Chapter 7 – Digital Filters

In this chapter

Filter Design methods

Improving the design

Transition from analog to digital Filters

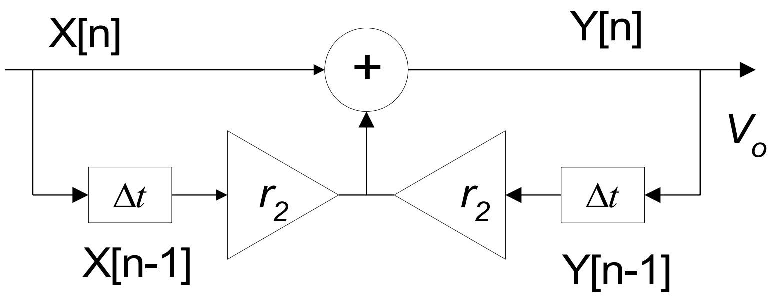
Introduction

Energy storage elements produce frequency response that acts as filters and we have seen in the analog filters how different combination of resistors, capacitors and op-amps exhibit the filter action. The same effect can be achieved using computers in the form of digital filters, but there is much more then just mimicking the analog response. Some of the techniques in the digital filters have no parallel in the analog design, such as the sampled data processing of averaging and windowing that is only possible in digital domain. Although the digital filters are diversified in nature, from the control system point of view, replacing the analog filters with the counter part digital filters is an important consideration in any system design.

A filter is generated when a delayed input and or the output is fed back to the system after being multiplied by a complex quantity (see Figure 7.1). The complex quantity multiplier forms the Transfer Function of the system and how to formulize it is the topic of the discussion in this chapter.

The analog filters of the previous chapter produced the frequency response through implementing a Transfer Function, whose coefficient values matched the component values. But the digital filters will be implemented as a solution of the difference equation (obtained from the corresponding Transfer Function), whose output will be a discrete time data. The coefficients would still be obtained from the corresponding frequency response Transfer Function.

In this chapter, we will also develop the tools to analyze the frequency and phase response and be able to analyze the time sequence generated by the filter output. The tools are based on a free plotting software (xmgarce) and the perl Tk module. (The listing and the installation procedure are provided in Appendix A.) In the first part, we will devote our attention to the digital filters that are derived from the analog designs and require feedback from the previous output, later we will discuss the sample data processing that does not require any feedback from the previous output.



******Insert Figure 7.1 here*******

Figure 7.1. The Block diagram of digital filters, with feedback and feed forward mechanism, where r_1 and r_2 are the complex number multipliers and x[n-1] and y[n-1] are the previous inputs and outputs.

Filter Design methods

Filters are identified by the frequency response they produce. The low-pass, high-pass, band-pass, and band-stop are the basic types of response a filter is characterized with. With each frequency response, there is the corresponding Transfer Function describing the input and output relationship. The analog domain Transfer Function is the Fourier Transform of the continuous time impulse response, whereas the digital domain Transfer Function is the z-Transform of the impulse response. The Fourier Transform is a function of continuous time frequency variable ω_t and the z-Transform is a function of discrete time frequency variable $\omega_{k\Delta t}$. The analog filters generated the output as if there was a continuous time convolution; on the other hand, the digital filter's output is from the discrete time convolution, as you will see later in the section.

We would like to show you in the next derivative that the filter frequency response due to a given input (essentially the Fourier Transform of the impulse response times the input function) is the same as convolution integral in the time domain. Once we establish the fact that the output of a system can be obtained by simply convolving the input with its impulse response then the digital signal processing may be implemented as an iterative algorithm of additions of input multiplied with the impulse response.

$y(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$ The convolution integral 7.1

The discrete time version is given as,

$$y_n = \sum_{k=-\infty}^{\infty} x[n-k] \times h[k] \qquad 7.2$$

For a sinusoidal input the output is given as,

$$y(t) = \int_{-\infty}^{\infty} h(\tau)e^{s(t-\tau)}d\tau$$

$$y(t) = e^{st}\int_{-\infty}^{\infty} h(\tau)e^{-s\tau}d\tau$$

$y(t) = H(s)e^{st}$ The Transformed output 7.3

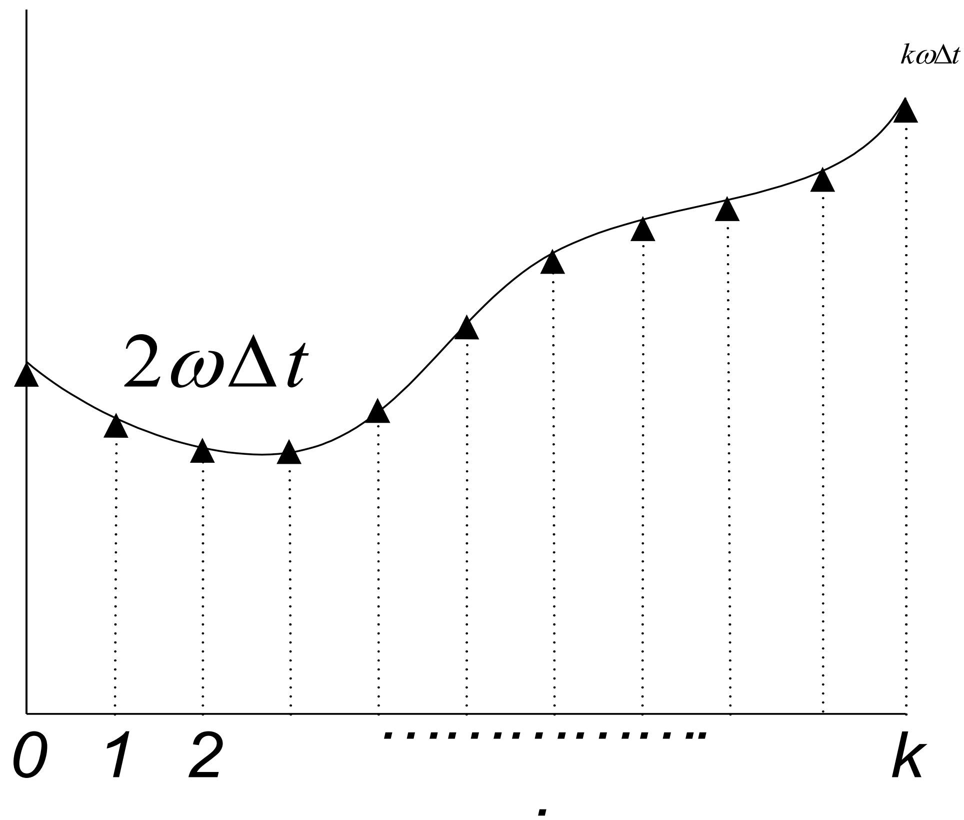
The Equation 7.3 being derived from the Equation 7.1 and 7.2, indicates the convolution of the impulse response with the input signals is equivalent of producing the frequency response of the Transformed output. It should also be mentioned that the same result could be achieved when we change the order of convolution, the Equation 7.3 could also be written as,

$$y[n] = \sum_{k=-\infty}^{\infty} h[n-k] \times x[k] \qquad 7.4$$

There is a subtle difference between the Equation 7.2 and 7.4. The Convolution of the impulse response with the input signals (Equation 7.4) requires a complete set of input data before the convolution can begin, whereas, the Equation 7.2 produces the output as soon as we have the first input signal available and is the preferred method for digital signal processing applications, but we must know the impulse response beforehand. Another important point is that the impulse response is usually a finite length sequence so there is no need to sum all the terms to infinity. Before we proceed further with the discussion of Transfer Function, the concept of sampling frequency must be clarified.

The frequency variable $\omega\Delta t$

There is a certain amount of time delta between successive scanning of input signals by digital computers. The input data taken go through the processing of multiplying with the complex number Transfer Function, before the next data are scanned, and this processing delay is Δt. The value of the continuous time input function is valid only at the time $k\omega\Delta t$, where *k* is the sample number, (see the Figure 7.2). Thus, we have a frequency variable that varies with the sample number instead of sample time.



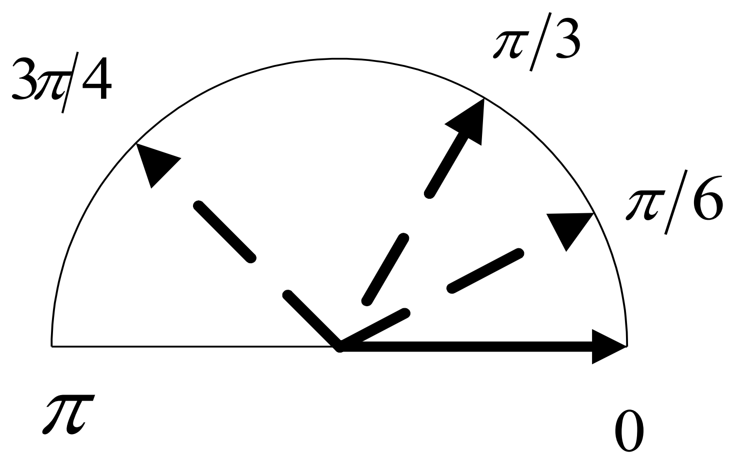
******Insert Figure 7.2 here*******
Figure 7.2. The continuous time and discrete time sampling correspond at $t = k\Delta t$

The term $\omega\Delta t$ will be used quite often in the digital filter design so let's reserve the capital Ω specifically for indicating the sampling frequency variable,

$$\omega\Delta t = \Omega$$
$$z^1 = e^{j\omega\Delta t} = e^{j\Omega} = \cos\Omega + j\sin\Omega$$
$$z^2 = e^{j\omega 2t} = e^{j2\Omega} = \cos 2\Omega + j\sin 2\Omega$$
$$z^{-1} = e^{-j\omega\Delta t} = e^{-jk\Omega} = \cos\Omega - j\sin\Omega$$

The sampling frequency variable Ω should be seen in the context of sampling rate. For example, the sampling rate is 1000 samples per second. The Nyquist frequency is 500 cycles per second or $2\pi 500$ radians per second. So if you are designing a filter that would suppress all the frequencies beyond 100 Hz ($2\pi 100$ radians/sec), then the corresponding cutoff point in terms of pole frequency is $0.2\Omega radians/sample$. Another way of looking at the sampling frequency is the time delay Δt. Suppose you have a data acquisition system that scans input values every 1 msec. The Nyquist frequency requires minimum of 2 samples, thus the highest frequency the system can process is 500 Hz, since $f = 1/_{2\Delta t} = 1/_{.002} = 500Hz$

The computers only see frequencies up to the Nyquist Frequency and that is different from the analog domain frequency where the range is unlimited. In digital domain, the Nyquist frequency is the highest frequency that we can pass through a digital filter. (We start seeing the alias effect after that.) The normalized sampling frequency function $z = e^{j\Omega}$ is a vector, rotating counter clockwise from $\Omega = 0$ to the maximum of π radians per sample, reaching the Nyquist frequency. If z was the only vector in our Transfer Function, then there is no gain and no loss, as there is no change in the output. The Figure 7.3 depicts the magnitude of the vector at various points on the semi circle.



******Insert Figure 7.3 here*******
Figure 7.3. The Transfer function composed of a single vector z, rotating counter clockwise. Notice the magnitude remains same through out the sampling frequency range of $\Omega = 0$ to $\Omega = \pi$.

The Transfer Function of a filter is a function of the frequency variable Ω and the two aspects of the function that we are interested in are **the frequency and phase response** and the coefficients that form the **difference equation**. You will soon realize that the Equation 7.4 is what needs to be implemented as an iterative algorithm in order to realize a digital filter. But to see the broad picture of how a filter will react at different frequencies, the frequency and phase response must be produced. In either case, it is imperative that we obtain the Transfer Function that corresponds to the desired frequency response.

Transfer Function revisited

Think of the Transfer Function as a complex quantity whose value depends on the frequency *s* of the input signal. As we know, in a linear and time invariant system, an input signal of the form (Ae^{-st}) when multiplied with a complex number $H(s)$ produces a new value of the same frequency *s*, but a different magnitude B and phase θ and that is what a filter outcome is, the magnitude and phase of the input signal may be altered but the frequency remains the same. A low pass filter shortens the magnitude of higher frequencies, whereas, the high-pass filter shortens the magnitude of the lower frequencies. Since, formulating a Transfer Function is central to filter design, let's see how a digital Transfer Function is formed.

z-Plane poles and zeros

Suppose, we have a feedback system where the output is fed back to the system after a delayed time and multiplied by a complex constant $re^{j\theta}$. For $\theta = 0^\circ$ the constant multiplier $re^{j\theta}$ is a real positive value $(re^{j0} = r)$, and a real negative value if $\theta = 180^\circ$ $(re^{j180} = -r)$.

$$y_n = x_n + ry_{n-1}$$

The delayed output is expressed as,

$$y_{n-1} = re^{j(k-1)\omega\Delta t} = re^{jk\omega\Delta t}e^{-j\omega\Delta t}$$

In the familiar Transformed notation,

$$x_n \to X \quad y_n \to Y \qquad ry_{n-1} \to rYz^{-1}$$

The input and output relationship may be expressed as,

$$Y(1 - rz^{-1}) = X$$

$$H(\omega\Delta t) = \frac{Y}{X} = \frac{1}{(1 - rz^{-1})} \qquad 7.5$$

The Equation 7.5 simply states that the pole of the system's Transfer Function multiplies the system's output and the delayed output is fed back to the system.

A different situation may arise in a feed forward system such as,

$$y_n = x_n - rx_{n-1}$$

The delayed input may be expressed as,

$x_{n-1} = re^{j(k-1)\omega\Delta t} = re^{jk\omega\Delta t}e^{-j\omega\Delta t}$

In the familiar Transformed notation,

$x_n \to X \qquad y_n \to Y \qquad x_{n-1} \to rXz^{-1}$

The input and output relationship may be expressed as,

$$Y = X(1 - rz^{-1})$$

$$H(\omega\Delta t) = \frac{Y}{X} = (1 - rz^{-1}) \qquad 7.6$$

The Equation 7.6 states that the zero of the system's Transfer Function multiplies the system's input and a delayed input is fed back to the system.

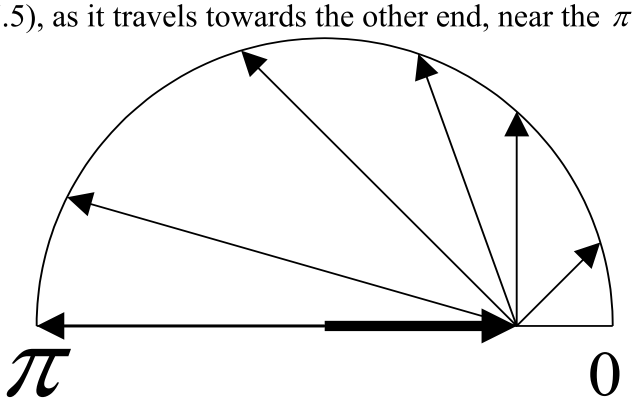
Let's analyze the Equations 7.5 and 7.6 by redefining them as polynomials, so we could see where the roots are,

$$H(z) = (1 - rz^{-1}) = \frac{z - r}{z} \qquad 7.7$$

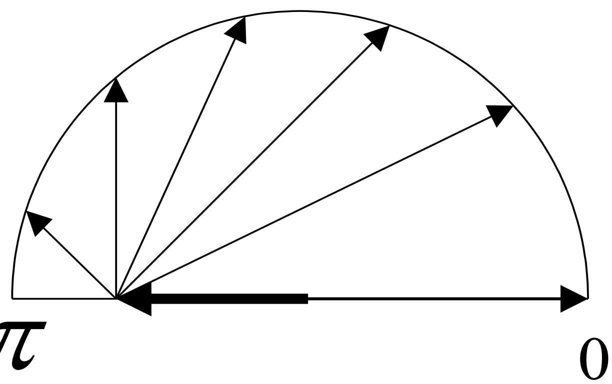
$$H(z) = \frac{1}{(1 - rz^{-1})} = \frac{z}{z - r} \qquad 7.8$$

The Equation 7.7 has a zero in the numerator at *r* and a pole in the denominator at 0, whereas the Equation 7.8 has a pole in the numerator at *r* and a zero in the denominator at 0.

The distance vector $(z - r)$ is a function of frequency variable Ω, whose length increases, (if $\theta = 0^\circ$, see thin arrows in Figure 7.4) or decreases (if $\theta = 180^\circ$, see Figure 7.5), as it travels towards the other end, near the π radians per sample.



****Insert Figure 7.4 here*****
Figure 7.4. The magnitude of the vector $(z-r)$, ($re^{j0}=r$) as it goes through different sampling frequency. The Notice the increase in magnitude as the vector $(z-r)$ as it rotates from $\Omega=0$ to $\Omega=\pi$.



****Insert Figure 7.5 here*****
Figure 7.5. The magnitude of the vector $(z-r)$, ($re^{j180}=-r$) as it goes through different sampling frequency. The Notice the decrease in magnitude as the vector $(z-r)$ as it rotates from $\Omega=0$ to $\Omega=\pi$.

The simple poles and zeros of Equation 7.7 and 7.8 are only the building blocks, a complex system may have several poles and zeros and the overall system response is a linear product of the individual response.

How do the Equation 7.7 and 7.8 act as filters of frequencies, see them in the context of vectors with magnitude as a function of frequency.

The low-pass Digital Filter Transfer Function

Imagine the length of the vector $(z-r)$ at different points of sampling frequencies as it moves along from 0 to π radians per sample. If r is the pole vector then its contribution to the magnitude is inversely proportional to its distance from the circle perimeter and if r is the zero vectors, then its contribution is directly proportional to its distance from the same perimeter. The response has a peak at the shortest distance from the edge of the circle.

Let's consider the effect of pole vectors on the magnitude of the Transfer Function, (refer to the Figure 7.4). The constant multiplier $re^{j\theta}$ is a real value vector when $\theta = 0^{\circ}$. The vector r lies on the right side of the semi circle and the quantity $(z - r)$ becomes a function of frequency variable Ω, whose length increases as the sampling frequency Ω increases. The magnitude has its highest value near the Nyquist frequency of π radians per sample. Having an inverse effect on the magnitude (being a pole vector), the higher the frequencies the lower the magnitude. The net effect is the suppression of the higher frequencies from the input signals compare to lower frequencies, creating a low-pass filter.

The magnitude of the vector $(z - r)$ is obtained by multiplying with its complex conjugate as shown below.

$$|H(\Omega)| = \left|\frac{1}{z-r}\right| = \left|\frac{1}{\sqrt{((\cos\Omega - r) + j\sin\Omega) \times ((\cos\Omega - r) - j\sin\Omega)}}\right| \qquad 7.9$$

$$|H(\Omega)| = \left|\frac{1}{1 - rz^{-1}}\right| = \frac{1}{\sqrt{(\cos\Omega + r)^2 - (j\sin\Omega)^2}} \qquad 7.10$$

Simplifying the Equation 7.10 using Trigonometric identity $\cos^2\theta + \sin^2\theta = 1$

$$|H(\Omega)| = \left|\frac{1}{1 - rz^{-1}}\right| = \frac{1}{\sqrt{(1 - 2r\cos\Omega + r^2}}$$

The Phase response as a function of frequency is given as,

$$\Phi = -\arctan\frac{r\sin\Omega}{r\cos\Omega + 1}$$

Example 7.1:

A Data Acquisition System has a sampling rate of 1024 samples per second. Formulate a single pole Transfer Function that would suppress all frequencies beyond 100 Hz.

Solution:

The Nyquist frequency:
$1024/2 = 512Hz$

Need a low-pass filter. The cutoff point as a fraction of the Nyquist frequency is

$$^{100}\!/_{500} = 0.2\Omega$$

Solving the Equation 7.9 for *r* when the magnitude is 0.707,

$$|\ (\Omega)| = \frac{1}{\sqrt{(1+\ ^{2}) - 2\ \ \cos(0.2\pi)}} = 0.707$$

$$r = 0.48$$

The Transfer Function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1-0.48z^{-1}} \qquad 7.13$$

The single pole Transfer Function of the Equation 7.13 must be converted into a difference equation for the filter to be realized and that you will see later in the section.

Gain:

It is desirable to restrict the output to a maximum gain of unity at the peak value. A Gain factor may be added to bring down the high magnitude of pole vector. Considering the previous example, a unity Gain at 0 Hz may be computed as follows,

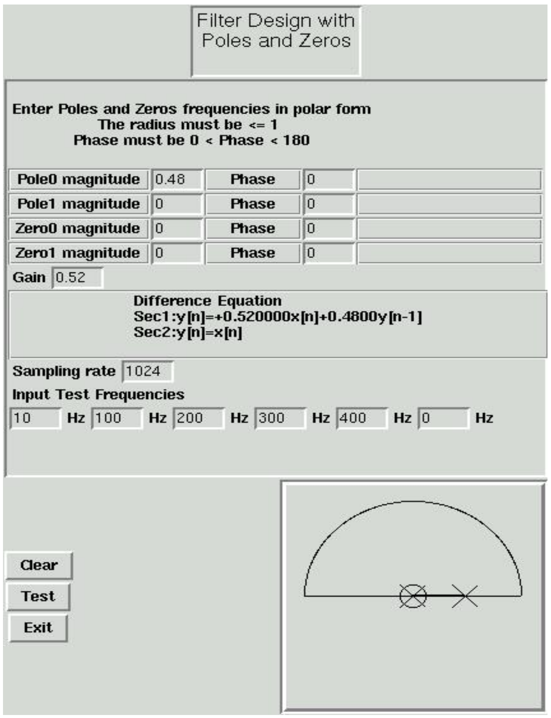
The maximum Gain at 0 Hz

$$G = \frac{z}{z-0.48} = \frac{e^{j0}}{e^{j0}-0.48} = 1.92$$

To bring the Gain down to 1 at 0 Hz, the Transfer Function must be multiplied by a Gain value of 1/1.92 or 0.52

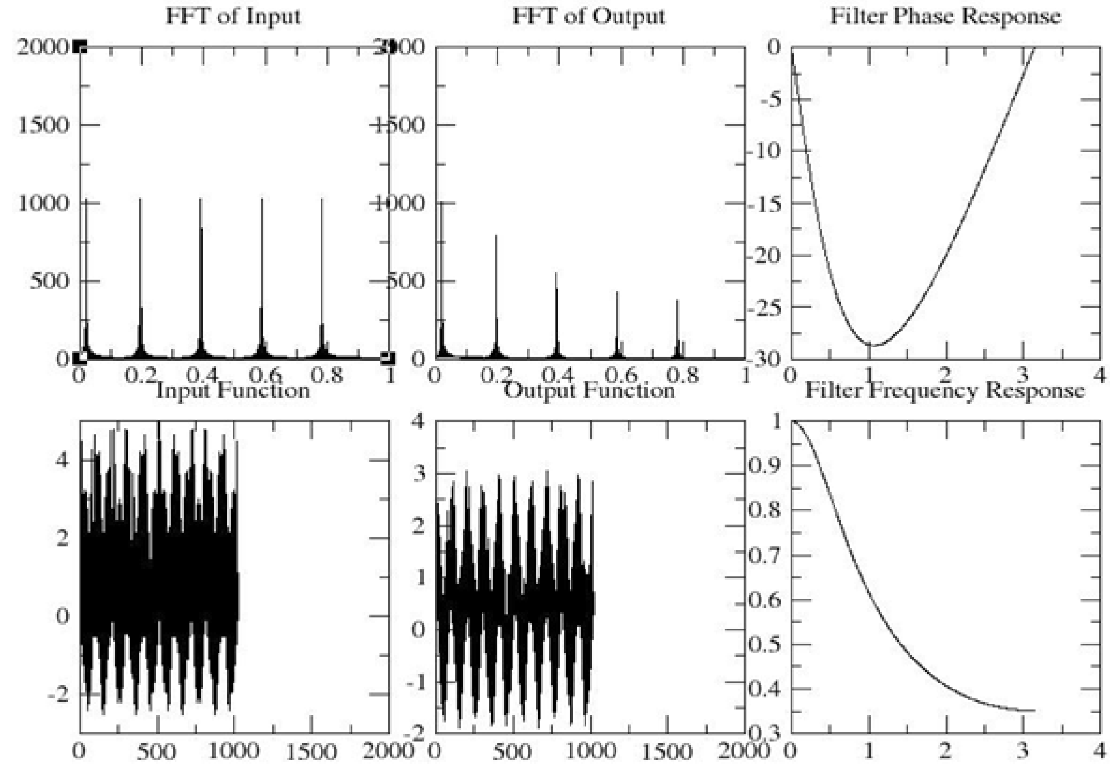
Testing the Filter Design

The real test of the filter design is to actually put it through a sweep frequency and view its outcome through a plot of the frequency response and make sure that the performance is acceptable. An application program “filterTest.pl” is presented for this purpose (the listing is provided in Appendix A). The program has a graphical user interface as shown in Figure 7.6, where the user can enter up to two poles and two zero vectors in polar coordinates and enter up to six different input test frequencies to generate test input function. For the Example 7.1 you may enter 0.48 for pole0 magnitude and select 10 Hz, 100 Hz, 200 Hz, 300 Hz and 400 Hz as input frequencies. Pressing “Test” will display six different plots, a) Input Function in time (obtained by combining all the input frequencies), b) plot of "FFT of Input", c) Filter response in time (plot section "Output Function"), d) plot of the "FFT of Output", e) the Frequency Response and f) The Phase Response. The filter action is clearly displayed by the drop in the magnitude of the frequencies as can be seen from the "FFT of Output" plot.



******Insert Figure 7.6 here*******

Figure 7.6. The Graphical User Interface of the application program "filterTest.pl".



****Insert Figure 7.7 here*****

Figure 7.7. The output generated by the application program “filterTest.pl” for the parameters of the Example 7.1. Six different plots are displayed, including; a) Input Function, b) “FFT of Input”, c) Output Function, d) “FFT of Output”, e) Frequency Response and f) The Phase Response.

It should be mentioned here that the magnitude *r* should always be less then 1 for a pole vector. A value greater then 1 will make the system unstable as it means you are feeding more then the input and that will make the system grow out of bound eventually.

The High-pass Digital Filter Transfer Function

If the pole lies on the left side of the semi circle ($\theta = 180^\circ$), then its distance would increase as we go further towards the lower end of 0 radians per sample, (see the Figure 7.8), since, having an inverse effect, the Transfer Function magnitude would be lower at the lower frequencies, the net effect is suppression of the lower frequencies from the input (a high-pass filter) signals, whereas the high frequencies are amplified.

The pole magnitude is,

$$|H(\Omega)| = \left|\frac{1}{1+rz^{-1}}\right| = \frac{1}{\sqrt{(1+r^2)+2r\cos\Omega}}$$

Separating the real and imaginary part,

$$|H(\Omega)| = \left|\frac{1}{1+rz^{-1}}\right| = \frac{1}{\sqrt{(r\cos\Omega \pm 1)^2 - (jr\sin\Omega)^2}} \qquad 7.14$$

The Phase response obtained from the real and imaginary portion of the Equation 7.14 as a function of frequency,

$$\Phi = \arctan\frac{r\sin\Omega}{r\cos\Omega + 1}$$

Example 7.2:

Going back to the previous example, we can change the low frequency filter to a high frequency by simply changing the phase angle of the vector from 0 to 180 degree, this would create the magnitude,

$$r = 0.48e^{-j180} = -0.48$$

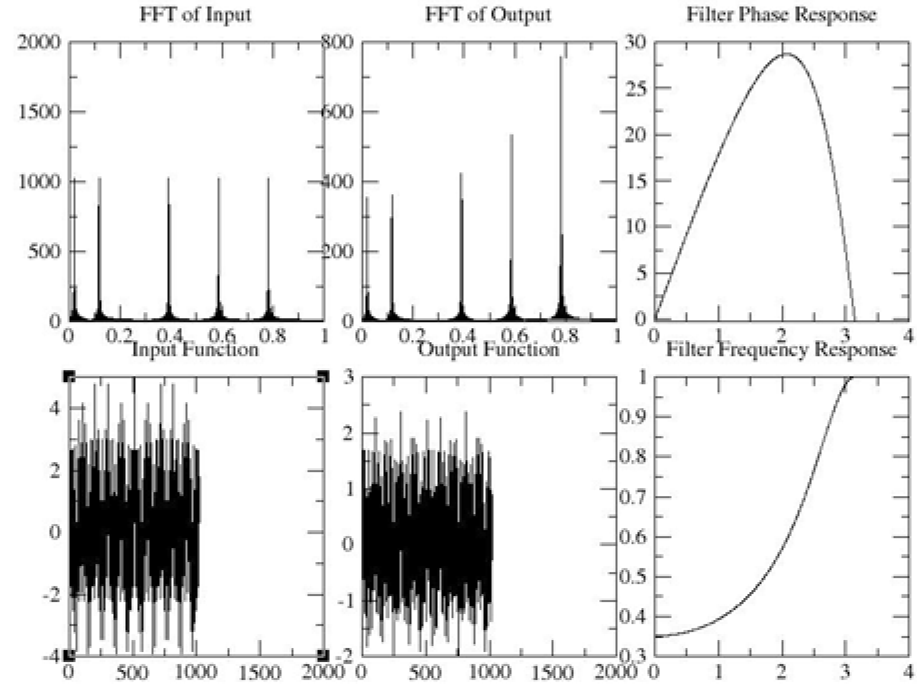
The Transfer Function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1+0.48z^{-1}} \qquad 7.15$$

The maximum Gain at Nyquist Frequency

$$G = \frac{z}{z+0.48} = \frac{e^{j180}}{e^{j180}+0.48} = 1.92$$

To bring the Gain down to a unity at the Nyquist sampling frequency of π Hz, the Transfer Function must be multiplied by a Gain value of 1/1.92 or 0.52. Using the same setting as of the Example 7.1, but changing only the pole0 Phase Angle from 0 to 180 degree we get the response plot as shown in Figure 7.8.



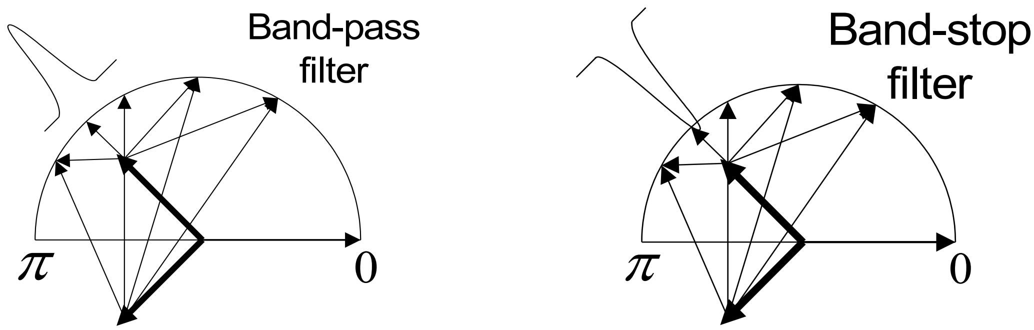
****Insert Figure 7.8 here*****

Figure 7.8. The output of the "filterTest.pl" for the Example 7.2, notice the difference in the output by simply changing the Phase angle from 0 to 180 degree. The filter action is clearly displayed by the rise in the magnitude of the frequencies beyond 100 Hz.

The Band-pass and the Band-stop filters

The response of the real vector poles and zeros was easy to visualize as the maximum magnitude occurred only at the phase angle θ of either 0 or 180 degree. But the effect is different if θ is anything other then 0 and 180 degree. The multiplier $re^{j\theta}$ forms a complex vector if $0 < \theta < 180^{\circ}$. With the complex vector $re^{j\theta}$, there is always a corresponding complex conjugate $re^{-j\theta}$, as shown in the Figure 7.9.a and 7.9.b. The combined magnitude of the two vectors, vary as the vectors move along from 0 to π, but precisely at the Phase angle (θ) the magnitude have its shortest value, resulting in the highest multiplier if it is a pole vector and lowest multiplier if it is a zero vector. This is the multiplier with which the input frequencies are multiplied. Thus, the combined magnitude of the conjugate poles and zeros, create a peak where the phase angle θ is, (see Figure 7.9.a for the pole effect and 7.9.b for the zeros effect).

In the process, the frequencies in the neighborhood of θ get also affected either enhanced or suppressed (based on the vector being a pole or a zero) and such neighborhood is the **Bandwidth** of the filter and is defined as the two adjacent points where the magnitude falls to 0.707 of the peak

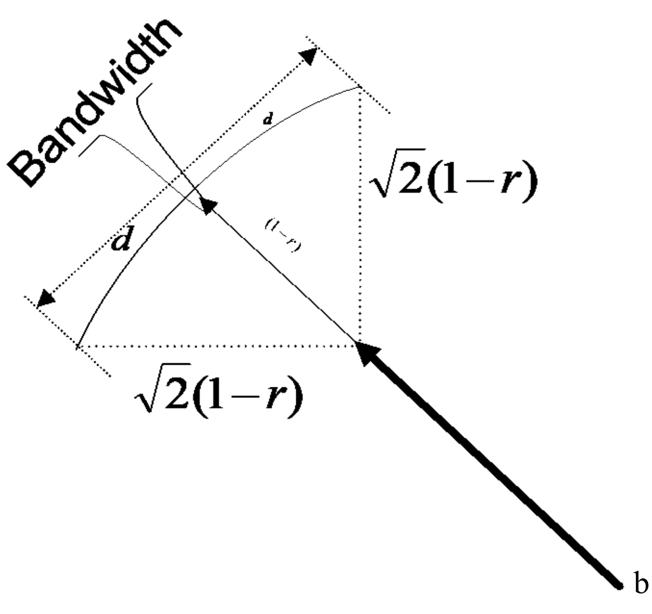


******Insert Figure 7.9 here*******
Figure 7.7.a. The Bandwidth of a band pass filter showing magnitude enhanced due to a complex pole vector, b) magnitude reduced due to a complex zero vector.

The Bandwidth

In order to compute the Bandwidth created by the vector $(1-r)$, imagine the part of unit circle (in the neighborhood of the vector) is a straight line, forming a triangle with the three sides, *d*, $\sqrt{2}(1-r)$ and $(1-r)$ as shown in the Figure 7.10. The distance *d* corresponds to one half of the bandwidth and is equal to the length $(1-r)$, thus the Bandwidth,

$$2d = 2(1-r)$$



******Insert Figure 7.10 here*******
Figure 7.10. The Bandwidth as an approximation of the triangle with the sides, *d*, $\sqrt{2}(1-r)$ and $(1-r)$.

The Band-Pass Digital Filter Transfer Function

The band-pass filters work by enhancing the magnitude of certain range of frequencies from the rest of the inputs, as created by combine magnitude of a pair of complex conjugate pole vectors. A Band-pass filter is specified by a peak frequency and the region surrounding it, where the magnitude falls to 0.707 of its peak magnitude.

The Figure 7.9.a describes the magnitude of the complex pole that has a peak at the angle θ that tapers off on both sides. Thus, a specific band-pass frequency response may be created by strategically placing the Phase angle θ in the pole vector of a filter Transfer Function. The following is the magnitude of a complex pole vector Transfer Function,

The complex conjugate pole vectors are given in the Equation 7.16,

$$H(z) = \frac{1}{(1-re^{j\theta}z^{-1})(1-re^{-j\theta}z^{-1})} = \frac{z^2}{(z-re^{j\theta})(z-re^{-j\theta})} \qquad 7.16$$

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

The Transfer Function Magnitude

$$|H(\Omega)| = 1/\{e^{j\omega}-re^{j\theta}\}\{e^{j\omega}-re^{-j\theta}\} = 1/e^{(2j\Omega)}-2r\cos\theta e^{j\Omega}+r^2$$

Separating the real and the imaginary,

$$|H(\Omega)| = \frac{1}{\sqrt{(1-2r\cos\theta\cos\Omega+r^2\cos 2\Omega)^2+(2r\cos\theta\sin\Omega-r^2\sin 2\Omega)^2}}$$

The Phase response as a function of frequency,

$$\Phi = -\arctan\frac{(2r\cos\theta\sin\Omega+r^2\sin 2\Omega)}{(1-2r\cos\theta\cos\Omega+r^2\cos 2\Omega)}$$

Consider the following example,

Example 7.3

A data acquisition system has a sampling rate of 2048 samples per second; define the Transfer Function magnitude at the peak frequency of 300 Hz and a bandwidth of 5 Hz.

The peak frequency 300 Hz corresponds to sampling frequency

$\theta = \pi 300 \times 2 / 2048 = .293\pi = 52.74^{\circ}$.

The Bandwidth of 5 Hz corresponds to sampling frequency

$1 - r = d = 5/1024 = 0.00488\Omega$

$r = 0.995$

The Transfer Function magnitude

$$|H(\Omega)| = \frac{1}{\sqrt{(1 - 2r\cos\theta\cos\Omega + r^2\cos 2\Omega)^2 + (2r\cos\theta\sin\Omega - r^2\sin 2\Omega)^2}}$$

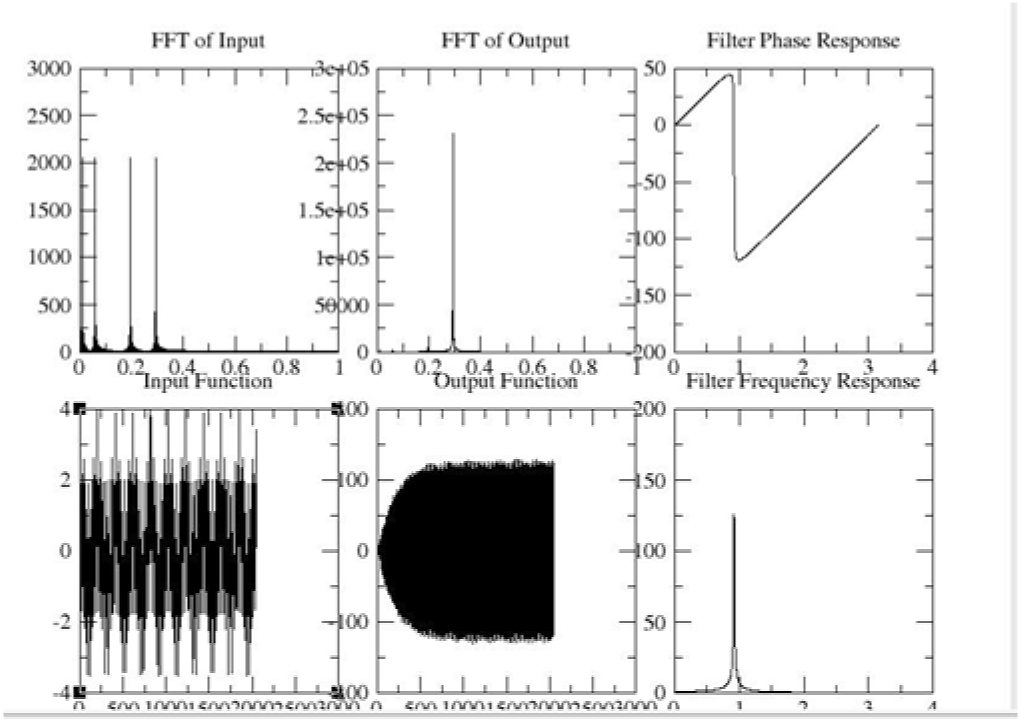
Since the peak occurs at $\theta = \Omega$, the maximum gain at the peak is,

$$|H(\Omega)| = \frac{1}{\sqrt{0.0047^2 + 0.0062^2}} = 128$$

Gain:

A unity gain at 300 Hz may be achieved by multiplying the Transfer Function with a constant value of 1/128=0.008.

Using the “filterTest.pl” program and placing the pole0 magnitude of 0.995 and Phase angle of 57 degree we obtain the different plots of the frequency and phase response as shown in the Figure 7.11.



******Insert Figure 7.11 here*******

The Figure 7.11 shows the frequency and phase response of the Transfer Function of the Example 7.1, as well as the plots of input and out function with their corresponding FFTs.

The Band-Stop Digital Filter Transfer Function

Just like the band-pass filters that work by enhancing the magnitude, the band-stop filters work by suppressing the magnitude of certain range of frequencies from the rest of the input. Consider the same complex vector $re^{j\theta}$ with its complex conjugate $re^{-j\theta}$, but this time instead of pole, it is a zero vector of the Transfer Function,

Having a direct relationship on the input frequencies, the magnitude forms a valley precisely at the phase angle θ. Similar to the band-pass filters, the frequencies in the neighborhood of θ also get affected, but this time they are suppressed. The neighborhood is considered as the two adjacent points where the magnitude is 1.414 of the lowest value.

You would expect that by inverting the poles to zeros, we would be able to change a band pass filter to a band stop filter, to some degree it is correct, but zero vector reduces more rapidly then the corresponding pole vectors (the appearance of the division operation with a small number is more prominent than the multiply operation). The Figure 7.12 is the response of the band stop filter obtained from replacing the pole vector of the Example 7.3 to a zero vector.

The complex vector zero magnitude is,

$$H(z) = (1 - re^{j\theta} z^{-1})(1 - re^{-j\theta} z^{-1}) = \frac{(z - re^{j\theta})(z - re^{-j\theta})}{z} \qquad 7.17$$

Separating the real and the imaginary,

$$|H(\Omega)| = \sqrt{(1 - 2r\cos\theta\cos\Omega + r^2\cos 2\Omega)^2 + (2r\cos\theta\sin\Omega - r^2\sin 2\Omega)^2} \qquad 7.18$$

The Phase response as a function of frequency,

$$\Phi = \arctan\frac{(2r\cos\theta\sin\Omega + r^2\sin 2\Omega)}{(1 - 2r\cos\theta\cos\Omega + r^2\cos 2\Omega)}$$

Consider the following example,

Example 7.4

Convert the band pass filter of the previous example to a band stop filter that would suppress all frequencies in the neighborhood of 300 Hz.

Solution:

The frequency to suppress (300 Hz) corresponds to the sampling frequency

$\theta = \pi 300 \times 2 / 2048 = .293\pi = 52.74^\circ$.

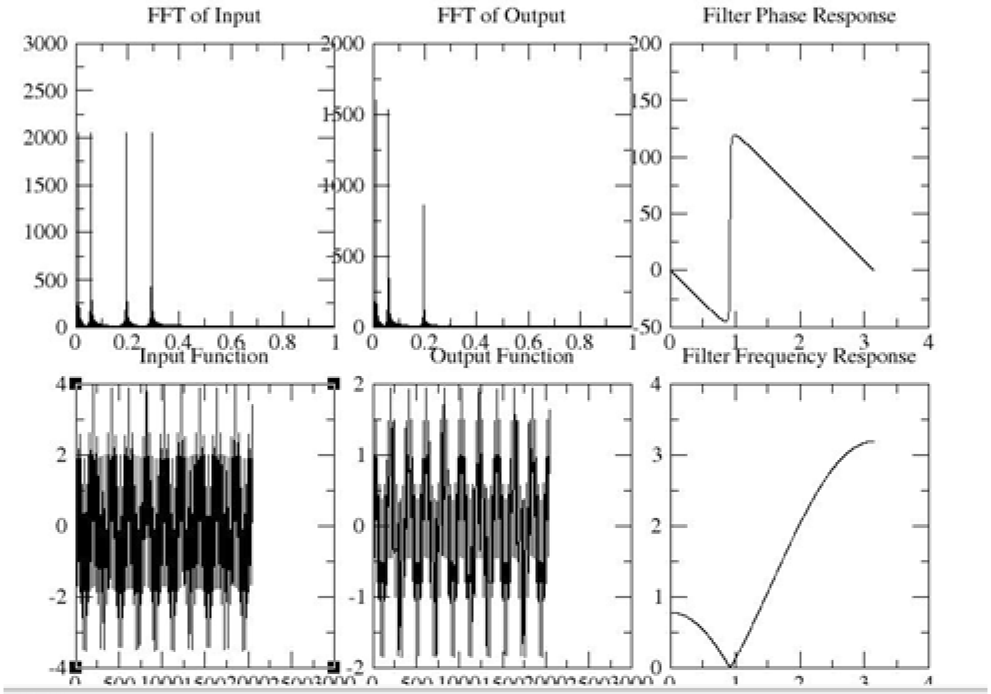
The Bandwidth of 5 Hz corresponds to sampling frequency

$1 - r = d = 5/1024 = 0.00488\Omega$

$r = 0.995$

Substituting the value of r and θ into the Transfer Function Equation 7.18, we obtain the desired Frequency and Phase response.

The Figure 7.12 is the plot of frequency and phase response of the filter with the zero vector $r = 0.995$ and the Phase angle $\theta = 52.74^\circ$, generated by the “filterTest.pl” program.



******Insert Figure 7.12 here*******
The Figure 7.12 shows the frequency and phase response of the Transfer Function of the Example 7.4, as well as the plots of input and out function with their corresponding FFTs.

Notch Filter

You can see that having a single zero vector does not produce a desirable response, instead, a better result may be obtained by a complex zero of magnitude one to the band-pass filter to create the corresponding band-stop or notch filter, as shown in the following example,

The peak frequency 300 Hz corresponds to sampling frequency

$$\theta = \pi 300 \times 2 / 2048 = .293\pi \text{ .}$$

The Bandwidth of 5 Hz corresponds to sampling frequency

$$1 - r = d = 5 \times 2 / 2048 / 2 = 0.00288\Omega$$

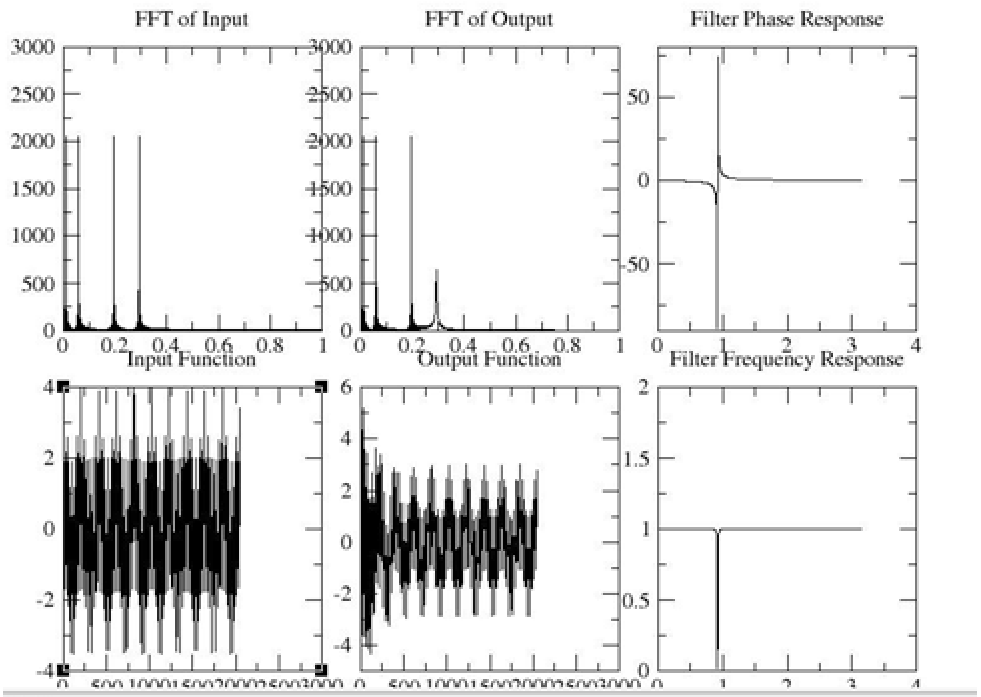
$$r = 0.99756$$

The required Transfer Function with a complex pole at $r = 0.99756$ and Phase $\theta = 293\pi$ and a complex zero at $r = 1$ and Phase $\theta = 293\pi$ is given as,

$$H(\Omega) = \frac{(z - e^{j293\pi})(z - e^{-j293\pi})}{(z - 0.99756e^{j293\pi})(z - 0.99756e^{-j293\pi})}$$

The Transfer Function magnitude is,

$$|H(\Omega)| = \frac{\sqrt{(1 - 1.21\cos\Omega + \cos 2\Omega)^2 + (1.21\sin\Omega - \sin 2\Omega)^2}}{\sqrt{(1 - 1.20\cos\Omega + 0.995\cos 2\Omega)^2 + (1.20\sin\Omega - 0.995\sin 2\Omega)^2}}$$



****Insert Figure 7.13 here*****

The Figure 7.13 shows the frequency and phase response of the Transfer Function of the notch filter of the Example 7.4, as well as the plots of input and out function with their corresponding FFTs.

The distance and the angle of the pole and zero vectors (with reference to the perimeter of the unit radius circle) provide some clues of what the individual vector response is going to be. Being a linear system, the product of the individual vector response is the overall filter response. A system may have several poles and zeros, but the best design is the one that achieves the objective with as few as possible.

Realizing the Filter Response

Once a suitable frequency and phase response is obtained, the next step is to realize the Filter in real time. As you can see, a filter may be realized by multiplying the Transfer Function with the input Transform, but that requires obtaining the inverse Transform of the Transfer Function (a very complex time consuming process). The alternate is to convolve each input with the coefficients of the difference equation that formed the filter Transfer Function (we have shown that Transform multiplication is same as convolution). To obtain the difference equation coefficients we need to convert poles and zeros into a polynomial form.

The polynomial form of the Transfer Function

Once the poles and zeros of the Transfer Function are identified, they must be expanded as polynomial in z, in order to obtain the filter coefficients. The following is a generalized form of the Transfer Function in terms of the poles and zero polynomials,

$$H(z) = \frac{Y_Z}{X_Z} = \frac{q_n z^n \cdots q_2 z^2 + q_1 z^1 + q_0 z^0}{p_n z^n \cdots p_2 z^2 + p_1 z^1 + p_0 z^0}$$

A z^1 is a first order system and z^2 is a second order system etc. In all, you may have to perform the Transform of only the first or the second order system. (Higher order *z* may be factored out to create 1st and second order poles and zeros.) A real pole or a real zero translates to a first order system, whereas a conjugate pole or a conjugate zero results in a second order system, as shown in the Equation 7.19..7.22.

The polynomial of the zero vector Transfer Function,

$$H(z) = (1 - rz^{-1}) = \frac{z - r}{z} \qquad 7.19$$

And the pole polynomial is

$$H(z) = \frac{1}{(1 - rz^{-1})} = \frac{z}{z - r} \qquad 7.20$$

The complex conjugate pole polynomial,

$$H(z) = \frac{1}{(1 - re^{j\theta} z^{-1})(1 - re^{-j\theta} z^{-1})} = \frac{z}{(z - re^{j\theta})(z - re^{-j\theta})}$$

$$H(z) = \frac{1}{1 - 2r\cos\theta z^{-1} + r^2 z^{-2}} \qquad 7.21$$

Similarly, the Transfer Function with complex conjugate zero is given as,

$$H(z) = (1 - re^{j\theta} z^{-1})(1 - re^{-j\theta} z^{-1}) = \frac{(z - re^{j\theta})(z - re^{-j\theta})}{z}$$

$$H(z) = 1 - 2r\cos\theta z^{-1} + r^2 z^{-2} \qquad 7.22$$

It should be noted that any pole frequency vector lying outside of the unit area is a sign of unstable filter. If $r > 1$ then every subsequent output will be greater then the previous one, resulting in an exponential growth without bound, thus the value of $r > 1$ should be avoided.

The design of the digital filters is essentially, selecting coefficients of the governing difference equation that form the polynomial of the filter Transfer Function.

The difference equation

The general form of the difference equation, relating the input and output is given as,

$$y_0(k) + b_1 y(k-1) + \ldots . b_m y(k-M) = a_0 x(k) + a_1 x(k-1) + .. + a_n x(k-N) \qquad 7.23$$

To the left of the Equation 7.23 we have the current output $y_0(k)$ plus all the previous outputs up to the order *M* of the difference equation. The right side of the equation has the current input $x_0(k)$ times its coefficient a_0 plus all the previous inputs multiplied by their coefficients.

Once the coefficients are identified, it is only a matter of reading the current input, multiply with the coefficient and add to the previously computed input and output, to generate the new output (essentially, solving a difference equation through convolution).

The coefficients of the Difference Equation

The previous chapter "Laplace Transform and the z-Transform" described the input and output relationship as a rational polynomial of frequency variable *s* and *z*. The *s* is the continuous time variable $e^{-j\omega t}$ and the z is the discrete time variable $e^{-j\omega k\Delta t}$. The factored form of the numerator polynomial created zeros and the denominator polynomial created poles. Now, we need to reverse the situation, from the newly created poles acquire the denominator polynomial and from the zeros obtain the numerator polynomial.

For example, you have the following poles and zeros,

$$\frac{z(z-0.8)(z+1)}{(z-0.5+j.7)(z-0.5-j0.7)(z-0.8)} \qquad 7.24$$

We can rewrite the factored form of Equation 7.24 as a rational polynomial of the variable z.

$$H(z)=\frac{Y(z)}{X(z)}=\frac{z^3-1.6z^2+0.8z}{z^3-1.64z^2+0.8z+0.8} \qquad 7.25$$

Rewriting the Equation 7.25 as the frequency response multiplied by the Transformed input value we get,

$$z^3Y(z)-1.64z^2Y(z)+0.8zY(z)+0.8Y(z)=z^3X(z)-1.6z^2X(z)+0.8zX(z)) \qquad 7.26$$

The inverse Transform of the Equation 7.26 is the desired time domain convolution as a solution of the difference equation we have been waiting for. But instead of going through the complex process of performing the inverse transform, the following analogy would provide you with a much neater and simpler solution.

Compare the general difference equation,

$$y_0(k)+b_1y(k-1)+....b_my(k-M)=a_0x(k)+a_1x(k-1)+..+a_nx(k-N) \qquad 7.27$$

With the following corresponding Transformed polynomial in power of *z*,

$$z^0Y(z)+b_1z^{-1}Y(z)+....b_mz^{-m}Y(z)=z^0a_0X(z)+a_1z^{-1}X(z)+..+a_nz^{-N}X(z) \qquad 7.28$$

You can see, there is a one to one correspondence between the power of *z* in Equation 7.28 and the sampling delay of the Difference Equation 7.27. The z^{-1} is one sample delay in the past and z^{-2} is two-sample delay in the past. Similarly, z^1 is one sample delay in the future and z^2 is two-sample delay in the future etc. Using the shift property of the z-Transform, we can recover the impulse response from the Equation 7.28 as follows,

$$y[n+3]-1.64y[n+2]+0.8y[n+1]+0.8y[n]=x[n+3]-1.6x[n+2]+0.8x[n+1] \qquad 7.29$$

The Equation 7.29 could be written as a function of variable *y*[*n*],

$$y[n]=1.25x[n+3]-0.75x[n+2]+0.8x[n+1]-1.25y[n+3]+2.05y[n+2]-y[n+1] \qquad 7.30$$

There is just one problem with Equation 7.30, the solution require values from the future. If *n* is the current sample, then [*n+3*] is the third sample in the future. Systems that depend upon values from the future are called the **non-causal** system and it is not possible to implement them in real time. But if your system is linear and time invariant, then the impulse response in the future is the same as impulse response in the past. The Equation 7.30 could be shifted to the values from the past by simply subtracting 3 from each sample time. The sample time [n+3] becomes [n] and [n+2] becomes [n-1] etc., making all the sample time as the values from the current and the past, makes the system a **causal** system and may be implemented as a solution in real time. Thus, the Equation 7.30 is equivalent of

$$y[n] = x[n] - 0.1.6x[n-1] + 0.8x[n-2] - 1.64y[n-1] - 0.8y[n-2] - 0.8y[n-3] \quad 7.31$$

The Equation 7.31 is the desired filter response corresponding to the poles and zeros from the Transfer Function of Equation 7.25. A recursive algorithm may be implemented to realize the output, but you can see, it becomes difficult to compute the rational polynomial as the degrees of the terms are increased. There is a simpler way of dividing the polynomials into sections, as being discussed next.

Cascading Transfer Functions

If you think the number of the coefficients to multiply in one equation is overwhelming, then you may reduce the complexity by dividing up the pole and zeros into sections and feed the output of one to the input of the other. A real pole and real zero is a first order section and a conjugate pole and zero is a second order section. The Equation 7.25 may be reduced to the following sections,

Section 1:

$$\frac{U(z)}{X(z)} = \frac{z}{(z-0.8)}$$

$$u[n] = x[n] + 0.8u[n-1]$$

Section 2:

$$\frac{V(z)}{U(z)} = \frac{(z-0.8)}{(z-0.5+j.7)(z-0.5-j0.7)}$$

$$v[n] = u[n-2] - 0.8u[n-1] + v[n-1] + 0.74v[n-1]$$

Section 3:

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

$$y[n] = v[n] + v[n-1]$$

Based on the techniques of poles and zeros mentioned above, we are ready to implement difference equation of some simple digital filters.

Low pass filter realization

The Transfer Function of the low pass filter of the Example 7.1 is given as

$$\frac{Y_z}{X_z} = \frac{z}{z - r}$$

$$zY_z - rY_z = zX_z$$

The corresponding difference equation in the causal form

$$y[n] = x[n] + r * y[n-1] \qquad 7.32$$

Substituting the value of $r = 0.48$ and solving the Equation7.32 with an iterative algorithm, we realize a low pass filter corresponding to the Example 7.1.

An iterative algorithm may be implemented to generate the corresponding Filter output. The application program "filterTest.pl" generates the series of input values ($x[0], x[1]...x[n]$) corresponding to the test frequencies and feeds the values into the Equation 7.32 and generates the corresponding output $y[0], y[1]...y[n]$. The Figure 7.7 shows the corresponding input and output function plot.

High pass filter realization

The Transfer Function of the high pass filter of the Example 7.2 is given as

$$\frac{Y_z}{X_z} = \frac{z}{z + r}$$

$$zY_z + rY_z = zX_z$$

The corresponding difference equation in the causal form

$$y[n] = x[n] - r * y[n-1] \qquad 7.33$$

Substituting the value of $r = 0.48$ and solving the Equation7.33 with an iterative algorithm, we realize a highs filter corresponding to the Example 7.2

Similar to the low-pass filter, the output generated by the application program "filterTest.pl" is shown in the Figure 7.8.

Example 7.5

A Data Acquisition System has a sampling rate of 1024ples per second. Formulate the difference equation that would suppress all frequencies beyond 20 Hz.

Solution:

The Nyquist frequency:
$1024/2 = 512Hz$

The pole frequency as a fraction of the Nyquist frequency:

20/512= 0.039 of the Nyquist frequency.

The pole distance:
$r = (1 - 0.039) = 0.966$

The maximum value of the peak is at 0 degree,
$\theta = 0$

The cutoff frequency vector

$re^{j0} = 0.966$

The Transfer Function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z}{z - 0.966}$$

The frequency domain multiplication

$zY(z) - 0.966Y(z) = zX(z)$

The convolution in time domain,

$$y[n] = x[n] + 0.966y[n-1] \qquad 7.34$$

The Equation 7.34 when implemented as an iterative algorithm produces the desired filter output. The time series output may be generated using the test program "filterTest.pl". Enter 0.966 for Pole0 magnitude and enter 0 for the Pole0 phase.

The band-pass and band-stop filter realization

The Transfer Function corresponding to the Band pass filter of the Example 7.3 is,

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

Substituting the required value of $\theta = 293\pi$ and $r = 0.995$ into the Transfer Function,

$$z^2 Y_z - 1.20 z Y_z + 0.99 Y_z = z^2 X_z \qquad 7.35$$

The causal form of the Equation 7.35

$$y[n] = x[n] + 1.20 y[n-1] - 0.99 y[n-2] \qquad 7.36$$

The output of the Equation 7.36, generated by the application program "filterTest.pl" is shown in the Figure 7.9.

Practice your skill with the graphical user interface

You could practice creating poles and zeros and see the response of the filter Transfer Function with the graphical user interface program provided as an aid to the user. The program "filterTest.pl" (the listing is presented in Appendix A), displays the user interface as shown in the Figure 7.6. The user may specify up to two different poles and two different zeros and the software display a plot of the magnitude and the phase response. The range of the vector *r* is between 0 and 1 and the phase angle between 0 and 180 degree. The pole and zero vectors are real if the phase is 0 or 180; otherwise, it is a complex conjugate vector. For test signals, you can select up to six input frequencies of equal magnitude, you can specify sampling rate varying between 2 and 65536 points. The program also displays the location of the poles and zeros with respect to the origin of the semi circle. A cross indicates the pole-frequency vector and a circle indicates the zero-frequency vector, inside or right on the edge of big semi circle. The program displays the following six diagrams,

1) Time response of the input data points,
2) FFT of the input data
3) Time response of filter output for one complete cycle
4) FFT of output data
5) Frequency response of the filter
6) Phase response of the filter

Press 'test' button to view the different plots (see Figure 7.6).

The following helpful hints would give you some intuitive feelings about the effect of different values of *r* and the phase angle θ on the filter response. Practice the helpful hints with the program "filterTest.pl" and verify the output.

1)

If the roots of the poles are real and negative, the filter is a high-pass filter; the peak value is at 180 degree,

$z + r = 0$, $z = -r$

Using the test program “filterTest.pl”, select input frequencies 1,100,200,300,400,500. Enter 0.8 for the value of the Pole0 radius; enter 1 for Zero0 radius, 180 for the Pole0 phase, 180 degree for the Zero0 and press ‘Test’. You should see frequencies below the 80 percent bandwidth are suppressed.

2)

If the roots of the poles are real and positive, the filter is a low-pass filter, the peak value is at 0 degree,

$z - r = 0$, $z = r$

On the graphical user interface of “filterTest.pl”, change the value of pole0 and Zero0 phase angle from 180 to 0; press ‘Clear’, press ‘Test’. The filter is changed from high-pass to low-pass filter.

3)

A complex pole as well as a complex zero creates a short band-pass filter. The peak is at the phase angle θ of the complex vector

Change the value of pole0 and Zero0 phase angle to 30; press ‘Clear’, press ‘Test’. The filter is changed to a narrow band-pass filter.

Increase the bandwidth by reducing the *r* vector length.

4)

The same combination of complex pole and zeros create band-stop filter, if the zero vector magnitude is greater then the pole vector magnitude. The notch occurs at the phase angle θ.

Continuing from the previous example, change the value of pole0 magnitude from 0.8 to 1 and Zero0 magnitude from 1 to 0.8; press ‘Clear’, press ‘Test’. The filter is changed to a narrow band-stop filter.

The depth of the notch depends upon the difference between the radius of the pole and zero vectors. Increase the bandwidth by reducing the *r* vector length.

5)

The response magnitude peak of the pole corresponds to the value $1 - r$ (the minimum length of the pole).

6)

The 3dB point, where the magnitude falls to $1/\sqrt{2}$ of its peak value determines the bandwidth of the pass-band. The bandwidth is approximately 2 times the vector length $1 - r$.

You can design simple filters by selecting poles and zero locations and viewing the response, but a precision filter requires other properties, such as a steeper roll off of the pass band region etc. Ideally, one would like to establish a cutoff frequency and expect a filter to preserve the pass-band range while sharply rejecting the stop-band. But such filters are not physically realizable, as they require infinite computations. Still, design methods are available to improve upon the roll off rate and achieve a good enough response that closely matches the ideal response.

Improving the design

One of the most important considerations in any filter design is to see how well the filter suppresses the frequencies in the roll off region. Although, we would like to see an action like a switch; all frequencies before the cutoff point appear in the system and the rest disappear. But if it is not possible then, preferably, have as narrow a region as possible. There is a trade-off between the flat response and sharp cutoff frequency. Different methods enhance different aspects of the filter characteristics and we have discussed Butterworth Filters in the previous chapter, here we will discuss Butterworth design from the Digital Filter's point of view. We will also discuss later Chebyshev's polynomials as an alternate way to improve the filter response.

Butterworth Filters

An explained in chapter 6, the Butterworth filters put extra weight by sharply increasing the pole magnitude at the cutoff point. It is like having multiple pole vectors at the same cutoff point.

$$|H(j\omega)|^{2N} = \frac{1}{1+(1/\omega_C)^2\omega^2+(1/\omega_C)^4\omega^4\cdots(1/\omega_C)^{2N}\omega^{2N}}$$

The poles are the roots of the equation,

$$(e^{j\omega\pi})^N = -1 \qquad 7.37$$

The Equation 7.37 indicates, there are *N* numbers of poles (please see the Chapter 6 for detailed explanation). The followings are the pole vectors for different values of *N*,

$$N=1 \qquad \frac{1}{(z-e^{j\pi 3/4})}$$

$$N=2 \qquad \frac{1}{(z-e^{j\pi 3/4})(z-e^{-j\pi 3/4})}$$

$$N=3 \qquad \frac{1}{(z-e^{j\pi 3/4})(z-e^{-j\pi 3/4})(z-e^{j\pi 5/4})(z-e^{-j\pi 5/4})}$$

Butterworth filters may be realized by cascading several single pole filters, where the output of one is the input to others.

Chebyshev filters:

Chebyshev filters provide a polynomial fit (as a weighted function) to a given Transfer Function, but with the consequences of producing ripples in the pass band or the stop band region.

The Chebyshev's series is defined as,

$$T_n(\omega) = \cos(n \arccos \omega) \qquad 7.38$$

The Figure 7.14 is the plot of the series with increasing *n*. You can see that the points on the x-axis are much closer at the end. This unequal spacing tends to put more weight at the end to improve upon the response near the edge.

The first few terms of the series are expanded as,

$$
\begin{aligned}
&T_0(\omega) = 1 \\
&T_1(\omega) = \omega \\
&T_2(\omega) = 2\omega^2 - 1 \\
&T_3(\omega) = 4\omega^3 - 3\omega \\
&T_4(\omega) = 8\omega^4 - 8\omega^2 + 1 \\
&T_5(\omega) = 16\omega^5 - 20\omega^3 + 5\omega \\
&T_6(\omega) = 32\omega^6 - 48\omega^4 + 18\omega^2 - 1 \\
&T_7(\omega) = 64\omega^7 - 112\omega^5 + 56\omega^3 - 7\omega \\
&\vdots \\
&T_{n+1}(\omega) - 2\omega T_n(\omega) + T_{n-1}(\omega) = 0
\end{aligned}
$$

The function has the property that for x < 1 it is cosine function but for x > 1 it becomes a hyperbolic function. If we normalize w with the cutoff frequency wc then for w/wc < 1 we have cosine function and for w/wc > 1 we have hyperbolic function, producing a steep drop in the magnitude of the Transfer Function at the cutoff point of w/wc = 1, which is also the transition point from pass band to stop band.

$$T_N(\omega) = \begin{vmatrix} \cos(N \cos^{-1} \omega) & \quad |\omega| \le 1 \\ \cosh(N \cos^{-1} \omega) & \quad |\omega| > 1 \end{vmatrix}$$

The Transfer Function generated by the Chebyshev polynomials is given as,

$$|H(\omega)|^2 = \frac{1}{1+\varepsilon^2[T_n(\omega/\omega_c)]^2} \quad 7.39$$

The characteristics of the Chebyshev filters is that we do achieve a sharper transition (steeper roll-off), but on the expanse of some distortion (ripples) either in the pass-band or the stop-band region.

The ω_c in Equation 7.39 is the cutoff frequency and as long as $|\omega_c| << |\omega|$, the term $H(\omega)$ has the following range

$$\frac{1}{1+\varepsilon^2} \leq |H(\omega)| \leq 1$$

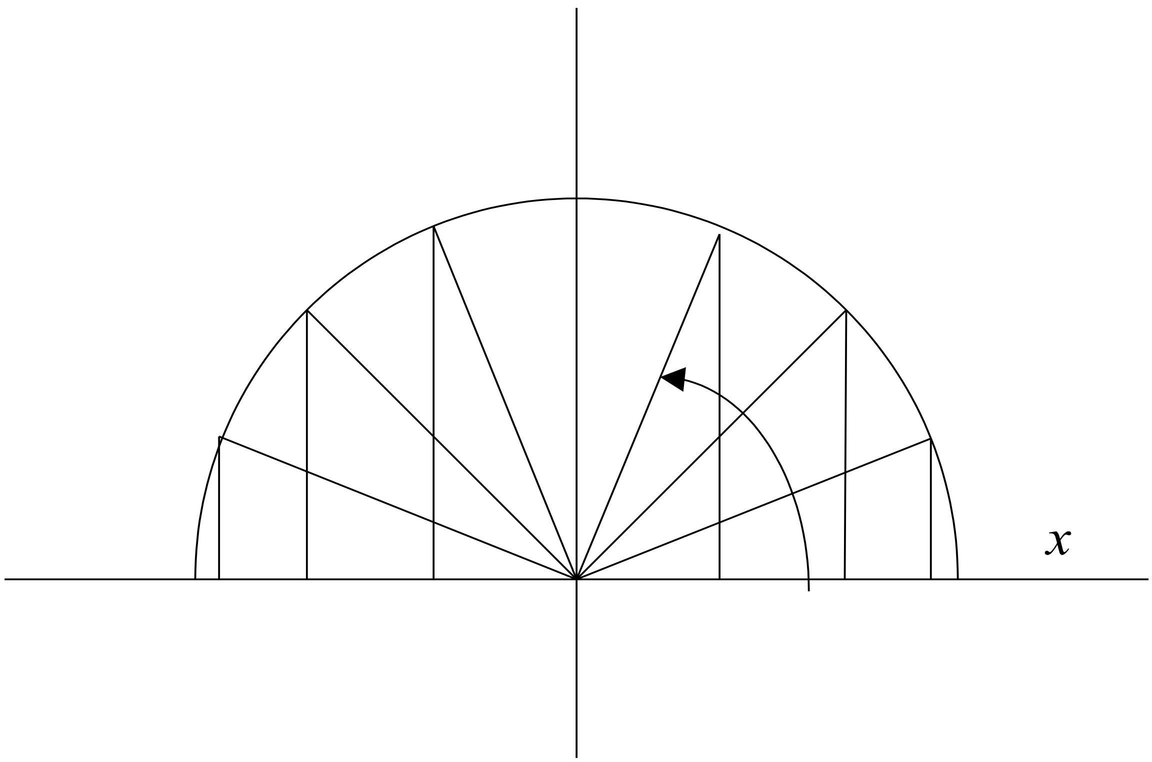
And at $\omega_c = \omega$ which is also the edge, we have $T_n(1) = 1$ indicating the gain is 1 at the cutoff point, that meets our criteria.

The ripples produced by the filter in the pass-band region are bounded by the interval $\frac{1}{1+\varepsilon^2}$.

The condition at the edge of the stop-band where $\omega_s = \omega$ is

$$\frac{1}{1+\varepsilon^2 T_n^2(\omega_s/\omega_c)} = \frac{1}{A^2}$$

$$T_n(x) = \cos(n \arccos x)$$



******Insert Figure 7.14 here*******

The Figure 7.14. The plot of the chebyshev series,

The Figure 7.15 is a Butterworth filter with the cutoff frequency of 1/3 of the Nyquist Frequency,

```
%impulse response
clear
clf
colordef white;
[b,a]=cheby1(10,0.5, 300/1000);
[h,w]=freqz(b,a,256,2000);
plot(w,abs(h),'-k');
```

Transition from analog to digital Filters

Transforming an analog filter into digital form in the z-transform is basically an operation of frequency warping, what that means is that the response of the analog filters that theoretically extends to infinity should be mapped to a function that extends to Nyquist frequency or one cycle of π radians per sample only. It should also include the repeat pattern of the response due to alias frequencies.

The pole frequency variable ω_a of the analog filters Transfer Function, which is a function in sin and cosine, is replaced with the following tangent function (using half angle formula of trigonometry) of digital discrete frequency ω_d of radians per sample.

$$\omega_a = \frac{2}{\Delta t}\tan\frac{\omega_d \Delta t}{2} \qquad 7.40$$

Let's take a simple analog low-pass *RC* filter of a single pole with the following Transfer Function,

$$H(\omega_a) = \frac{A}{\omega_a + p_i} \qquad 7.41$$

Substitute the value of ω_a as given in Equation 7.40,

$$H(\omega_a) = \frac{A}{j(\frac{2}{\Delta t}\tan\frac{\omega_d \Delta t}{2}) + p_i} \qquad 7.42$$

A plot of the new Transfer Function of Equation 7.42 is shown in the Figure 7.15.a together with the corresponding analog Transfer Function in the Figure 7.15.b. A comparison of the indicates that the response of the two is identical up to the cutoff frequency, but then the analog response approaches 0 only at frequency $\omega_a \to \infty$, whereas the digital filter response approaches 0 at the Nyquist frequency. Beyond the Nyquist point the digital response repeats the pattern for the next increment of pi radians,

Replacing the analog frequency variable with its equivalent digital frequency variable function as given in the Equation 6.1 serves our purpose of converting an analog filter into its equivalent digital filter.

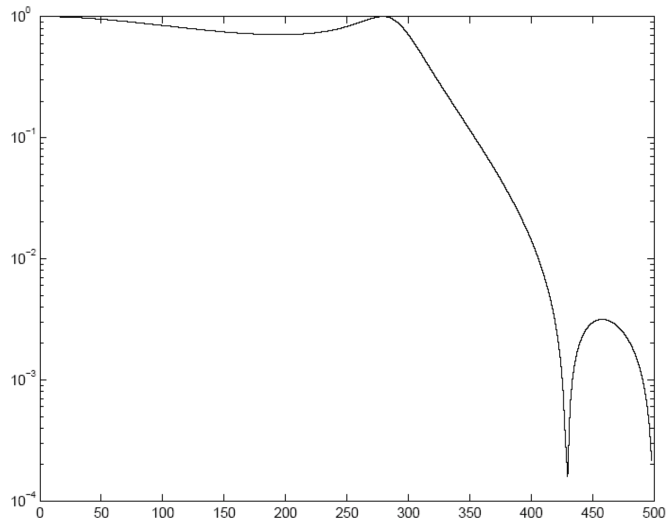
We can rewrite the Equation 7.40 into the familiar exponent form,

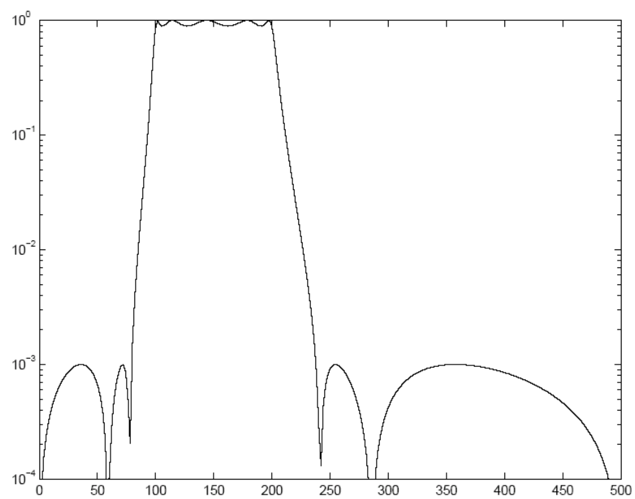
$$\frac{2}{\Delta t}\tan\frac{\omega_d \Delta t}{2} = \frac{2}{\Delta t}\frac{1-e^{-\omega_d \Delta t}}{1+e^{-\omega_d \Delta t}} \qquad 7.43$$

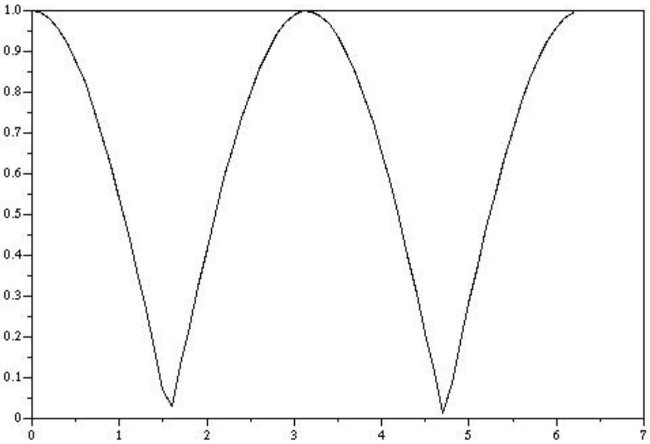
Substituting the standard notation of $z = e^{\omega_d \Delta t}$ into Equation 7.43,

$$\omega_a = \frac{2}{\Delta t}\frac{1-z^{-1}}{1+z^{-1}} \qquad 7.44$$

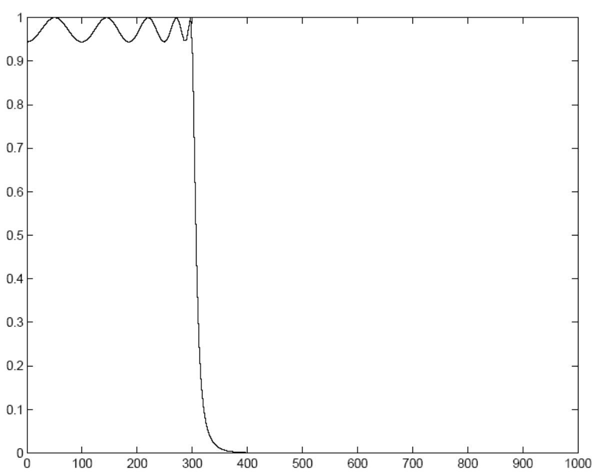
The relationship in Equation 7.44 is known as bi-linear z-Transform that relates an analog frequency to its digital form.

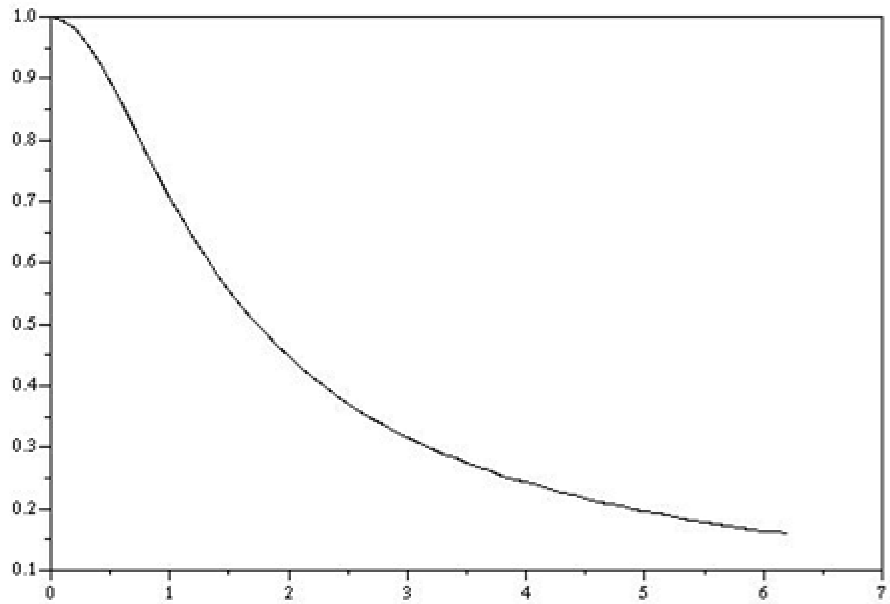


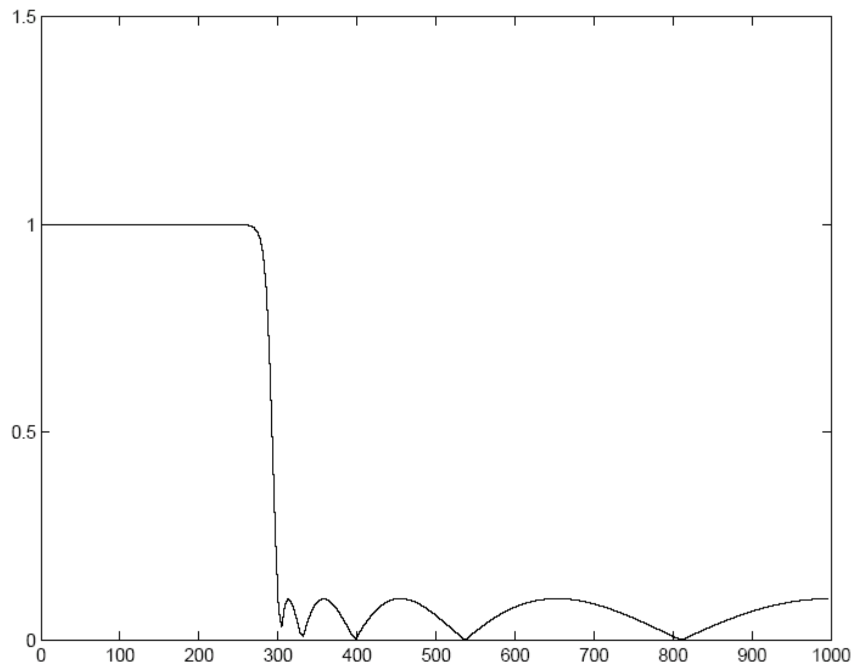




****Insert Figure 7.15.a here*****







****Insert Figure 7.15.b here*****

Example 7.6

Let's analyze the analog *RC* filter response by replacing it with the bi-linear z-transform of Equation 7.44,

$$H(\omega_a) = \frac{\omega_a}{\omega_a + p}$$

$$H(\omega_a) = \frac{(\omega_a\,\Delta t/2)(1+z^{-1})}{((\omega_a/2)+1)+(\omega_a\,\Delta t/2-1)z^{-1}} \qquad 7.45$$

Expanding the Equation 7.45 in terms of z,

$$H(z) = \frac{(\omega_a\,\Delta t/2)+(\omega_a\,\Delta t/2\,z^{-1})}{((\omega_a/2)+1)+(\omega_a\,\Delta t/2-1)z^{-1})}$$

And we obtain the difference equation as,

$$y(n) = \omega_a \Delta t/2\,x[n] + \omega_a \Delta t/2\,x[n-1] - (\omega_a/2+1)\,y[n] - (1+\omega_a \Delta t/2)\,y[n-1] \qquad 7.46$$

.

Example 7.7

What is the equivalent cutoff analog frequency, if a digital filter is design with the sampling rate of 1K samples per second and a cutoff frequency of 700π radians per sample?

From the sampling rate, we compute the sampling period,

$\Delta t = 2\pi/1000 = .002\pi$ rad s^{-1}

The sampling frequency is

$\omega_s = 2\pi \times 10^3$ rad s^{-1}

The digital filter cutoff point

$$\omega_{dc}\Delta t = (0.7\times 10^3).002\pi = 0.2\pi$$

The equivalent analog filter cutoff is,

$$\omega_{ac}\,\Delta t/2 = \tan(\omega_{dc}\,\Delta t/2) = \tan(0.2\pi) = 0.3$$

Example 7.8

Design a digital filter with no ripples in the pass-band or stop-band region. The sampling rate is 1 K samples per second with a cutoff frequency of 1 KHz. The filter must reduce frequencies beyond 1.5 KHz to more then 20 dB.

Solution:

Choice is a Butterwirth filter, since Chebyshev and elliptical filters have ripples in pass-band region.

The frequency in radian per sample:

$$\omega_{d1}\Delta t = 2\pi \times 10^3 \times (1/10^3) = 2\pi \text{ rad s}^{-1}$$

$$\omega_{d2}\Delta t = 2\pi \times 1.5 \times 10^3 \times (1/10^3) = 3\pi \text{ rad s}^{-1}$$

The corresponding analog frequencies are,

$$\omega_{a1} = \tan(\omega_{d1}\,\Delta t/2\,2) = \tan(0.1\pi) = 0.325$$

$$\omega_{a2} = \tan(\omega_{d2}\,\Delta t/2\,2) = \tan(0.2\pi) = 0.726$$

The magnitude of the Butterwoirth nth order filter from Equation 6.1 is given by,

$$|H(\omega)|^2 = \frac{1}{1+(\omega/\omega_c)^{2n}}$$

Where $\omega_c = \omega_{a1} = 0.325$. The order n to be computed from the condition of 20 dB down at $\omega_{a2} = 0.726$

$$1+(0.726/0.325)^{2n} = 20$$

$$n = 1.83$$

We take the next increment

$$n = 2$$

The poles of the Butterwirth filters s_1 and s_2

$$s_1 s_2 = 0.325(-0.707 \pm j0.707) = -0.23 \pm j0.23 \text{ And no zeros}$$

$$H(s) = \frac{s_1 s_2}{(s+s_1)(s+s_2)} = \frac{0.1058}{s^2 + 0.46s + 0.1058} \qquad 7.47$$

Replacing s with bi-linear z-Transform $s = \frac{1-z^{-1}}{1+z^{-1}}$ in Equation 7.47,

$$H(z) = \frac{0.1058}{\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^2 + 0.46\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 0.1058} \qquad 7.48$$

Simplifying Equation 7.48,

$$H(z) = 0.0676\frac{1+2z^{-1}+z^{-2}}{1-1.141z^{-1}+0.413z^{-2}}$$

The corresponding difference Equation

$$y(k) - 1.141y(k-1) + 0.413y(k-2) = 0.0676x(k) + 0.135x(k-1) + 0.676x(k-2)$$

That can be computed with an iterative computer algorithm.

The frequency response is given by

$$H(\omega) = 0.0676\frac{1+2e^{-j\omega\Delta t}+e^{-j2\omega\Delta t}}{1-1.141e^{-j\omega\Delta t}+0.413e^{-j2\omega\Delta t}}$$

Conclusion

In this chapter, the recursive forms of the digital filters were discussed. We used poles and zeros to formulate the Transfer Function corresponding to the filter frequency response. The filter output was realized through implementing convolution as the solution of the difference equation, whose coefficients were obtained from the filter Transfer Function. The frequency and phase response was plotted using graphical user interface test software. The program also generated the coefficient values corresponding to the difference equation that formed the filter Transfer Function. The user could enter choice of test patterns and sampling rate and the program generated filter output for one complete cycle.