10. Practical Aspects of Filtering

Filters are commonly applied to data to eliminate unwanted frequencies and emphasize those of interest. A good example is our application of the boxcar and Gaussian smoothing filters to the smooth temperature data in an earlier exercise. An ideal filter would have an amplitude spectrum that is zero outside the pass band and unity within it. The problem with sharp filters is that their implementation requires long time sequences (remember that the uncertainty principle tells us that a sharp feature in one domain is broad in the other) and truncating these sequences leads to a Gibb’s phenomenon in the filter amplitude response. Sharp causal filters also apply large strongly frequency-dependent phase shifts to the data. All filters involve a tradeoff between the sharpness of the filter and the length of the sequences required to implement them.

**Butterworth Filter**

There are many different filters but the Butterworth filter is a popular filter with a simple amplitude response and for many seismologists it is the only filter they will ever use. The amplitude spectrum of the analog Butterworth low-pass filter is defined by

\[
|G_{\text{lowpass}}(f)|^2 = \frac{1}{1 + \left(\frac{|f|}{f_c}\right)^{2n}}
\]  

(10-1)

where \(f_c\) is the cutoff frequency and \(n\) is the order of the filter. At the cutoff frequency the response is \(1/2\) the value at 0 frequency. As \(n\) increases the sharpness of the cutoff increases (Figure 1) but this is done at the expense of the output tending to be ‘ringy’ as you will see in Exercise 5.

The amplitude spectrum of a high pass filter is given by

\[
|G_{\text{highpass}}(f)| = 1 - |G_{\text{lowpass}}(f)|
\]  

(10-2)

A pass band filter is created by

\[
|G_{\text{passband}}(f)|^2 = \frac{1}{1 + \left(\frac{|f| - f_b}{f_c}\right)^{2n}}
\]  

(10-3)

where \(f_b\) is the center of the pass band and \(f_c\) is the pass band half width. The notch or band stop filter is simply

\[
|G_{\text{bandstop}}(f)| = 1 - |G_{\text{bandpass}}(f)|
\]  

(10-4)

**Filters in the Frequency Domain**

To a filter in the frequency domain, one simply applies a Fourier transform to the signal of interest, multiplies by the frequency response of the filter, and then performs an inverse Fourier transform.
For the Butterworth filters described above, one can see that it would be necessary to specify the phase of the filter as well as the amplitude response. One simple option is to assume that the phase is zero. This works well but it leads to a filter that is acausal or non-realizable (in the natural world) – that is the output of the filter will have an output that depends on the input at later times.

Butterworth filters are technically analog filters but can be approximated very closely with a digital filter.

The amplitude response of filters are generally displayed on a plot with logarithmic scales (Figure 2). Amplitude differences are expressed in decibels. For the amplitude response $G$ the difference in decibels between two amplitude values is given by

$$dB = 20 \log_{10} \left| \frac{G_1}{G_2} \right|$$

or if one prefers to think in terms of the power spectrum

$$dB = 10 \log_{10} \left( \frac{|G_1|^2}{|G_2|^2} \right)$$

The filter roll-off can be specified in decibels per octave where one octave corresponds to a doubling in frequency. The ripple measures how much the amplitude response varies in the passband and stopband.

**Acausal Filters in the Time Domain**

We can apply create an acausal digital finite impulse response (FIR) by extending the filter to negative coefficients

$$y_t = \sum_{k=-P}^{N-1} g_k x_{t-k}$$

where the filter $g_k$ is non zeros for $-P < 0 < N$ (with $P$ and $N$ integers).

Another very convenient way to make any time domain filter zero-phase is to apply it once, then reverse the output time series and apply the filter a second time, before reversing the output back again. The phase lags from the two applications will be off opposite sign and thus cancel each other out.

Mathematically we can write.

$$y(t) = g(-t) *[g(t) * x(t)]$$

Reversing a time series is equivalent to taking the complex conjugate in the frequency domain. Thus, in the frequency domain we get

$$Y(f) = G^*(f) G(f) X(f) = |G(f)|^2 X(f)$$
A Simple Feedback Filter

Consider a simple filter in which the output is delayed by one sample and added back into the input after scaling by $-a_1$. This can be represented schematically by

$$x_t \rightarrow [L] \rightarrow a_i \xrightarrow{z^{-1}} y_t$$

(10-9)

A pure feedback filter (i.e., the non-feedback portion of $L'$ just passes through what enters) can be written

$$x_t \rightarrow [x_t - a_1 y_{t-1}] \rightarrow y_t$$

(10-10)

or

$$y_t = x_t - a_1 y_{t-1}$$

(10-11)

We can write this in the Z-transform domain (remembering that a unit time delay is obtained by multiplying by $z$) as

$$Y(z) = X(z) - a_1 z Y(z)$$

(10-12)

Solving for $Y(z)$ gives

$$Y(z) = \frac{X(z)}{1 + a_1 z}$$

(10-13)

Since the transfer function, which we will term $G(z)$, is just output over input we get

$$G(z) = \frac{\text{output}}{\text{input}} = \frac{Y(z)}{X(z)} = \frac{1}{1 + a_1 z}$$

(10-14)

This is the first example of a filter in which the transfer function is not a simple polynomial.

If we let the input be an impulse function

$$x_t = \delta_t = (1, 0, 0, 0, ...)$$

(10-15)

then output obtained from equation (7) is

$$y_t = (1, -a_1, a_1^2, -a_1^3, ...)$$

(10-16)

The output for an impulse continues on indefinitely (hence the name infinite impulse response filter). A stable filter is one for which the energy is bounded. For $|a_1| < 1$, the total energy is

$$E = \sum_{k=0}^{\infty} \left((-a_1)^k\right)^2 = \frac{1}{1 - a_1^2}$$

(10-17)

For $|a_1| \geq 1$, $E$ is unbounded and the filter is unstable.

Another way to express the stability is to consider the transfer function of the feedback filter $(1+a_1 z)^{-1}$. This is just the reciprocal of a two-element finite impulse response filter $(1+a_1 z)$. The condition for stability, $|a_1| < 1$ is the requirement that $(1+a_1 z)$ is minimum phase.

Feedback or Infinite Impulse Response (IIR) Filters

In many time-domain filtering applications finite impulse response filters are inefficient because long filters are required to implement many specific responses. Often it is more practical to use feedback filters (also known as recursion filters or infinite impulse response (IIR) filters) which are defined by
The first part of the filter is just a finite impulse response filter but the second part used the previous M outputs of the filter. Unlike a finite impulse response filter a feedback filter will generally have an infinitely long response to an impulse – hence the alternative name ‘infinite impulse response’.

It can be shown that in the Z-transform domain the feedback filter response can be written

\[ Y(z) = G(z)X(z) \tag{10-19} \]

where

\[ G(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1z + b_2z^2 + ... + b_Nz^N}{a_0 + a_1z + a_2z^2 + ... + a_Mz^M} = \frac{k_B(z - \beta_1)(z - \beta_2)...(z - \beta_N)}{k_A(z - \alpha_1)(z - \alpha_2)...(z - \alpha_M)} \tag{10-20} \]

The terms \( \beta_i, i = 1, 2, ... N \) are termed the zeros of the filter and the terms \( \alpha_j, j = 1, 2, ... N \) are termed the poles of the filter. Quite commonly the response of a geophysical measurement system (e.g., a seismometer) is described in terms of poles and zeros which are themselves obtained by factoring the Z-transform polynomials. Equation (10-20) can be converted to the frequency response of the filter simply by substituting

\[ z = \exp\left(-\frac{2\pi ik}{N}\right), \quad k = 0, 1, ..., N - 1 \tag{10-21} \]

into equation. If you are comfortable with the complex plane you can learn a lot about the filter by plotting the poles and zeros since – the filter will have a small amplitude response where zeros plot near the unit circle and a high amplitude response where the poles plot near it. I will not go into this in any more detail other than to note that if you need to know more about poles and zeros and their application in seismology there is a good book by Scherbaum, F., *Of Poles and Zeros: Fundamentals of Digital Seismology*, 2nd edition, 2001.

I should also caution that many people define the Z-transform alternately as

\[ Z\left[ b_j \right] = B(z) = b_0z^0 + b_1z^{-1} + b_2z^{-2} + ... + b_{N-1}z^{-(N-1)} = \sum_{j=0}^{N-1} b_kz^{-j} \tag{10-22} \]

instead of

\[ Z\left[ b_j \right] = B(z) = b_0z^0 + b_1z^1 + b_2z^2 + ... + b_{N-1}z^{(N-1)} = \sum_{j=0}^{N-1} b_kz^j \tag{10-23} \]

so when interpreting a filter response you need to be clear as to which definition is being used.

### Applying Butterworth Filters in Matlab

In the Exercise 5 you will use Matlab to explore the use of an Infinite Impulse Response approximation to the analog Butterworth filter.