Real Rational Filters, Zeros and Poles

Jack Xin (Lecture) and J. Ernie Esser (Lab) *

Abstract
Classnotes on real rational filters, causality, stability, frequency response, impulse
response, zero/pole analysis, minimum phase and all pass filters.

1 Real Rational Filter and Frequency Response

A real rational filter takes the form of difference equations in the time domain:

\[ a(1)y(n) = b(1)x(n) + b(2)x(n - 1) + \cdots + b(nb + 1)x(n - nb) \]
\[ -a(2)y(n - 1) - \cdots - a(na + 1)y(n - na), \]

where \( na \) and \( nb \) are nonnegative integers (number of delays), \( x \) is input signal, \( y \) is filtered output signal. In the z-domain, the z-transforms \( X \) and \( Y \) are related by:

\[ Y(z) = H(z) X(z), \]

where \( X \) and \( Y \) are z-transforms of \( x \) and \( y \) respectively, and \( H \) is called the transfer function:

\[ H(z) = \frac{b(1) + b(2)z^{-1} + \cdots + b(nb + 1)z^{-nb}}{a(1) + a(2)z^{-1} + \cdots + a(na + 1)z^{-na}}. \]

The filter is called real rational because \( H \) is a rational function of \( z \) with real coefficients in numerator and denominator.

The frequency response of the filter is \( H(e^{j\theta}) \), where \( j = \sqrt{-1}, \theta \in [0, \pi] \). The \( \theta \in [\pi, 2\pi] \) portion does not contain more information due to complex conjugacy. In Matlab, the frequency response is obtained by:

\[ [h, \theta] = \text{freqz}(b, a, N), \]

where \( b = [b(1), b(2), \cdots, b(nb + 1)] \), \( a = [a(1), a(2), \cdots, a(na + 1)] \), \( N \) refers to number of sampled points for \( \theta \) on the upper unit semi-circle; \( h \) is a complex vector with components \( H(e^{j\theta}) \) at sampled points of \( \theta \). The phase response is the angle of \( H(e^{j\theta}) \).

*Department of Mathematics, UCI, Irvine, CA 92617.
1.1 Example

Example 1: consider:

\[ H(z) = \frac{1 + 0.2 z^{-1}}{(1 - 0.867 z^{-1})(1 - (0.067 + j0.867) z^{-1})(1 - (0.067 - j0.867) z^{-1})}, \]  

(4)

then verify by direct multiplication or applying Matlab command (compute polynomial coefficients from roots)

\[
\text{poly}([0.867, 0.067 + j \times 0.867, 0.067 - j \times 0.867]) = [1.0000, -1.0010, 0.8724, -0.6556]
\]

that:

\[ H(z) = \frac{1 + 0.2 z^{-1}}{1 - z^{-1} + 0.827 z^{-2} - 0.6556 z^{-3}}. \]  

(5)

Compute frequency response:

\[ [h, w] = \text{freqz}([10.2], [1 - 10.827 - 0.6556]), \]

then plot:

\[
\text{plot}(w, \text{abs}(h)),
\]

to get amplitude response in Fig. 1, and

\[
\text{plot}(w, \text{angle}(h))
\]

to get phase response in Fig. 2.

Let us explain Fig. 1 by a zero-pole analysis. Notice that filter \( H \) has a real zero at \( z = -0.2 \), and 2 complex poles at \((0.067 \pm j0.867)\). We plot zero and poles by:

\[ \text{zer} = -0.2; \text{pol} = [0.867(0.067 + j \times 0.867)(0.067 - j \times 0.867)]; \text{zplane(zer, pol)}. \]

See Fig. 3.

The two peaks in Fig. 1 are due to the two poles 0.867 and 0.067 +0.867 j. The first pole being real (phase angle =0) determines the location of the first peak at \( \theta = 0 \). Similarly, the phase angle of the second pole

\[ \text{atan}(0.867/0.067) = 1.4937, \]

sets the location of the second peak. The zero -0.2 (at angle \( \pi \)) causes the attenuation at high frequency (the dipping of the curve in Fig. 1 near 3).

In summary,
Figure 1: Illustration of amplitude frequency response of the rational filter in example 1.

- Real poles near $z = 1 (-1)$ lead to gain at low (high) frequencies.
- Complex poles near unit circle lead to gain at frequencies near the poles’ angles.
- Real zeros near $z = 1 (-1)$ lead to loss (attenuation) in low (high) frequencies.
- Complex zeros near unit circle lead to loss (attenuation) at frequencies near the poles’ angles.

### 1.2 Impulse Response

Impulse is $\delta(n) = 1$ at $n = 1$, $\delta(n) = 0$, if $n \geq 2$. In Matlab, it is implemented as $imp = [1, \text{zeros}(1, N)]$, $N = 64$ or a larger number as needed. Impulse response of a rational filter is:

$$h = \text{filter}(b, a, imp);$$

an example is:

$$b = 1, a = [1, -0.9].$$

you’ll see an exponentially decaying curve, Fig. 1.2.
The impulse response can be obtained by solving difference equation:

\[ y(n) - 0.9 y(n - 1) = \delta(n), \quad n = 1, 2, \cdots. \]  

(6)

Let \( n = 1 \), then \( y(-1) = 0 \) by causality, \( y(1) = \delta(1) = 1 \). Letting \( n = 2, 3, \cdots \), we have:

\[
\begin{align*}
  y(2) - 0.9 y(1) &= 0, \\
  y(3) - 0.9 y(2) &= 0, \\
  y(n) - 0.9 y(n - 1) &= 0,
\end{align*}
\]

(7)

so \( y(2) = 0.9, \ y(3) = 0.9 y(2) = 0.9^2, \ y(n) = 0.9^{n-1} \) which is the exact impulse response, and agrees with Fig. 4. Because the impulse response consists of infinitely many nonzero terms for all values of positive integer \( n \), it is called Infinite Impulse Response (IIR).

Example 2:

\[ y(n) = b_1 x(n) + b_2 x(n - 1) + b_3 x(n - 2), \]

Substituting \( x(n) = \delta(n), \ n = 1, 2, \cdots \), we have:

\[
\begin{align*}
  y(1) &= b_1, \ y(2) = b_2, \ y(3) = b_3, \ y(4) = 0, \ y(n) = 0, \ n \geq 4.
\end{align*}
\]
Impulse response is:

\[ [b_1, b_2, b_3, 0, \cdots, 0], \]

essentially 3 dimensional. Such a filter is called *Finite Impulse Response* (FIR).

IIR and FIR filters are related. By geometric series:

\[
\frac{1}{1 - 0.9 z^{-1}} = 1 + 0.9 z^{-1} + 0.9^2 z^{-2} + \cdots,
\]

whose finite truncation is a FIR. Reversely, an IIR filter can be viewed as a limit of a sequence of FIR filters.

The impulse response of example 2 does not change sign, which is due to the pole \( z = 0.9 \) being real. Impulse response will change sign if an complex pole appears.

**Example 3:**

\[
H(z) = \frac{1 - 0.5 z^{-1}}{1 - z^{-1} + 0.5 z^{-2}},
\]

\[ h = \text{filter}([1, -0.5], [1 - 10.5], [1, \text{zeros}(1,31)]); \text{plot}(h), \]

gives the curve in Fig. 5. To show the poles, do \( \text{roots}([1, -1, 0.5]) \) to get \( (1 \pm j)/2 \).
1.3 Group and Phase Delays

If $H(z) = z^{-L}$, $L \geq 1$, then its frequency response is:

$$H(e^{j\theta}) = e^{-j\theta L},$$

whose phase is:

$$\phi = -\theta L,$$

and so:

$$L = -\frac{d\phi}{d\theta},$$

or:

$$L = -\frac{\phi}{\theta}.$$  

If a filter consists of multiple delays as in example 2, then we measure the overall delay by group delay (denoted by $\tau_g$):

$$\tau_g(\theta) = -\frac{d\phi(\theta)}{d\theta},$$

computed in Matlab as:

$$[gd, w] = \text{grpdelay}(b, a, N).$$

The phase delay is:

$$\tau_p(\theta) = -\frac{\phi(\theta)}{\theta}.$$
2 Minimum Phase and All Pass Filters

Consider real, causal, stable and rational filters. Here stable means that the poles are inside unit circle. Let $\beta$ be a zero of $H(z)$, then

$$H(z) = (1 - \beta z^{-1}) H_0(z).$$

Let us move the zero as follows:

$$\beta \rightarrow \overline{\beta}^{-1},$$

where bar is complex conjugate. The $\overline{\beta}^{-1}$ is called conjugate inverse of $\beta$, and it has the same phase angle as that of $\beta$.

Then the modified filter has transfer function:

$$H_1(z) = \overline{\beta} [1 - \overline{\beta}^{-1} z^{-1}] H_0(z),$$

with frequency response ($z = e^{j\theta}$):

$$(\overline{\beta} - e^{-j\theta}) H_0(e^{j\theta}).$$

By the identify:

$$\overline{\beta} - e^{-j\theta} = -e^{-j\theta} (1 - \overline{\beta} e^{j\theta}) = -e^{-j\theta} (1 - \beta e^{-j\theta}),$$

we see that

$$|H_1(e^{j\theta})| = |H(e^{j\theta})|.$$ 

So replacing a zero by its conjugate inverse does not change the magnitude of frequency response. The phase is another story:
Theorem 2.1. If $|\beta| < 1$, the group delay of $H_1(z)$ is large than that of $H(z)$ at all angular frequency $\theta \in [0, \pi]$. In other words, moving a zero from outside (inside) of unit circle to its conjugate inside (outside) the unit circle reduces (increases) group delay.

Proof: Let $\beta = \beta_r + j\beta_i$, $\zeta = \arctan(\beta_i/\beta_r)$. The phase of $(1 - \beta z^{-1})$ is:

$$
\phi(\theta) = \text{angle}(1 - (\beta_r + j\beta_i)(\cos \theta - j\sin \theta))
= \arctan\left(\frac{\beta_r \sin \theta - \beta_i \cos \theta}{1 - \beta_r \cos \theta - \beta_i \sin \theta}\right)
= \arctan\left(\frac{\sin(\theta - \zeta)}{|\beta|^{-1} - \cos(\theta - \zeta)}\right)
$$

Direct calculation shows that:

$$
\tau_g(\theta) = -\frac{d\phi}{d\theta} = \frac{|\beta| - \cos(\theta - \zeta)}{|\beta| + |\beta|^{-1} - 2 \cos(\theta - \zeta)}.
$$

Replacing $\beta$ by its conjugate inverse leaves $\zeta$ unchanged, also $|\beta| + |\beta|^{-1}$ is unchanged. So if $|\beta| < 1$, then $|\beta|^{-1} > 1$, $\tau_g$ increases when $\beta \to \beta^{-1}$.

A filter with minimum group delay among all filters having the same amplitude frequency response is called minimum phase filter. A real causal stable and rational (RCSR) filter is minimum phase if and only if all its zeros are inside the unit circle.

If $H(z)$ is not minimum phase, suppose it has a factor $(1 - \beta z^{-1})$, $|\beta| > 1$, then the factor can be written as:

$$
1 - \beta z^{-1} = (1 - \bar{\zeta}^{-1} z^{-1}) \frac{1 - \beta z^{-1}}{1 - \bar{\zeta}^{-1} z^{-1}},
$$

Let $a = \bar{\beta}^{-1}$, the ratio in (9) is put in the form:

$$
\frac{\beta^{-1} z^{-1}}{1 - a z^{-1}} = \beta \frac{\bar{a} - z^{-1}}{1 - a z^{-1}}.
$$

We have

Proposition 2.1. Filter $H_a(z) = \frac{\bar{a} - z^{-1}}{1 - a z^{-1}}$, $|a| < 1$, is a stable all pass IIR filter.

By all pass, we mean that $H_a(e^{j\theta}) = \text{constant}$ for all $\theta$. In fact,

$$
H_a(e^{j\theta}) = \frac{\bar{a} - e^{-j\theta}}{1 - a e^{-j\theta}} = -e^{-j\theta} \frac{1 - \bar{a} e^{j\theta}}{1 - a e^{-j\theta}},
$$

clearly, $|H_a(e^{j\theta})| = 1$. 8
A general all pass filter is:

\[ \prod_{k=1}^{p} \frac{a_k - z^{-1}}{1 - a_k z^{-1}}. \]

Finally, by flipping zeros from outside to their conjugate inverses inside of unit circle to reduce group delay, we have the decomposition:

\[ H(z) = H_{\text{min-phase}} \cdot H_{\text{all-pass}}(z). \]

### 3 Butterworth and Chebychev Filters

Classical Butterworth filter is a low pass filter with monotone decreasing amplitude of frequency response and maximum gain at zero frequency. Suppose we wish to remove noise above 30 Hz from a signal sampled at 100 Hz. Normalized frequency \( \theta = \text{frequency divided by half the sampling frequency} \). The cut-off relative frequency is \( 30/50 \). To generate Butterworth filter coefficients, enter:

\[
[b, a] = \text{butter}(5, 30/50);
\]

then:

\[
[h, w] = \text{freqz}(b, a, 128); \text{plot}(w/pi, \text{abs}(h)).
\] (10)

To compare with order 10 Butterworth filter, enter the above lines again with 10 in place of 5. Fig. 6 is a plot with both in there. We see that order 10 filter has a sharper transition from passband to stopband.

The definition of the Butterworth filter as a real rational filter may be a little beyond the scope of these notes. However, if we don’t mind our filter being non-causal, then we can more simply define a Butterworth filter directly in the frequency domain. Let \( \xi \) be the frequency variable and define a cutoff frequency \( \xi_0 \). Then we can define a low pass Butterworth filter by

\[ B(\xi) = \frac{1}{1 + (\frac{\xi}{\xi_0})^{2n}}. \]

Here \( n \) denotes the order of the filter. As the order \( n \) increases, \( B(\xi) \) should look more and more like the ideal low pass filter

\[ L_{\text{ideal}}(\xi) = \begin{cases} 
1 & \text{if } |\xi| \leq \xi_0 \\
0 & \text{if } |\xi| > \xi_0 
\end{cases}. \]

In a discrete setting, consider applying the frequency domain defined version of the Butterworth low pass filter to a signal of length \( N \). In MATLAB notation, letting \( k = 1:N \) index the frequency and defining \( v_0 \) as the cutoff frequency, we can define
\[ B = \text{fftshift}(1./(1 + (2*(k-N/2)/N/v0).^(2*n))). \]

We can then apply \( B \) to a signal \( s \) using the fast Fourier transform according to

\[ s_{\text{filtered}} = \text{real}(\text{ifft}(B.*\text{fft}(s))); \]

Note that this definition of the Butterworth filter differs from MATLAB’s \texttt{butter} function, which computes coefficients for a causal Butterworth filter. However, since that function involves finding roots of a polynomial, it can be unstable for high order filters.

![Amplitude response of order 5 (blue) and 10 (red) Butterworth filter](image)

Figure 6: Illustration of amplitude frequency response of order 5 (blue) and 10 (red) low pass Butterworth filter with cut-off relative frequency at 0.6.

Chebychev filter of type I has a ripple (oscillatory) in the passband. Enter:

\[
[b, a] = \text{cheby}1(4, 1, 0.6);
\]

then (10). Here 4 is the filter order, 1 is 1 decibel on the amplitude of ripples (amp in decibel = 20 \( \log_{10} \) (amp)), amp is short for amplitude). Repeat this for order 10, the combined plot is Fig. 7.

High pass, band pass and stopband Butterworth and Chebyshev filters are left in Matlab projects of next section.

### 4 Matlab Projects

\textit{Exercise 1:} take \( a, b \) as in example 3. Compute group and phase delay, and plot them as a function of \( \theta \).
Figure 7: Illustration of amplitude frequency response of order 5 (blue) and 10 (red) low pass Chebyshev type I filter with cut-off relative frequency at 0.6.

\[
\begin{align*}
\text{gd, } w &= \text{grpdelay}(b, a, 128); \text{plot}(x, gd). \\
\text{h, } w &= \text{freqz}(b, a, 128), \\
pd &= -\text{angle}(h)/w; \\
\text{plot}(w, pd).
\end{align*}
\]

Exercise 2: consider Fibonacci sequence as a difference equation:

\[
y_{n+1} = y_n + y_{n-1},
\]

with initial data \(y(1) = 1, y(2) = 1\). Define:

\[
\begin{align*}
a &= [1, -1, -1]; \\
b &= [1, \text{zeros}(1,9)]; \\
x &= b; \\
y &= \text{filter}(b, a, x),
\end{align*}
\]

you’ll see the first 10 Fibonacci numbers.

Exercise 3: High Pass Butterworth and Chebyshev Filters

\[
\begin{align*}
[b, a] &= \text{butter}(4, 0.8,'high'); \\
[b, a] &= \text{cheby}1(4, 1, 0.8,'high');
\end{align*}
\]

plot their amplitude frequency responses and compare.
Exercise 4: Band Pass Butterworth and Chebyshev Filters

\[ [b, a] = \text{butter}(4, [0.4, 0.7]); \]
\[ [b, a] = \text{cheby}1(4, 1, 0.8, [0.4, 0.7]); \]
plot their amplitude frequency responses and compare.

Exercise 5: Stop Band Butterworth and Chebyshev Filters

\[ [b, a] = \text{butter}(4, [0.4, 0.7], 'stop'); \]
\[ [b, a] = \text{cheby}1(4, 1, [0.4, 0.7], 'stop'); \]
plot their amplitude frequency responses and compare.

Exercise 6: Low Pass Butterworth Filter for Images

Although we mostly have been considering one dimensional signals, the frequency domain definition of the Butterworth filter extends naturally to two dimensional images. The same formula applies, but with frequency index \( k \) replaced by the distance to the origin in the two dimensional centered frequency domain \([2]\). First define a \( 100 \times 100 \) test image \( z \) by

\[ z = \text{zeros}(100); z(46:55,46:55) = 100; \]

Then define the distance to the origin by

\[ [J,I] = \text{meshgrid}(-49:50,-49:50); \]
\[ D = \text{sqrt}(I.^2+J.^2); \]

Let \( D_0 \) be the cutoff frequency and let \( n \) be the order of the filter. Then we can define a low pass 2D Butterworth filter by

\[ F = \text{fftshift}(1./(1+(D/D_0).^(2*n))); \]

We can display \( F \) as an image by typing \texttt{imagesc}(F); \texttt{colormap(gray)}; Apply \( F \) to the test image \( z \) using the two dimensional fast Fourier transform, which can be done by typing

\[ Z = \text{real}(\text{ifft2}(F.*\text{fft2}(z))); \]

We can view the resulting image by again using \texttt{imagesc}(Z); Experiment with different values of \( D_0 \) and \( n \), and in particular note the ringing effect when \( n \) is large.
References

