ and $\omega = 0$. A point on the unit circle is $z = e^{j\phi}$. This situation is illustrated in figure 4.3. We know that the one pole transfer function is

$$H(z) = \frac{1}{1 - a_1 z^{-1}} \tag{4-21}$$

with magnitude

$$|H(z)| = \left|\frac{1}{1 - a_1 z^{-1}}\right| \tag{4-22}$$

Consider the inverse square of the magnitude, substituting in our expressions for a_1 and z,

$$\begin{aligned}\frac{1}{|H(\omega)|^2} &= |1 - Re^{j\phi}| \\ &= (\cos\phi - R)^2 \\ &= 1 - 2R\cos\phi + R^2\end{aligned} \tag{4-23}$$

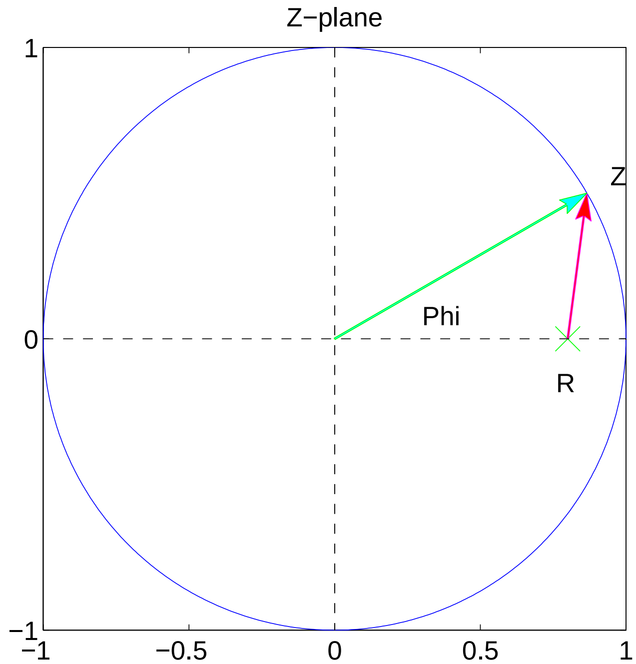

Figure 4.3: One pole $a = re^{j\omega}$ in z-plane. $r = R$, $\omega = 0$ and $z = e^{j\phi}$

The peak of $|H(\omega)|^2$ should be the value of angle ϕ that makes $1 - 2R\cos\phi + R^2$ minimum, that is when $\phi = 0$. The power at this peak is

$$|H(\omega)|_p^2 = \frac{1}{(1-R)^2} \tag{4-24}$$

According to equation (4-19),

$$B = \frac{1}{2}|H(\omega)|_p^2 = \frac{1}{2(1-R)^2} \tag{4-25}$$

The corresponding ϕ can be obtained by solving for it in the equation

$$2(1-R)^2 = 1 - 2R\cos\phi + R^2 \tag{4-26}$$

$$\cos\phi = 2 - \frac{1}{2}\left(R + \frac{1}{R}\right) \tag{4-27}$$

The filter's bandwidth is the span $[-\phi, \phi]$ — a distance of 2ϕ. When R is close to one, we can express R as a small amount ϵ less than one: $R = 1 - \epsilon$. We can then take advantage of the expansions:

$$\frac{1}{R} = \frac{1}{1-\epsilon} = 1 + \epsilon + \epsilon^2 + \epsilon^3 + \cdots \tag{4-28}$$

and

$$\cos(\epsilon) = 1 - \frac{\epsilon^2}{2!} + \frac{\epsilon^4}{4!} + \cdots \tag{4-29}$$

So, from (4-27),

$$\begin{aligned} \cos\phi &= 2 - \frac{1}{2}[(1-\epsilon) + (1+\epsilon+\epsilon^2+\epsilon^3+\cdots)] \\ &\approx 1 - \frac{\epsilon^2}{2} - O(\epsilon^3) \\ &\approx \cos\epsilon \end{aligned} \tag{4-30}$$

(where $O(\epsilon^3)$ is shorthand for terms in the expansion of order ϵ^3 or higher). Therefore, $\phi \approx \epsilon$. So, when R is close to the unit circle,

$$B = 2\phi \approx 2\epsilon = 2(1-R) \tag{4-31}$$

or

$$R \approx 1 - B/2 \tag{4-32}$$

If want a filter with a particular bandwidth, equation (4-32) gives us a way to determine its pole location.

Self-Test Exercises

1. Derive equation (4-27) from (4-26). (*Popup Answer: Start from (4-26):*

   $$\begin{aligned} 2(1-2R+R^2) &= 1 - 2R\cos\phi + R^2 \\ 1 - 4R + r^2 &= -2R\cos\phi \end{aligned}$$

   so

   $$\begin{aligned} \cos\phi &= -\frac{1}{2R}(1-4R+R^2) \\ &= 2 - \frac{1}{2}(R + \frac{1}{R}) \end{aligned}$$

)

2. Solve for R as in text book equation (3.7), but with bandwidth of 200Hz. Is it true that when R is far away from one, B grows large? (*Popup Answer:* $R = 1 - \pi(200/44100) = 0.9858$. *Yes.*)

Resons

A *reson filter* is a more general case of resonance, in which a pair of complex conjugate poles can be located at any desired frequency. If the complex poles are $Re^{\pm j\theta}$, the transfer function can be written as

$$\begin{aligned} H(z) &= \frac{1}{(1-Re^{j\theta}z^{-1})(1-Re^{-j\theta}z^{-1})} \\ &= \frac{1}{1 - Rz^{-1}(e^{j\theta}+e^{-j\theta}) + R^2z^{-2}} \\ &= \frac{1}{1-(2R\cos\theta)z^{-1} + R^2z^{-2}} \end{aligned} \tag{4-33}$$

The corresponding filter equation is

$$y_t = x_t + (2R\cos\theta)y_{t-1} - R^2 y_{t-2} \tag{4-34}$$

As you know, the closer the poles get to the unit circle, the sharper the filter's magnitude peak gets, and therefore the smaller the bandwidth is. Text book figure 4.3 shows this. The angular position of the pole determines the resonant frequency of the filter.

Self-Test Exercises

1. Derive the reson filter's transfer function (4-33) from the filter equation (4-34). (*Popup Answer: Equation (4-34) can be written as*

$$\begin{aligned} Y &= X + (2R\cos\theta)z^{-1}Y - R^2z^{-2}Y \\ Y(1 - 2R\cos\theta z^{-1} + R^2z^{-2}) &= X \\ Y &= \frac{1}{1 - 2R\cos\theta z^{-1} + R^2z^{-2}}X \end{aligned} \tag{4-35}$$

   and so becomes

$$H(z) = \frac{1}{1 - 2R\cos\theta z^{-1} + R^2z^{-2}}$$

)

Designing a Reson Filter

By now you know that if you want to design a reson filter, you need to give the resonant frequency and the bandwidth. However, the peak of a reson filter's magnitude response doesn't always occur exactly at the frequency θ of the pole — it shifts sometimes. How much it shifts, we will calculate now.

Our goal first is to find the peak frequency, or the frequency that maximizes the magnitude response. Let's see the following example.

Example 2 Find the maximum of the following function:

$$f(x) = -x^2 + 10x \tag{4-36}$$

Figure 4.4 is a plot of the function. The key to finding maxima and minima lies in the slope of the curve. At the top of a peak or the bottom of a valley, the curve has a slope of zero, in other words, the derivative equals zero. Stated algebraically, our task is to solve the equation $\mathrm{d}f(x)/\mathrm{d}x = 0$ and locate those values of x where the curve has a horizontal tangent,

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = -2x + 10 = 0 \tag{4-37}$$

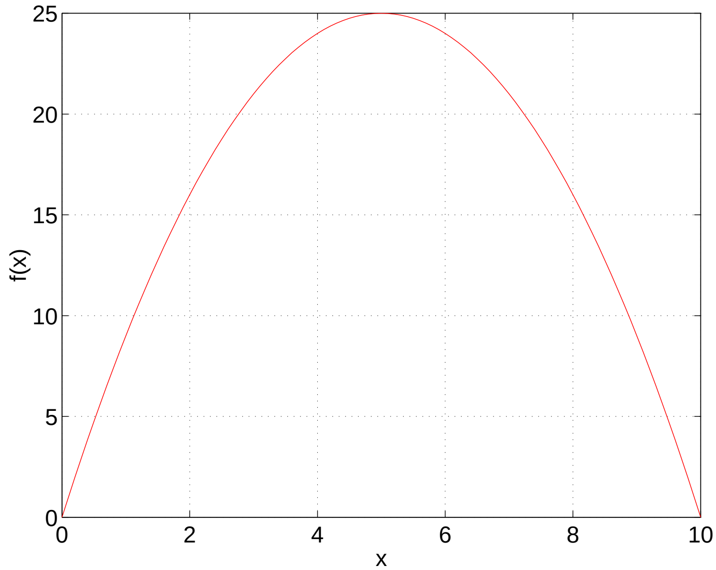

Figure 4.4: The plot of $f(x) = -x^2 + 10x$, where $x = 5$ is the extremum and $f(x)_{max} = 25$.

The solution of this equation is $x = 5$. Substituting this into the original function, the maximum is

$$f(5) = -5^2 + 10 * 5 = 25 \tag{4-38}$$

Figure 4.4 supports this conclusion.

We can use the same method to find the value of frequency where the magnitude response reaches its maximum. We already know that the transfer function of a reson is

$$\begin{aligned} H(z) &= \frac{1}{(1 - Re^{j\theta}z^{-1})(1 - Re^{-j\theta}z^{-1})} \\ &= \frac{z^2}{(z - Re^{j\theta})(z - Re^{-j\theta})} \end{aligned} \tag{4-39}$$

Substituting the phasor $z = e^{j\phi}$ into the above equation,

$$H(\phi) = efrace^{2j\phi}(e^{j\phi} - Re^{j\theta})(e^{j\phi} - Re^{-j\theta}) \tag{4-40}$$

Let's pay attention to the denominator since that is the part that produces poles. After some calculations the square of denominator becomes

$$(1 - R^2)^2 + 4R^2 \cos^2\theta - 4R(R^2 + 1)\cos\theta\cos\phi + 4R^2\cos^2\phi. \tag{4-41}$$

Let's calculate the denominator's derivative with respect to $\cos\phi$ (let's pretend that $\cos\phi$ is just the name of a variable, and treat everything else as a constant when we take the derivative), and set that equal to zero,

$$-4R(R^2 + 1)\cos\theta + 8R^2\cos\phi = 0 \tag{4-42}$$

which yields the value of $\cos\phi$ where the peak actually occurs. To be consistent with the text book, let's call the angle at the maximum ψ,

$$\cos\psi = \frac{1+R^2}{2R}\cos\theta \tag{4-43}$$

Remember we set the pole at $Re^{j\theta}$. Equation (4-43) shows the relationship between the pole angle θ and the magnitude response peak angle ψ. It is only when $R = 1$ (when the pole is on the unit circle) that the peak occurs at precisely the same angle (frequency) as the pole. Just as I did in example 2, you can find the maximum of the magnitude response as

$$\begin{aligned} |H(\psi)| &= \frac{1}{(1-R^2)\sin\theta} \\ &= \frac{1}{A_0} \end{aligned} \tag{4-44}$$

where $A_0 = (1-R^2)\sin\theta$ is called the *gain factor* and is used to normalize the magnitude response to be one at the peak. This ensures that the reson's amplification doesn't depend on its particular resonant frequency or bandwidth.

Given that we've normalized the magnitude response by the gain factor, the filter's transfer function becomes,

$$|H(z))| = \frac{A_0}{1-(2R\cos\theta)z^{-1}+R^2z^{-2}} \tag{4-45}$$

At the resonant frequency ψ,

$$|H(\psi)| = \frac{A_0}{A_0} = 1 \tag{4-46}$$

which is exactly our desired result: a bandpass filter which doesn't alter the signal's magnitude at its peak. The filter that corresponds to this transfer function is

$$y_t = A_0x_t + (2R\cos\theta)y_{t-1} - R^2y_{t-2} \tag{4-47}$$

So, here is the procedure to design a reson:

1. Decide on the desired bandwidth B and resonant frequency ψ.

2. Calculate the pole radius R from the bandwidth B

$$R \approx 1 - \frac{B}{2} \tag{4-48}$$

3. Calculate the cosine of the pole angle θ (thankfully, we don't need to calculate θ itself, as the cosine is what appears in the filter equation)

$$\cos\theta = \frac{2R}{1+R^2}\cos\psi \tag{4-49}$$

4. Calculate the gain factor A_0

$$A_0 = (1-R^2)\sin\theta \tag{4-50}$$

5. The resulting filter equation is

$$y_t = A_0x_t + (2R\cos\theta)y_{t-1} - R^2y_{t-2} \tag{4-51}$$

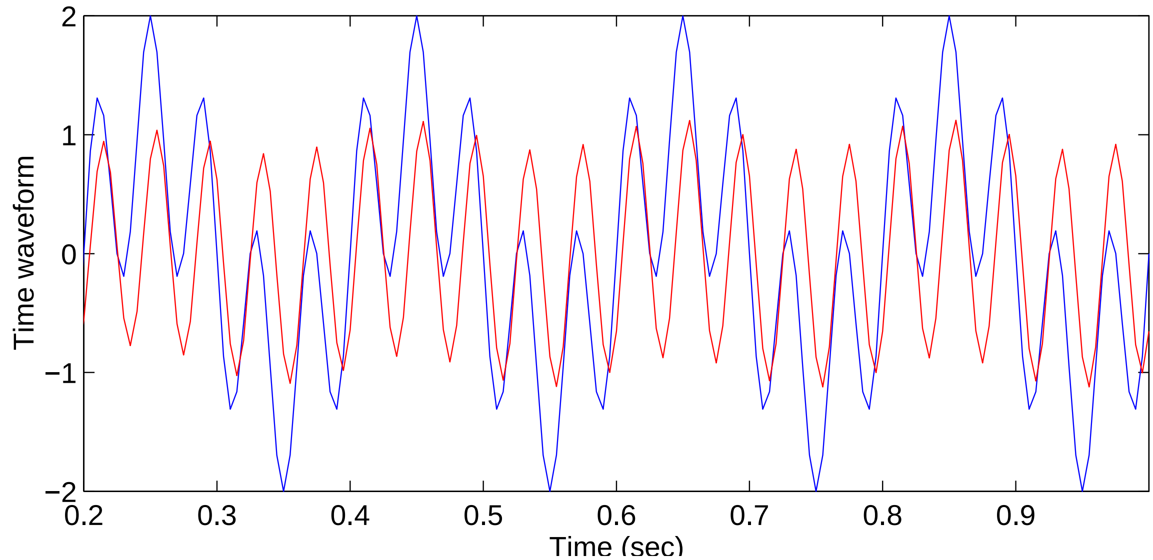

Figure 4.5: Input signal (blue) and output of the reson filter of figure 4.6 (red).

Example 3 Digital resonators are useful in many applications, including simple bandpass filtering and speech generation. The angle of the pole determines the resonant or pass band frequency. The bandwidth is related to the pole's radius. Using the above procedure, we will now make a reson filter and use it to do simple signal filtering.

The blue line in figure 4.5 is a simple signal that I'd like to filter. It is the sum of two sine waves,

$$x = \sin(2\pi t f_1) + \sin(2\pi t f_2) \tag{4-52}$$

where $f_1 = 5$Hz and $f_2 = 25$Hz. Let's design a reson filter to remove the 5Hz component. The signal was sampled at $F_s = 200$Hz, and we convert frequencies in Hz to radians/sample using $\theta = 2\pi f / F_s$. We want to retain the component at 25Hz, so the corresponding normalized frequency will be $\psi = (2\pi)(25)/200 = \pi/4$. Let's make the bandwidth $B = 0.1$ radians/sample which is 3.1831Hz. The pole radius can be calculated as in the procedure, and is $R = 0.95$; the angle of the pole is therefore 0.2504π instead of $\pi/4$. Why is the pole at this angle? (*Popup answser: see text book equation (5.7); note that* $R \neq 1$).

Figure 4.6 shows this filter's magnitude and phase response. From the magnitude response, you can see that it acts as a simple bandpass filter. The pair of complex poles $p_0 = 0.95e^{0.2504j\pi}$ and its conjugate mate $p_0^* = 0.95e^{-0.2504j\pi}$ are shown in the z-plane in figure 4.7.

The filtered signal is shown as the red curve in figure 4.5. The output signal is obviously different from the input. Let's check its frequency components so we will have a better idea of how they have been changed. This is shown in figure 4.8. The 25Hz component is passed and the 5Hz one is heavily suppressed, but not completely. If we decrease the bandwidth (make R closer to the unit circle), the 5Hz component will be suppressed further.

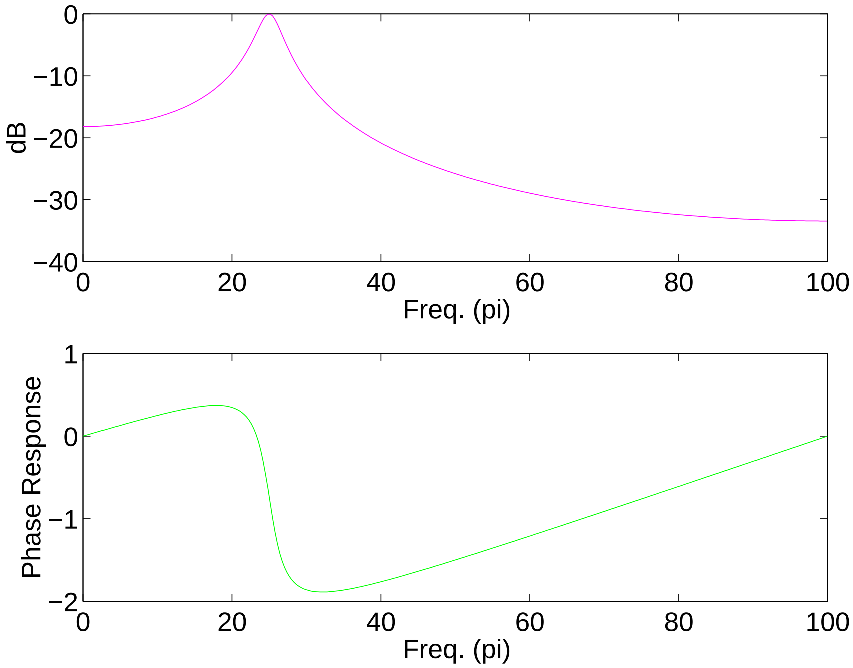

Figure 4.6: Reson filter's magnitude (top) and phase (bottom) response. The poles are at $p_0 = 0.95e^{0.2504j\pi}$ and its conjugate mate, $p_0^* = 0.95e^{-0.2504j\pi}$.

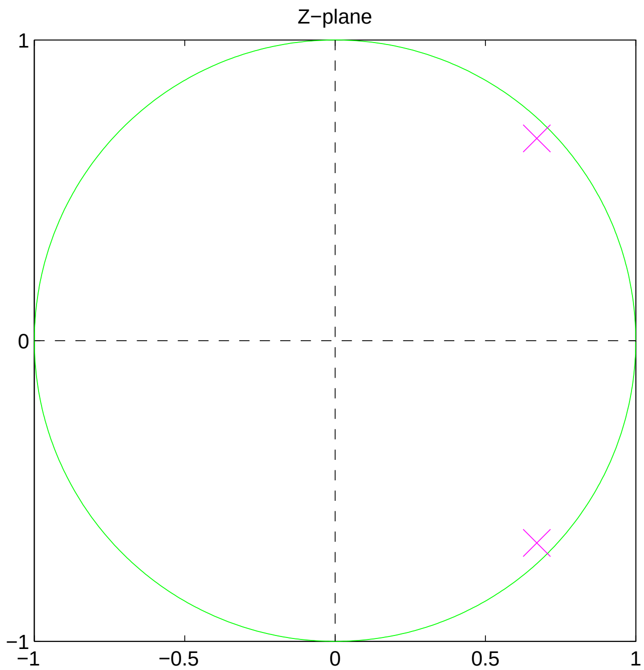

Figure 4.7: Locations of the poles in the z-plane for the filter shown in Figure 4.6. The poles are at $0.95e^{\pm 0.2504j\pi}$.

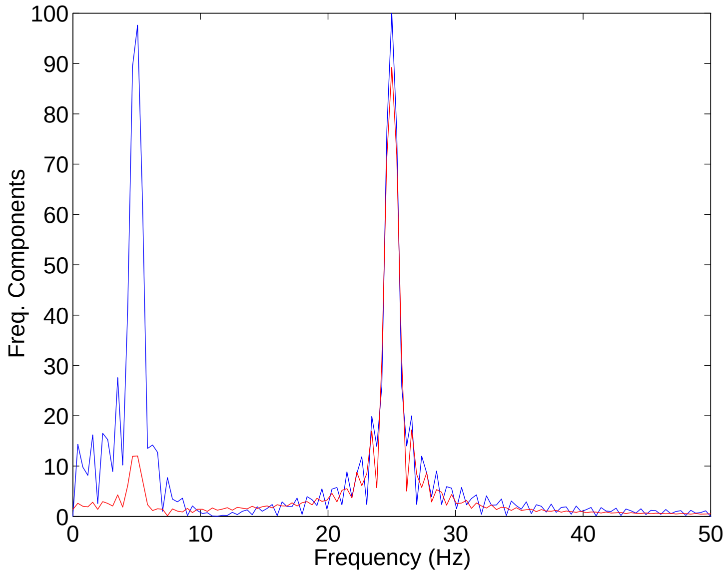

Figure 4.8: Spectra corresponding to the signals in figure 4.5. Original is blue; filtered is red. The first frequency component at 5Hz is suppressed and the second one at 25Hz is passed.

Self-Test Exercise

1. Derive equation (5.4) in the text book (the expression for the maximum of the magnitude response). (*Popup Answer: From equation (5.2),*

$$\frac{1}{|H(\phi)|^2} = (1 - R^2)^2 + 4R^2 \cos^2 \theta - 4R(R^2 + 1) \cos \theta \cos \phi + 4R^2 \cos^2 \phi$$

Substitute (5.3) into above equation

$$\begin{aligned}
\frac{1}{|H(\psi)|^2} &= (1 - R^2)^2 + 4R^2 \cos^2 \theta - 4R(R^2 + 1) \cos \theta \cos \phi + 4R^2 \cos^2 \phi \\
&= (1 - R^2)^2 + 4R^2 \cos^2 \theta - 4R(R^2 + 1) \cos \theta (\frac{1 + R^2)}{2R}) \cos \theta + 4R^2 (\frac{1 + R^2)}{2R})^2 \cos^2 \theta \\
&= (1 - R^2)^2 + 4R^2 \cos^2 \theta - 2(1 + R^2)^2 \cos^2 \theta + (1 + R^2)^2 \cos^2 \theta \\
&= (1 - R^2)^2 + 4R^2 \cos^2 \theta - (1 + R^2)^2 \cos^2 \theta \\
&= (1 - R^2)^2 + \cos^2 \theta [4R^2 - 1 - 2R^2 - R^4] \\
&= (1 - R^2)^2 - (1 - R^2)^2 \cos^2 \theta \\
&= (1 - R^2)^2 (1 - \cos^2 \theta) \\
&= (1 - R^2)^2 \sin^2 \theta
\end{aligned}$$

)

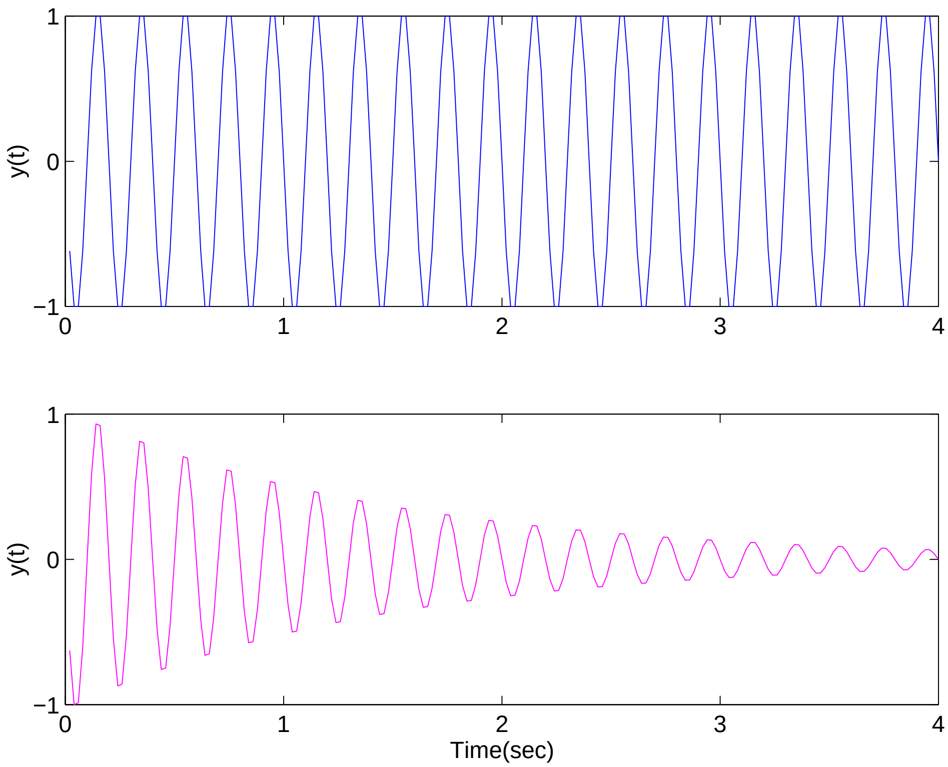

Figure 4.9: Impulse response of a reson filter (4-56). $R = 1$ is on top and $R = 0.5$ is in the bottom plot.

Resonators Everywhere

Examples of resonators can be found in many places. One example is the damped sine wave, which is merely the impulse response of reson,

$$y_t = \delta_t + (2R\cos\theta)y_{t-1} - R^2 y_{t-2} \tag{4-54}$$

where δ_t is the unit impulse input,

$$\delta_t = \begin{cases} 1 & t = 0 \\ 0 & t > 0 \end{cases} \tag{4-55}$$

Ignoring the derivation procedure (which you can see in the text book), the impulse response y_t can be expressed as

$$y_t = R^t \frac{\sin[\theta(t+1)]}{\sin\theta} \tag{4-56}$$

When $R = 1$, this is a sinusoid generator (see figure4.9, top) and when $R < 1$, it is a damped sine wave (see figure4.9, bottom).

Mixing Feedback and Feedforward Filters

We now seen feedforward filters with zeros in the transfer function and feedback filters with poles. Zeros suppress frequency components and poles enhance them. Quite often, we want to combine poles and zeros to improve the filter's features, such as the flatness of the passband and the abruptness with which its response transitions between the passband and the stop band. One example of this combination is putting a pair of zeros ($z = \pm 1$) in a reson filter,

$$\mathcal{H}(z) = \frac{1 - z^{-2}}{1 - 2R\cos\theta z^{-1} + R^2 z^{-2}} \tag{4-57}$$

This will improve version of reson. See text book figure 7.2 (page 95). Generally the transfer function of filter can written as

$$\mathcal{H}(z) = \frac{a_0 + a_1 z^{-1} + \cdots + a_{m-1} z^{-(m-1)}}{1 + b_1 z_{-1} + \cdots + b_n z^{-n}} \tag{4-58}$$

the filter is

$$y_t = a_0 x_t + a_1 x_{t-1} + \cdots + a_{m-1} x_{t-(m-1)} - b_1 y_{t-1} - \cdots - b_n y_{t-n} \tag{4-59}$$

Depending on the coefficients b_i and a_i ($i = 1, 2, 3...$), the filter will show different features. Some of these features have proven so useful that these forms of (4-58) have acquired special names, such as elliptic, Butterworth, etc (usually, based on the form of the numerator and/or denominator polynomial).

Implementation

Just as in lesson 3, feedback filters have their delays implemented with queues which are “circular arrays”. However, there are two additional complications that we must deal with: complex poles and accuracy of numerical computation.

Avoiding Complex Numbers

In principle, there is no reason that we couldn't perform all our computations using complex numbers and just output the real part of the result as the filtered signal. However, we often want to perform filtering in real time, and so would like to avoid any unnecessary computation. A cute trick allows us to eliminate the need for complex numbers.

Remember that all complex poles will appear in conjugate pairs. Since each pole is a root of a polynomial, that means that the denominator of the filter's transfer function will have even degree, and that when it is factored the conjugate pairs z_i will appear as $(z - z_i)(z - z_i^*)$. If we multiply this out, we get $z^2 - 2\,\mathrm{Re}[z_i]z + |z_i|^2$. In other words, the imaginary parts of poles are gone. Since we're already using a_i for the feedforward coefficients and b_i for the feedback ones in the filter equation, let's set $c_i = 2\,\mathrm{Re}[z_i]$ and $d_i = |z_i|^2$. If the transfer function's denominator is of order N (N even), then we can multiply out the polynomials for each complex conjugate pair and rewrite the denominator as

$$(z^2 + c_0 z + d_0)(z^2 + c_1 z + d1) \cdots (z^2 + c_{N/2-1} z + d_{N/2-1}) \tag{4-60}$$

The numerator might simply be $A(z+1)^N$. In that case, if we multiply numerator and denominator by z^{-N}, we obtain equation (7.3) in text book chapter 13.

Limitations of Numerical Accuracy

When we talk the mathematics of filters, we assume that numbers have infinite accuracy. Unfortunately, this is not the case for computer implementation. These days, computers (including digital signal processors) typically use either 32 or 64 bits to represent numbers (be they fixed or floating point). That may seem like a great deal of precision, but it is unfortunately not uncommon for intermediate results in numerical computation to require many more bits to retain needed accuracy. There are a number of ways this can happen, but one is where a computation is performed iteratively, so that a long chain of operations is applied to inputs before they become outputs. At each step of this chain, the result has limited precision. In other words, the number we get is not exactly correct — in general, it can't be, since we don't have infinite precision.

This loss of precision can mount rapidly, eventually destroying the result. We can think of this limited precision as an *error*. If the input is from a 16-bit A/D, for example, and we assume it does its conversion absolutely accurately to the smallest bit (which, as you have seen in lesson 2 and will revisit in lesson 8, is probably not realistic), then that's around $\log_{10} 2^{16} \approx 5$ decimal digits, to be generous. Each arithmetic operation we perform, regardless of the number of bits we use, can have the undesirable effect of decreasing the number of digits of precision we have. Sums of approximate values have errors which are the sums of their addends'. Multiplication tends to magnify errors. Let's see this with a simple example.

Example 4 Consider the following two recurrence relations for computing the series $\{x_n\} = \{1/3^n\}$ ($n = 0, 1, 2, \ldots$):

$$x'_n = \frac{1}{3} x'_{n-1} \tag{4-61}$$

$$x''_n = \frac{10}{3} x''_{n-1} - x''_{n-2} \tag{4-62}$$

Both of these equations are mathematically “correct”. They also look like the expressions we use for our filters. Yet, they yield very different results because of loss of significance. Let’s use an initial value of $x'_0 = 0.99996$ for (4-61) and initial values of $x''_0 = 1$ and $x''_1 = 0.33332$ for (4-62). This is an initial error of 0.00004 for x'_0 and $0.00001\bar{3}$ for x''_1. I’ll use MATLAB to compute the first ten terms in each sequence with double-precision accuracy for each calculation. The MATLAB code is

```
% significance.m
% 2/22/02 MDS
% Examine how error grows in an iterative computation
stdout = 1;

% This isn’t efficient, but it is straightforward

% Values of n for computation
n = [0:10];

% Series 1: the real McCoy
x = 1./3.^n;

% Series 2: x’_n = 1/3 x’_{n-1}
xp = 0.99996;

for i = [1:10],
  xp(i+1) = 1/3 * xp(i);  % Remember, MATLAB indices start at 1, not 0
end;

% Series 3: x’’_n = 10/3 x’’_{n-1} - x’’_{n-2}
xpp = [1 0.33332];

for i = [2:10],
  xpp(i+1) = 10/3 * xpp(i) - xpp(i-1);
end;

% Print out the results
fprintf(stdout, ’ n          x_n              x’’_n            x’’’’_n\n’);
for i = [0:10],
  fprintf(stdout, ’%2.1d\t%12.10f\t%12.10f\t%12.10f\n’, ...
          i, x(i+1), xp(i+1), xpp(i+1));
end;
```

The output this script produces is:

```
 n          x_n              x’_n             x’’_n
 0      1.0000000000    0.9999600000    1.0000000000
 1      0.3333333333    0.3333200000    0.3333200000
 2      0.1111111111    0.1111066667    0.1110666667
```

3	0.0370370370	0.0370355556	0.0369022222
4	0.0123456790	0.0123451852	0.0119407407
5	0.0041152263	0.0041150617	0.0029002469
6	0.0013717421	0.0013716872	-0.0022732510
7	0.0004572474	0.0004572291	-0.0104777503
8	0.0001524158	0.0001524097	-0.0326525834
9	0.0000508053	0.0000508032	-0.0983641945
10	0.0000169351	0.0000169344	-0.2952280648

The first column is accurate to the full precision of IEEE floating point (that is, the computation is that accurate; the output was limited to ten decimal places). The second column's error is *stable*: it decreases in an exponential manner, and the value for x'_{10} is within 0.0000000007 of the actual value. The third column's error is *unstable*: it increases exponentially.

Couple this with the need, as described in the text book in chapter 13, section 8, for very precise pole placement, and I think you can see that we need to be careful about our implementations. We saw in this lesson's section on combining feedforward and feedback filters the general transfer function for a filter (4-58) and its direct implementation (4-59), which involved $m-1$ delayed values of x and n delayed values of y.

However, it is better to implement such a digital filter as a cascade of second-order ones, by factoring the numerator and denominator (which should be "easy," since we know the locations of the poles and zeros) and multiplying out terms with complex conjugate pairs as described in the previous section. The resultant fraction will have numerator and denominator which are each a product of second-order polynomials. Each ratio of second-order polynomials is a subfilter, which can be implemented by a simple update equation. If we take the output of one subfilter and present it as the input of the next, the output of the last subfilter will be equivalent to the output of the entire original filter (we saw this cascade of filters in lesson 3).

Assignment 4

1. Text book chapter 5 problem 2.

2. Text book chapter 5 problem 7 (a) and (b).

3. Given the input signal
$$x = \sin(2\pi f_1 t) + \sin(2\pi f_2 t) \tag{4-63}$$
with $f_1 = 5Hz$ and $f_2 = 25$. Design a reson filter to filter out the frequency component f_2. Use MATLAB to implement your filter and plot its magnitude response, poles in the z-plane, and input and output signals. Submit your code with your results.