# Chapter 3 The *z*-Transform and Digital Filters

For a linear time-invariant (LTI) continuous-time system, the input-output relation of the system is described by a differential equation. To solve the differential equation, the Laplace transform is often used. For a discrete-time system, the input-output relation is expressed in terms of a *difference equation* which consists of multiplications and summations. As the Laplace transform is used to describe analog systems, the *z*-transform is used to describe discrete-time systems. In this chapter, the *z*-transform is defined and the relationship between the *z*-transform and the Laplace transform is discussed. Also, discrete-time systems are classified according to the length of the impulse response. Several methods to compute the frequency response of a discrete-time system are going to be introduced. Finally, a design procedure of digital filters is explained and practical considerations on implementation of digital filters are discussed.

## 3.1 The *z*-Transform

The Laplace transform of a continuous-time signal $x(t)$ is defined as

$$X(s) = L\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-st}dt. \tag{3.1}$$

where $L\{\}$ denotes the Laplace transform operation and the complex variable $s = \sigma + j\omega$. The Laplace transform evaluated on the $j\omega$ axis results in the Fourier transform. The Laplace transform of the impulse-sampled signal $x_s(t)$ described in the previous chapter can be expressed as

$$L\{x_s(t)\} = \int_{-\infty}^{\infty}\left(\sum_{n=0}^{N-1} x(nT)\delta(t-nT)\right)e^{-st}dt \tag{3.2}$$

if there are $N$ nonzero samples. By changing the order of the integration and the summation the transform becomes

$$L\{x_s(t)\} = \sum_{n=0}^{N-1} x(nT)\left(\int_{-\infty}^{\infty}\delta(t-nT)e^{-st}dt\right). \tag{3.3}$$

Because the integral in Eq. (3.3) collapses to $e^{-snT}$, the transform becomes

$$L\{x_s(t)\} = \sum_{n=0}^{N-1} x(nT)e^{-sTn}. \tag{3.4}$$

Note that the Laplace transform of the impulse-sampled signal has nothing to do with the integration with respect to the continuous time $t$. Instead, the transform becomes the summation in terms of a discrete-time index $n$. By introducing a new complex variable $z$

$$z = e^{sT} = e^{(\sigma + j\omega)T} = e^{\sigma T} e^{j\omega T} = re^{j\theta} \quad (3.5)$$

where the real variable $r$ is termed the radius, the transform becomes (with $x(nT)$ replaced by $x(n)$ as in Chapter 2)

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n}. \quad \textbf{z-transform} \quad (3.6)$$

Eq. (3.6) is defined as the $z$-transform of a sequence $x(n)$. When the sequence $x(n)$ has the infinite length, the $z$-transform of $X(z)$ of $x(n)$ is defined alternatively as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}. \quad \textbf{z-transform of a infinitely long sequence}$$

Note that the integration with respect to $t$ in the Laplace transform is replaced by the summation with respect to $n$ in the $z$-transform.

Now let us examine the relationship between the Laplace transform and the $z$-transform. When $\sigma$ is zero, the radius $r$ is one (or unity) in Eq. (3.5). Let us assume that $\sigma$ is zero for the time being. If $\omega = 0$, then $z = 1$. If $\omega = \pi/(2T)$, then $z = j$. If $\omega = \pi/T$, then $z = -1$ and so on. Thus, the frequency interval between $-\pi/T$ and $\pi/T$ on the $j\omega$ axis in the Laplace transform domain ($s$ plane) is mapped onto the unit circle $e^{j\theta}$ once in the $z$-transform domain ($z$ plane) as shown in Figure 3.1. Note that $e^{j\theta}$ makes a unit circle when $\theta$ varies from $-\pi$ to $\pi$ or from 0 to $2\pi$.

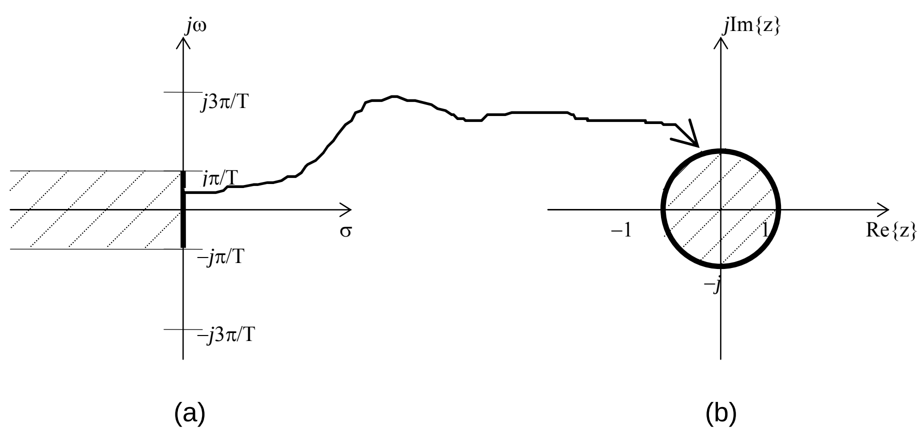



Figure 3.1 Relation between two transform domains: (a) *s* plane (b) *z* plane.

When $\sigma$ is negative, the radius $r$ is less than one. This means that the strip in the left half of the $s$ plane (shaded area) maps inside the unit circle in the $z$ plane. The frequency interval between $\pi/T$ and $3\pi/T$ in the $s$ plane is mapped onto the unit circle in the $z$ plane again and the frequency interval between $3\pi/T$ and $5\pi/T$ in the $s$ plane is mapped onto the unit circle in the $z$ plane again, and so on. This indicates that the entire left half plane of the $s$ plane is mapped inside the unit circle. Like the Laplace transform evaluated on the $j\omega$-axis is referred to as the Fourier transform, a $z$-transform evaluated on the unit circle at $N$ points equally spaced in angle is referred to as the discrete Fourier transform (DFT).

## 3.2 *z*-Transform Examples

In this section examples of some $z$-transforms and inverse $z$-transforms are given. Readers are urged to go over all the examples.

**Example 3.2.1**

A continuous-time impulse response $h(t) = e^{-2t}u(t)$ where $u(t)$ is the unit step function.

(a) Find the Laplace transform of $h(t)$.

(b) The signal $h(t)$ is sampled at every 0.1[s] to generate $h(n) = a^n u(n)$ where $a = e^{-.2}$. Find the $z$-transform of $h(n)$.

Solution

(a) $H(s) = \int_0^\infty e^{-2t} e^{-st} dt = \dfrac{1}{s+2}$.

(b) $H(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} \left(az^{-1}\right)^n = \dfrac{1}{1-az^{-1}} \left( = \dfrac{1}{1-e^{-2(.1)}z^{-1}} \right)$.

The summation converges when $|az^{-1}| < 1$ which implies $|z| > |a|$. Thus, we say that the region of convergence (ROC) is that $|z| > |a|$. Note that the Laplace transform pole is at $s = -2$ and the $z$-transform pole is at $z = e^{-2T}$.

**Example 3.2.2**

Find the $z$-transform of $x(n) = \delta(n-n_0)$.

Solution

$X(z) = z^{-n_0}$. ROC: $|z| > 0$.

**Example 3.2.3**

Find the $z$-transform of $x(n) = u(n) - u(n-N)$.

Solution

$X(z) = \sum_{n=0}^{N-1} z^{-n} = \dfrac{1-z^{-N}}{1-z^{-1}}$ ROC: $|z| > 0$.

**Example 3.2.4**

Find the inverse $z$-transform of $X(z) = \dfrac{1}{1 - az^{-1}}$ with the ROC: $|z| > |a|$.

Solution

From Example 3.2.1, $x(n) = a^n u(n)$.

**Example 3.2.5**

Find the inverse $z$-transform of $X(z) = \dfrac{1}{(1 - az^{-1})(1 - bz^{-1})}$.

Solution

$$X(z) = \frac{A}{1 - az^{-1}} + \frac{B}{1 - bz^{-1}}$$

where $A = \left.\dfrac{1}{1 - bz^{-1}}\right|_{z=a} = \dfrac{a}{a - b}$ and $B = \left.\dfrac{1}{1 - az^{-1}}\right|_{z=b} = \dfrac{b}{b - a}$.

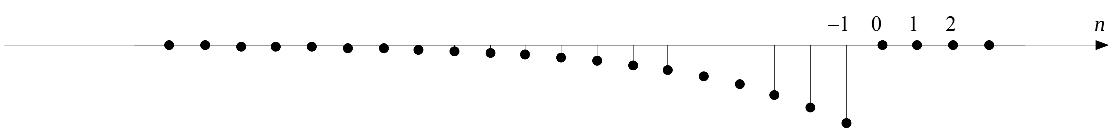
$$X(z) = \frac{1}{a - b}\left[\frac{a}{1 - az^{-1}} - \frac{b}{1 - bz^{-1}}\right]$$

Thus, $x(n) = \dfrac{1}{a - b}\left[a^{n+1} - b^{n+1}\right]u(n)$.

So far the sequences we dealt with were *right-sided sequences*. A right-sided sequence means that the sequence has zeros for $n < n_0$ where $n_0$ is an integer. A left-sided sequence is the sequence that has zeros for $n \geq n_0$. For example, an autocorrelation sequence, that is called a double-sided sequence, can be divided into the left-sided sequence and the right-sided sequence. Readers are encouraged to go over the following two examples.

**Example 3.2.6**

Find the $z$-transform of the left-sided sequence, $x(n) = -a^n u(-n-1)$.

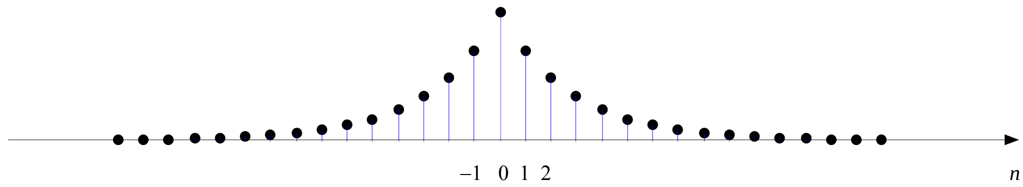



Solution

$$X(z) = \sum_{n=-\infty}^{-1} -a^n z^{-n} = -\sum_{n=1}^{\infty} \left(a^{-1}z\right)^n = \frac{-a^{-1}z}{1 - a^{-1}z} = \frac{1}{1 - az^{-1}} \quad \text{ROC: } |z| < |a|.$$

**Example 3.2.7**

Find the $z$-transform of the double-sided sequence, $x(n) = a^{|n|}$.




Solution

The sequence can be decomposed into a sum of two sequences: $x(n) = a^n u(n) + a^{-n}u(-n-1)$. The first term is a right-sided sequence and the second term is a left-sided sequence. The $z$-transform of the first sequence is $\frac{1}{1-az^{-1}}$ with ROC of $|a| < |z|$. The $z$-transform of the second term is $-\frac{1}{1-\frac{1}{a}z^{-1}}$ with ROC of $|z| < |1/a|$. Thus, the $z$-transform of $x(n)$ is

$$X(z) = \frac{1}{1-az^{-1}} - \frac{1}{1-\frac{1}{a}z^{-1}} \quad \text{ROC: : } |a| < |z| < |1/a|.$$

Note that in the case of Example 3.2.7, the region of convergence is specified by a ring as shown in Fig. 3.2.

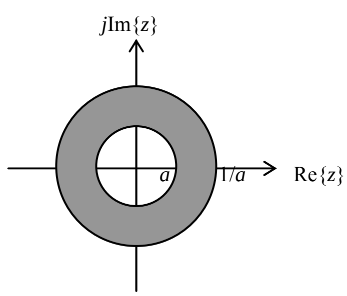



Fig. 3.2 The region of convergence is specified by a ring.

## 3.3. Properties of the *z*-Transform

Let $x(n) \xleftrightarrow{z} X(z)$ implies that $x(n)$ and $X(z)$ are in the $z$-transform relation. The following are some of the important properties of the $z$-transform.

1. $ax_1(n) + bx_2(n) \xleftrightarrow{z} aX_1(z) + bX_2(z)$

   This property is called the *linearity*. This holds because the summation in the $z$-transform is a linear operation.

2. $x(n-n_0) \xleftrightarrow{z} z^{-n_0}X(z)$

The delay of $n_0$ in the time domain results in the multiplication of $z^{-n_0}$ in the frequency domain. A unit delay corresponds to the multiplication of $z^{-1}$.

(Proof) $$\sum_{n=n_0}^{N-1+n_0} x(n-n_0)z^{-n} = \sum_{m=0}^{N-1} x(m)z^{-(m+n_0)} = z^{-n_0}\sum_{m=0}^{N-1} x(m)z^{-m} = z^{-n_0}X(z).$$

3. $a^n x(n) \stackrel{z}{\longleftrightarrow} X(a^{-1}z)$ .

(Proof) $$\sum_{n=0}^{N-1} a^n x(n)z^{-n} = \sum_{n=0}^{N-1} x(n)\left[a^{-1}z\right]^{-n} = X(a^{-1}z).$$

4. $x(-n) \stackrel{z}{\longleftrightarrow} X(z^{-1})$

This is the $z$-transform of a reflected signal.

(Proof) $$\sum_{n=-(N-1)}^{0} x(-n)z^{-n} = \sum_{m=0}^{N-1} x(m)z^{-(-m)} = \sum_{m=0}^{N-1} x(m)\left[z^{-1}\right]^{-m} = X(z^{-1}).$$

5. $nx(n) \stackrel{z}{\longleftrightarrow} -z\dfrac{dX(z)}{dz}$ (Derivative of the transform)

(Proof) $$\frac{dX(z)}{dz} = \frac{d}{dz}\left(\sum_{n=0}^{N-1} x(n)z^{-n}\right) = -z^{-1}\sum_{n=0}^{N-1} nx(n)z^{-n}.$$

### 3.4 Discrete-Time System (Digital Filter)

Because discrete-time systems are usually implemented on a microprocessor to perform filtering operation, they are often called *digital filters*. A discrete-time system with the impulse response $h(n)$, the input $x(n)$, and the output $y(n)$ is shown in Figure 3.3.

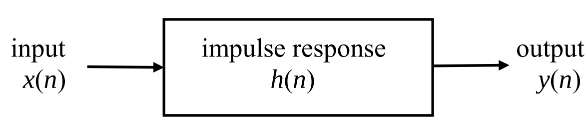



Figure 3.3 Discrete-Time System (Digital Filter)

Digital filters are classified by the length of the impulse response. If the length of the filter's impulse response is finite, then the filter is called the *finite impulse response* (*FIR*) *filter*. If the impulse response is infinitely long, then the filter is called the *infinite impulse response* (*IIR*) *filter*.

1. *FIR filter* (*MA system*; *all-zero system*)

Assume that the impulse response of a discrete-time system is given as follows.

$$h(n) = \begin{cases} b_n, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

Notice that the length of the impulse response is finite. This filter is called a *finite impulse response* (*FIR*) *filter*. If the input of the filter is $x(n)$, then the output of the filter is expressed in terms of the convolution sum:

$$y(n) = h(n) * x(n) = \sum_{m=0}^{M} h(m)x(n-m) = \sum_{m=0}^{M} b_m x(n-m). \tag{3.8}$$

If $b_m$ is constant, $1/(M+1)$ for example, then the output is the average of the $M+1$ input samples (a present input sample, $x(n)$, and $M$ past input samples, $x(n-1)$, $x(n-2)$, etc.). The output is the *average* value of the input samples that are *moving* according to $n$. Thus, the FIR filter is often called the *moving average* (*MA*) *system*. Eq. (3.8) is also called the *difference equation* that describes the input-output relationship of the system in the time domain. The $z$-transform of Eq. (3.8) becomes

$$Y(z) = \sum_{m=0}^{M} b_m z^{-m} X(z). \tag{3.9}$$

where $Y(z)$ and $X(z)$ are the $z$-transforms of $x(n)$ and $y(n)$, respectively.

Now let us define the transfer function of the filter:

$$\boxed{H(z) = \frac{Y(z)}{X(z)}} \quad \textbf{Transfer Function} \tag{3.10}$$

The transfer function $H(z)$ is a rational function in terms of the complex variable $z$ and is defined as the output transform over the input transform. Roots of the numerator polynomial are called the *zeros* of the system and roots of the denominator polynomial are called the *poles* of the system. In the case of FIR systems, the transfer function becomes

$$H(z) = \sum_{m=0}^{M} b_m z^{-m}. \tag{3.11}$$

Note that the same result is obtained by taking the *z-transform* of the impulse response specified by Eq. (3.7). In other words, the transfer function $H(z)$ is the $z$-transform of the impulse response $h(n)$. Note from Eqs. (3.8) and (3.9) that a convolution in the time domain becomes a multiplication in the frequency domain. Because there are $M$ zeros and no poles in the transfer function, this system is called an *all-zero system*.

2. *IIR filter* I (*AR* or *all-pole system*)

Suppose the transfer function of a certain digital filter is given as

$$H(z) = \frac{1}{1 - a_1 z^{-1}}. \tag{3.12}$$

The impulse response of the system is given by

$$h(n) = a_1^n u(n). \tag{3.13}$$

Note that the length of the impulse response is infinite. Thus, this filter is called an *infinite impulse response* (*IIR*) *filter*. The input-output relationship of the system is expressed as

$$(1 - a_1 z^{-1}) Y(z) = X(z). \tag{3.14}$$

In terms of the difference equation, the input-output relation is

$$y(n) = a_1 y(n-1) + x(n). \tag{3.15}$$

Note that the present output is related to the one-step previous output and the present input.

**Example 3.4.1**

Find the impulse response of the system whose input, $x(n)$, and output, $y(n)$, are described by the following difference equation:

$$y(n) = 0.8y(n-1) + 2x(n).$$

Assume that $h(n) = 0$ for $n < 0$.

<u>Solution</u>

• Method1: Because the impulse response is the response of the system to the unit impulse, input $x(n)$ needs to be replaced by $\delta(n)$ and output $y(n)$ needs to be replaced by $h(n)$ such that the difference equation becomes

$$h(n) = 0.8h(n-1) + 2\delta(n).$$

Now $h(n) = 0$ when $n < 0$.

$h(0) = 0.8h(-1) + 2\delta(0) = (0.8)(0) + (2)(1) = 2$

$h(1) = 0.8h(0) + 2\delta(1) = 2(0.8)$

$h(2) = 0.8h(1) + 2\delta(2) = 2(0.8^2)$, and so on. Note that $\delta(n) = 0$ when $n \neq 0$.

Thus, $h(n) = 2(0.8^n)u(n)$.

• Method 2: The *z-transform* of the difference equation is

$Y(z) = 0.8z^{-1}Y(z) + 2X(z)$ or $[1 - 0.8z^{-1}]Y(z) = 2X(z)$.

Thus, the transfer function becomes

$$H(z) = \frac{2}{1-0.8z^{-1}}.$$

The inverse *z-transform* of the above gives $h(n) = 2(0.8^n)u(n)$.

As can be seen in Eq. (3.15), the present output sample is determined by the previous output samples and the present input sample. Because the present output sample *itself* is related to the previous output samples recursively or *regressively*, this kind of system is also called the *self-regressive system* or *autoregressive* (*AR*) *system*. In general, an AR system has a transfer function:

$$H(z) = \frac{b_0}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_K z^{-K}} = \frac{b_0}{1 - \sum_{m=1}^{K} a_m z^{-m}}. \tag{3.16}$$

Because this has $K$ poles and no zeros (except for zeros at the origin), the system is called the *all-pole system*.

Now the difference equation is given by

$$y(n) = b_0 x(n) + \sum_{m=1}^{K} a_m y(n-m). \tag{3.17}$$

In the AR system, the present output sample is determined by the present input sample, and the previous output samples.

3. *IIR Filter* II (*ARMA or pole-zero system*)

An IIR system with both poles and zeros will have the transfer function:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_K z^{-K}} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{1 - \sum_{m=1}^{K} a_m z^{-m}}. \tag{3.18}$$

This system is called the *pole-zero system* or the *autoregressive moving average* (*ARMA*) *system*. In this case, the difference equation is

$$y(n) = \sum_{m=0}^{M} b_m x(n-m) + \sum_{m=1}^{K} a_m y(n-m). \tag{3.19}$$

The present output sample is related to the present input sample as well as the past input samples and the previous output samples.

4. *Causal or noncausal filter*

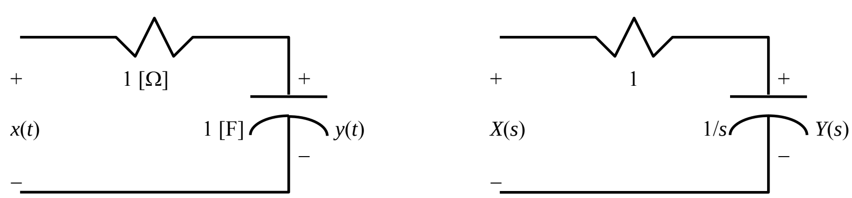
Finally, we would like to define one more term that is frequently used. In any linear time-invariant system, when there is an input there is an output. It is like there is cause and effect in the system. In the case of the impulse response the impulse is the cause and the impulse response is the effect. In reality, there is no effect when there is no cause. Because impulse $\delta(n)$ is 1 when $n = 0$ and 0 otherwise, the impulse response will be zero when $n < 0$ (No cause, no effect!) If $h(n) = 0$ for $n < 0$, then the system is *causal* or *realizable*. A system whose impulse response is nonzero for $n < 0$ is called the *noncausal* or *nonrealizable* system. This means that even when there is no cause, there is an effect.

## 3.5. Frequency Response of a Digital Filter

In this section, frequency response of a discrete-time system is explained after reviewing the frequency response of an analog filter. Consider the following example.

### Analog Filter Frequency Response

Find the frequency response of the following analog lowpass filter (LPF) shown in Fig. 3.4(a). Input and output voltages are denoted by $x(t)$ and $y(t)$ whose Laplace transforms are given by $X(s)$ and $Y(s)$, respectively, as shown in Fig. 3.4 (b)




(a) (b)

Figure 3.4 (a) LPF circuit (b) LPF in the Laplace transform domain.

Using the voltage division, one can easily show that the output $Y(s)$ is related to the input $X(s)$ by the following equation.

$$Y(s) = \frac{\frac{1}{s}}{1+\frac{1}{s}} X(s) .$$

The transfer function is given by

$$H(s) = \frac{Y(s)}{X(s)} = \frac{\frac{1}{s}}{1+\frac{1}{s}} = \frac{1}{s+1} .$$

The frequency response of the filter is the transfer function *H*(*s*) evaluated on the $j\omega$ axis such that

$$H(\omega) = H(s)\big|_{s=j\omega} = \frac{1}{1+j\omega} .$$

The magnitude response is given by

$$|H(\omega)| = \frac{1}{\sqrt{1+\omega^2}} .$$

The phase response is given by

$$\arg\{H(\omega)\} = \tan^{-1}\left(\frac{0}{1}\right) - \tan^{-1}\left(\frac{\omega}{1}\right) = -\tan^{-1}\omega$$

The plots of the magnitude response and the phase response are shown in Figure 3.5 (a). Note that both the horizontal axis and the vertical axis are in the linear scale in Fig 3.5(a).

In most applications, to plot the magnitude response of an analog filter, the vertical axis is in dB (decibel) and the horizontal axis is in the log scale as shown in Fig. 3.5 (b). We are going to review the decibel (dB) and Bode plots below.

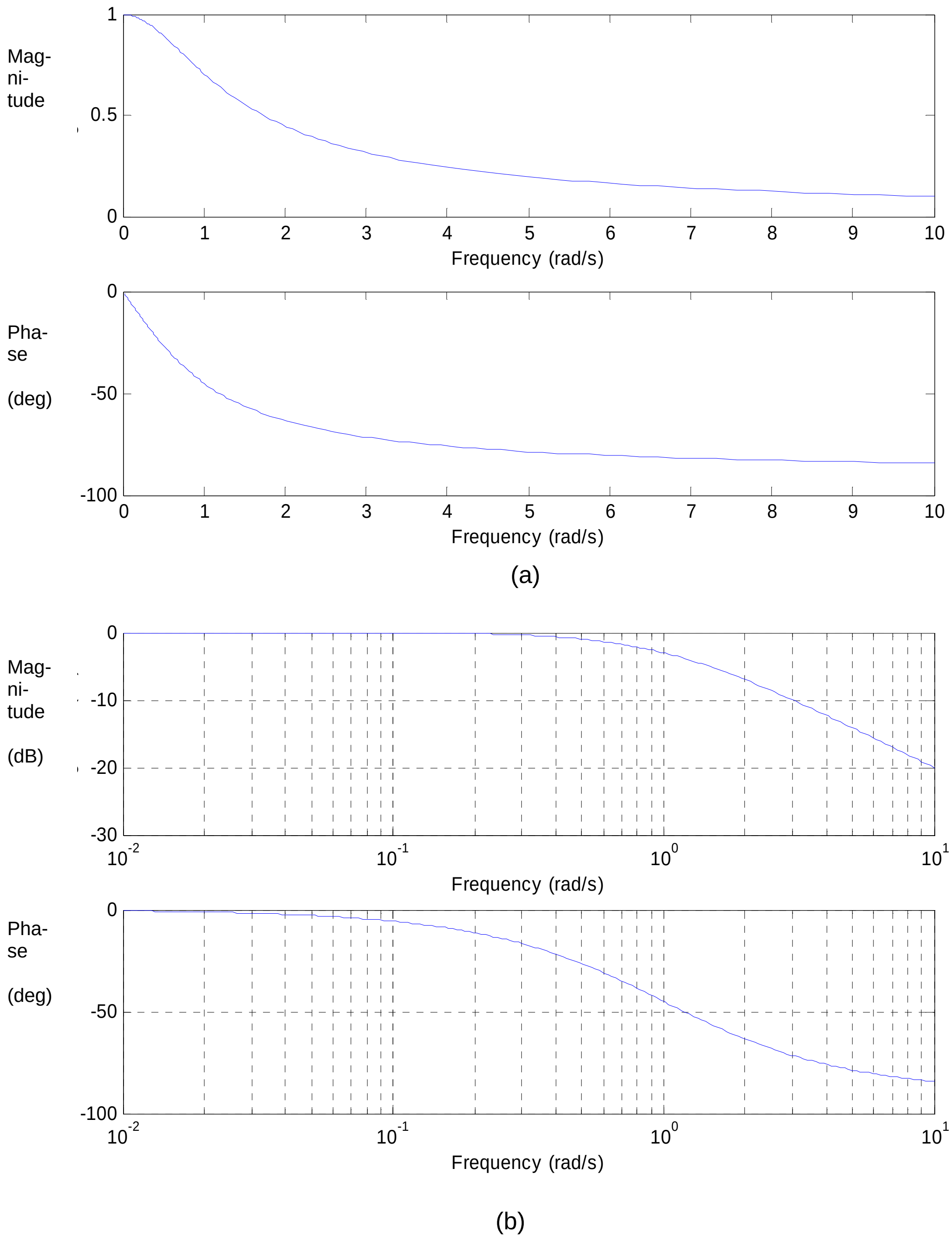



Fig. 3.5 (a) Magnitude and Phase plots - linear scale (b) Bode plots

**dB (decibel)**

The decibel (dB) is a logarithmic unit used to describe a ratio. The ratio may be power, sound pressure, voltage or current. For example, suppose we have an amplifier whose input power is $P_i$ and output power is $P_o$. The power gain in dB of the amplifier is defined to be

$$10 \log_{10}(P_o/P_i) \text{ [dB]}.$$

Without the multiplication factor, $\log_{10}(P_o/P_i)$ is called the Bel (to honor Alexander Bell who invented the telephone). If the output power is twice as much than the input power, the gain in dB is

$$10 \log_{10}(P_o/P_i) = 10 \log_{10}2 = 3 \text{ [dB]}.$$

If the output has a million times the power of the input, the gain in dB is

$$10 \log_{10}(P_o/P_i) = 10 \log_{10}(10^6) = 60 \text{ [dB]}.$$

So far we use the dB to describe the ratio of powers. What about the voltage or current ratio? Remember that power is a voltage squared divide by a resistance or a current squared multiplied by a resistance. Thus, the voltage ratio and the current ratio respectively are:

$$10 \log_{10}(V_o/V_i)^2 = 20 \log_{10}(V_o/V_i)$$

$$10 \log_{10}(I_o/I_i)^2 = 20 \log_{10}(I_o/I_i).$$

Because transfer function is the ratio of the output voltage (or current) to the input voltage (or current), the magnitude response in dB is given by

$$20 \log_{10} |H(\omega)|.$$

**Bode Plot**

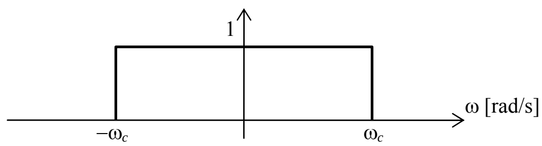
There are a number of ways to represent the frequency response graphically, but the method suggested by H. W. Bode in the 1930s is particularly useful. Bode's method consists of plotting two curves: the *magnitude response in dB* and *phase* as functions of the *log of frequency*. Usually the magnitude response in dB and the phase are plotted linearly along the vertical axis on graph paper that has several cycles of a log scale on the horizontal axis. Each cycle represents a factor of ten in frequency. This special paper is known as semilog graph paper as shown in Fig 3.5 (b).

At low frequencies ($\omega \ll 1$) the gain is flat and unity, or 0 dB. At high frequencies ($\omega \gg 1$) the gain rolls off inversely with frequency, decreasing by a factor of 2 (or 6 dB) for every frequency doubling. This results in a straight line on semilog graph paper with a slope of −6 dB per/octave. Alternatively, the slope can be expressed as −20 dB per/decade (a factor of 10 in frequency). The gain at the cutfoff frequency ($\omega = 1$) is

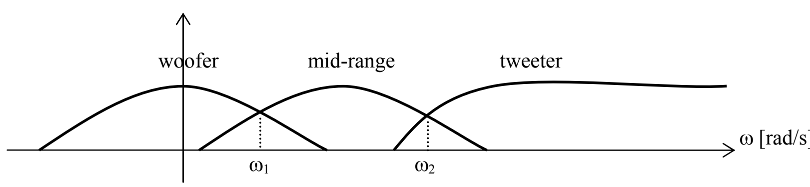
$1/\sqrt{2}$ = 0.707, or −3 dB. Looking at the phase plot, we see it is 0 degrees well below the cutoff frequency, −90 degrees well above, and −45 degrees at the cutoff frequency.

## Cutoff Frequency

In the case of an ideal lowpass filter as shown below, the cutoff frequency is clearly determined.




The cutoff frequency is $\omega_c$ as any signal whose frequency below $\omega_c$ will be passed and any signal whose frequency above $\omega_c$ will be completely stopped. However, in practical filters, there is no clear-cut cutoff frequency. Often, −3-dB cutoff frequency (or simply 3-dB cutoff frequency) is defined and used. For example, a three-way loud speaker system consists of three speakers: a woofer for low frequencies, a tweeter for high frequencies and a midrange for frequencies in between. The reason why we use three-way speakers is that it is hard to reproduce whole range of audio frequencies using a single speaker. The following plot shows the typical frequency response characteristics of the woofer, midrange and tweeter.




As shown in the figure, at $\omega_1$, the woofer delivers half (−3 dB) of the power and the midrange delivers another half (−3 dB) of the power so that the total power delivered by the two speakers is one (or 0 dB). At $\omega_2$, the same thing happens between the midrange and the tweeter. Now $\omega_1$ and $\omega_2$ are referred to as 3-dB cutoff frequencies. Don't forget that at the 3-dB cutoff frequency, the power is reduced to 0.5 and the magnitude is reduced to 0.707.

## Digital Filter Frequency Response

As we have seen in the previous example, a transfer function $H(s)$ evaluated on the $j\omega$ axis is the frequency response of a continuous-time system. We also know that the $j\omega$ axis in the $s$-plane is mapped onto $e^{j\theta}$ (a unit circle) in the $z$-plane. Thus, the frequency response of a discrete-time system can be obtained by evaluating the transfer function, $H(z)$, on the unit circle. The frequency response of a digital filter is obtained by

$$H(\theta) = H(z)\big|_{z=e^{j\theta}} = \frac{\sum_{m=0}^{M} b_m e^{-j\theta m}}{1 - \sum_{m=1}^{K} a_m e^{-j\theta m}} . \tag{3.20}$$

To plot $H(\theta)$, one may evaluate this at $N$ points equally spaced in angle. The number of points $N$ can be arbitrarily chosen. Study the following two examples.

**Example 3.5.1**

Find the frequency response of a system whose transfer function is specified by

$$H(z) = 0.25 + 0.5z^{-1} + 0.25z^{-2} .$$

Plot the magnitude response and determine what kind of filter this is: lowpass or highpass?

Solution

The frequency response of the system is obtained by evaluating the transfer function on the unit circle $e^{j\theta}$:

$$\begin{aligned} H(\theta) &= 0.25 + 0.5e^{-j\theta} + 0.25e^{-j2\theta} \\ &= 0.5e^{-j\theta} + 0.25e^{-j\theta}(e^{j\theta} + e^{-j\theta}) \\ &= 0.5e^{-j\theta}(1 + \cos\theta) \end{aligned}$$

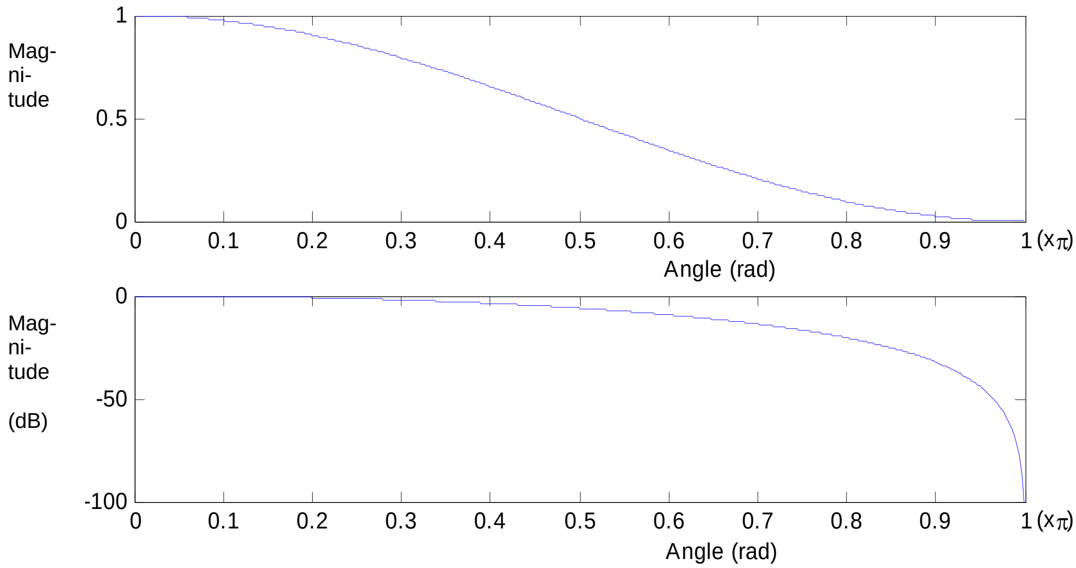
The magnitude response

$$|H(\theta)| = 0.5(1 + \cos\theta) .$$

The phase response is

$$\arg\{H(\theta)\} = -\theta .$$

The magnitude response plots in linear scale and in dB are shown below.




The second plot is the magnitude response in dB. Note that the horizontal axis is in the linear scale. The log scale is never used for the angle to plot the magnitude response of a digital filter. From the plots, it can be observed that the filter passes the low frequency but blocks the high frequency. (Note that the angle $\pi$ corresponds to the highest frequency of the signal.) Thus, the filter is a lowpass filter.

**Example 3.5.3**

Find the frequency response of a system whose transfer function is specified by

$$H(z) = \frac{1}{1 - az^{-1}}.$$

<u>Solution</u>

$$H(\theta) = \frac{1}{1 - ae^{-j\theta}} = \frac{1}{1 - a\cos\theta + ja\sin\theta}$$

$$\left|H(\theta)\right| = \frac{1}{\sqrt{(1 - a\cos\theta)^2 + (a\sin\theta)^2}}$$

$$= \frac{1}{\sqrt{1 - 2a\cos\theta + a^2}}$$

$$\arg\{H(\theta)\} = -\tan^{-1}\left\{\frac{a\sin\theta}{1 - a\cos\theta}\right\}.$$

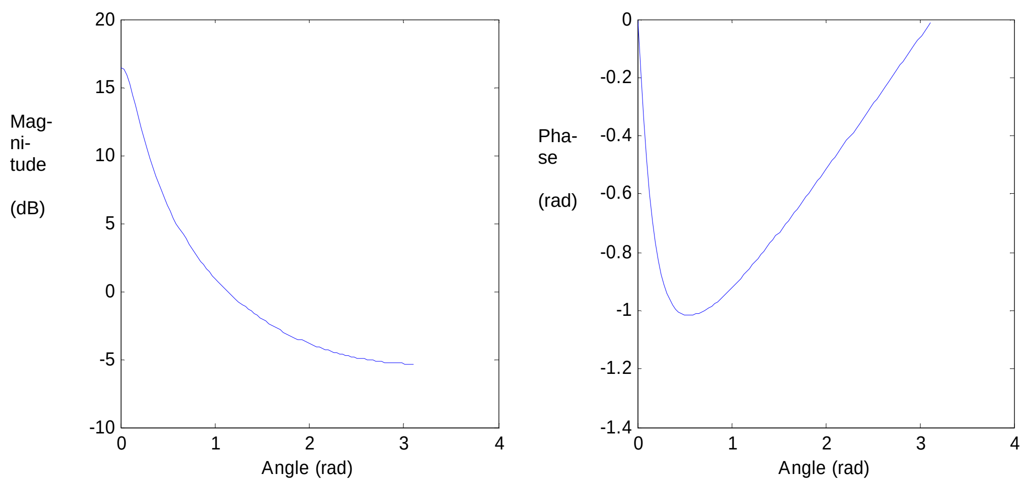
**MATLAB Example**

Find the frequency response of a system whose transfer function is specified by

$$H(z) = \frac{1}{1-0.85z^{-1}}.$$

(Suggestion)
Use MATLAB command freqz. Type `help freqz` for usage of the command. Type the following commands.

```
b=1;  % Numerator polynomial coefficients
a=[1 -0.85];      % Denominator polynomial coefficients
[H,w]=freqz(b,a,100);
subplot(121), plot(w,20*log10(abs(H))) % Magnitude in dB plot
ylabel('Magnitude (dB)')
xlabel('Angle (rad)')
subplot(122), plot(w,angle(H))      % Phase in radian plot
ylabel('Phase (rad)')
xlabel('Angle (rad)')
```




## 3.6. Computation of Frequency Response

A practical way of computing the frequency response of the system is to evaluate $H(\theta)$ in Eq. (3.20) at $N$ points equally spaced in angle between 0 and $2\pi$ such that

$$H(k) = H(\theta)\Big|_{\theta=\frac{2\pi}{N}k} = \frac{\sum_{m=0}^{M} b_m e^{-j\frac{2\pi}{N}km}}{1-\sum_{m=1}^{K} a_m e^{-j\frac{2\pi}{N}km}} \quad \text{for } k = 0, 1, 2, \cdots, N-1. \tag{3.21}$$

The alternative method can be devised by rewriting Eq. (3.21) as

$$H(k)=\frac{\sum_{m=0}^{M} b_m e^{-j\frac{2\pi}{N}km}+\sum_{m=M+1}^{N-1} 0\cdot e^{-j\frac{2\pi}{N}km}}{1-\sum_{m=1}^{K} a_m e^{-j\frac{2\pi}{N}km}+\sum_{m=K+1}^{N-1} 0\cdot e^{-j\frac{2\pi}{N}km}} \quad \text{for } k = 0,\ 1,\ 2,\ \cdots,\ N-1. \tag{3.22}$$

The numerator in Eq. (3.22) is the DFT of the $N$-point sequence (by padding $N-M-1$ zeros at the end)

$$\{b_0, b_1, b_2, \cdots, b_M, 0, \cdots, 0\} \tag{3.23}$$

and the denominator is the DFT of the $N$-point sequence (by padding $N-K-1$ zeros at the end)

$$\{1, -a_1, -a_2, \cdots, -a_K, 0, \cdots, 0\}\ . \tag{3.24}$$

Now the frequency response of the system can be computed by the following. First, the DFT $B(k)$ of the $N$-point sequence specified by Eq. (3.23) is computed. Second, the DFT $A(k)$ of the $N$-point sequence specified by Eq. (3.24) is computed. Finally, $H(k)$ is computed by

$$H(k)=\frac{B(k)}{A(k)} \quad \text{for } k = 0,\ 1,\ 2,\ \cdots\cdots,\ N-1. \tag{3.25}$$

A fast way to compute the DFT known as the fast Fourier transform (FFT) is going to be introduced in the next chapter. The FFT is used to compute $B(k)$ and $A(k)$ in practice.

Another way of computing a frequency response of the discrete-time system is to compute the DFT of the impulse response $h$(n).

$$H(k)=\sum_{n=0}^{N-1} h(n) e^{-j\frac{2\pi}{N}nk} \tag{3.26}$$

In the case of an IIR system, $h(n)$ has to be truncated. This truncation may cause some error. However, as long as the truncated portion is negligible, this method will provide for an adequate frequency response.

### 3.7 Linear Phase and FIR system

What kind of system is usually chosen in practice? FIR or IIR? One advantage of using FIR filters over IIR filters is that with FIR filters a linear phase can be obtained. The linear phase in the frequency domain implies a constant time delay in the time domain. The following two examples - one about the linear phase and the other about the nonlinear phase - will illustrate the difference between the linear phase and the nonlinear phase.

Linear Phase Example

An LTI system has the frequency response given by

$$H(\theta) = |H(\theta)|e^{-j\alpha\theta}$$

where $\alpha$ is a constant. The phase $\lambda$ is given by

$$\lambda(\theta) = -\alpha\theta.$$

Note that the phase is a line with the slope, $-\alpha$. It is referred to as the linear phase. Assume that the input of the system is given by

$$x(n) = A\,e^{j\theta_0 n}.$$

Let us find the output of the system. The output is obtained by convolving the input with the impulse response of the system.

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} h(k) A e^{j\theta_0 (n-k)} \\ &= A e^{j\theta_0 n} \sum_{k=-\infty}^{\infty} h(k) e^{-jk\theta_0} \\ &= A e^{j\theta_0 n} H(\theta_0) \end{aligned}$$

The first term is the input and the second term is the frequency response of the system evaluated at $\theta = \theta_0$. Thus, the amplitude of the output is the input amplitude multiplied by the magnitude response at $\theta_0$ ($|H(\theta_0)|$ ). The phase of the output is changed according to the phase response at $\theta_0$. Thus, the output is expressed as (because $H(\theta) = |H(\theta)|e^{-j\alpha\theta}$)

$$\begin{aligned} y(n) &= A|H(\theta_0)|\, e^{j\theta_0 n} e^{-j\alpha\theta_0} = A\left|H(\theta_0)\right| e^{j(\theta_0 n - \alpha\theta_0)} \\ &= A\left|H(\theta_0)\right| e^{j\theta_0 (n-\alpha)}. \end{aligned}$$

This holds for any frequency $\theta_0$. The linear phase ($-\alpha\theta$) in the frequency domain results in the constant delay ($\alpha$) in the time domain.

If an input is given by

$$x(n) = A\cos(\theta_0 n),$$

then the output will be

$$y(n) = A|H(\theta_0)|\cos(\theta_0 n - \theta_0\alpha) = A|H(\theta_0)|\cos\theta_0(n - \alpha).$$

There is the constant delay of $\alpha$ for any frequency $\theta_0$. This means that all frequency components of an input signal will have the same delay.

Nonlinear Phase Example

Let us repeat the above example with the frequency response

$$H(\theta) = |H(\theta)|\, e^{-j\alpha\theta^2}.$$

Note that the phase $(-\alpha\theta^2)$ is nonlinear. It is quadratic. The output becomes

$$\begin{aligned} y(n) &= A|H(\theta_0)|\, e^{j\theta_0 n} e^{-j\alpha\theta_0^2} = A\left|H(\theta_0)\right| e^{j(\theta_0 n - \alpha\theta_0^2)} \\ &= A\left|H(\theta_0)\right| e^{j\theta_0(n-\alpha\theta_0)}. \end{aligned}$$

In this case, the delay is $\alpha\theta_0$ that depends on the frequency $\theta_0$. In other words, different frequency components of an input signal would have different delays. This results in a distortion that is referred to as the *phase distortion*.

From the two examples above, one can conclude that it is very important to maintain a linear phase. By making coefficients of an FIR filter symmetric or anti-symmetric, a linear phase is guaranteed. Now let us find the phase of the following two symmetric and anti-symmetric FIR filters.

**Example 3.7.1**

The impulse response of an FIR filter is 5-point long and is symmetric (or even symmetric) about the midpoint, i.e., $h(0) = h(4)$, $h(1) = h(3)$. Find an expression of the frequency response.

Solution

The frequency response of the filter is

$$\begin{aligned} H(\theta) &= h(0) + h(1)e^{-j\theta} + h(2)e^{-j2\theta} + h(3)e^{-j3\theta} + h(4)e^{-j4\theta} \\ &= e^{-j2\theta}\left[\left\{h(0)e^{j2\theta} + h(4)e^{-j2\theta}\right\} + \left\{h(1)e^{j\theta} + h(3)e^{-j\theta}\right\} + h(2)\right] \\ &= e^{-j2\theta}\left[h(0)\left\{e^{j2\theta} + e^{-j2\theta}\right\} + h(1)\left\{e^{j\theta} + e^{-j\theta}\right\} + h(2)\right] \\ &= e^{-j2\theta}\left[2h(0)\cos 2\theta + 2h(1)\cos\theta + h(2)\right] \\ &= \pm|H(\theta)|\; e^{j\lambda(\theta)} \end{aligned}$$

where

$$\pm|H(\omega)| = 2\sum_{n=1}^{2} h(2-n)\cos(n\omega) + h(2)$$

and

$$\lambda(\theta) = -2\theta.$$

This frequency response has a linear phase that will give a constant delay.

**Example 3.7.2**

The impulse response of an FIR filter is 6-point long and anti-symmetric (or odd symmetric), i.e. $h(0) = -h(5)$, $h(1) = -h(4)$, and $h(2) = -h(3)$. Find an expression of the frequency response.

Solution

The frequency response of the filter is

$$H(\theta) = h(0) + h(1)e^{-j\theta} + h(2)e^{-j2\theta} + h(3)e^{-j3\theta} + h(4)e^{-j4\theta} + h(5)e^{-j5\theta}$$

$$= e^{-j2.5\theta}\left[h(0)\left(e^{j2.5\theta} - e^{-j2.5\theta}\right) + h(1)\left(e^{j1.5\theta} - e^{-j1.5\theta}\right) + h(2)\left(e^{j0.5\theta} - e^{-j0.5\theta}\right)\right]$$

$$= e^{-j2.5\theta}\left[j2h(0)\sin 2.5\theta + j2h(1)\sin 1.5\theta + j2h(2)\sin 0.5\theta\right]$$

$$= e^{j\frac{\pi}{2}}e^{-j2.5\theta}\left[2h(0)\sin 2.5\theta + 2h(1)\sin 1.5\theta + 2h(2)\sin 0.5\theta\right]$$

$$= \pm|H(\theta)|\, e^{j\lambda(\theta)}$$

where

$$\pm|H(\theta)| = 2\sum_{n=0}^{2} h(2-n)\sin\left((n+.5)\theta\right)$$

and

$$\lambda(\theta) = \pi/2 - 2.5\theta.$$

This frequency response has a linear phase and it will give a constant delay.

From the examples above, one can conclude that an FIR filter with symmetric or anti-symmetric impulse response produces a linear phase in the frequency domain and that results in a constant delay in the time domain. In general, $N$-point symmetric filter will have the following frequency response when $N$ is *even*.

$$H(\theta) = 2e^{-j\left(\frac{N-1}{2}\right)\theta}\sum_{n=0}^{\frac{N}{2}-1} h\left(\frac{N}{2}-1-n\right)\cos\left((n+.5)\theta\right).$$

Note that the frequency response is zero at $\theta = \pi$. As $\pi$ is the highest frequency in the digital frequency domain, a symmetric FIR filter with even number of coefficients cannot be a highpass or bandstop filter. This will be explained again in the next section.

## 3.8 FIR Lowpass Filter Design Using Windows

In this section, we will design linear-phase FIR lowpass filters using windows. Through a window in a building one can see a part of the whole view. Windows in this section are time windows that will select some particular points of a long sequence by truncating unwanted points. This way one can obtain a finite-length sequence from an infinitely long sequence. To design an FIR lowpass filter, we start from an ideal lowpass filter. A continuous-time ideal lowpass filter (analog filter) with the cutoff frequency of $\omega_c$ has the following frequency response (See Fig. 3.6).

$$H_a(\omega) = \begin{cases} 1, & -\omega_c \le \omega \le \omega_c \\ 0, & \text{elsewhere} \end{cases} \tag{3.27}$$

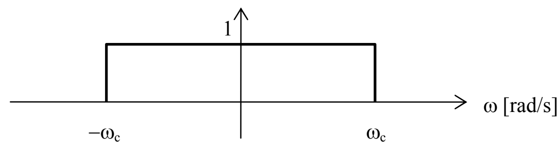



Fig. 3.6 Ideal lowpass filter frequency response

The impulse response is obtained by taking the inverse Fourier transform.

$$h_a(t) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega t} d\omega = \frac{1}{2\pi}\left(\frac{1}{jt}\right)\left[e^{j\omega_c t} - e^{-j\omega_c t}\right] = \left(\frac{\omega_c}{\pi}\right)\frac{\sin\omega_c t}{\omega_c t}. \tag{3.28}$$

The continuous-time impulse response is infinitely long as one shown in Fig. 3.7.

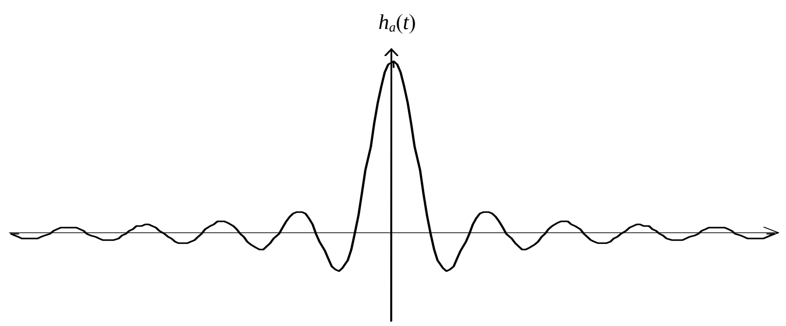



Figure 3.7 Impulse Response of the ideal lowpass filter

The ideal lowpass filter is not realizable as the impulse response is not causal. There are three necessary steps to design a digital FIR filter from the continuous-time impulse response: i) sampling, ii) shifting and iii) normalization. Let us choose the sampling interval $T = 1$[s] so that the cutoff frequency in digital filter $\theta_c = \omega_c T = \omega_c$.

There are two possibilities to sample the analog filter impulse response. First, $h_a(t)$ is sampled every second at $t = 0, \pm 1, \pm 2, \cdots$. (Do not forget that it is important to maintain symmetry to have a linear phase!) This kind of sampling will produce a filter with odd number of points. Second, $h_a(t)$ can be sampled every second at $t = \pm 0.5, \pm 1.5, \pm 2.5, \cdots$.

This will result in a filter with even number of points. In either case, the sampled impulse response has to be shifted to the right so that $h(n) = 0$ when $n < 0$. Finally, the filter coefficients should be normalized so that $|H(0)| = 1$ for lowpass filters and $|H(\pi)| = 1$ for highpass filters.

**Example 3.8.1**

Design a 7-point FIR lowpass filter with the cutoff frequency $\theta_c$ of $0.2\pi$.

Solution

$h_a(t) = (0.2)\dfrac{\sin 0.2\pi t}{0.2\pi t}$ from Eq. (3.28).

Now $h_a(t)$ is sampled at $t = 0, \pm 1, \pm 2, \pm 3$ so that

$h_a(-3) = 0.1009,\ h_a(-2) = 0.1514,\ h_a(-1) = 0.1871,\ h_a(0) = 0.2000,$
$h_a(1) = 0.1871,\quad h_a(2) = 0.1514,\quad h_a(3) = 0.1009.$

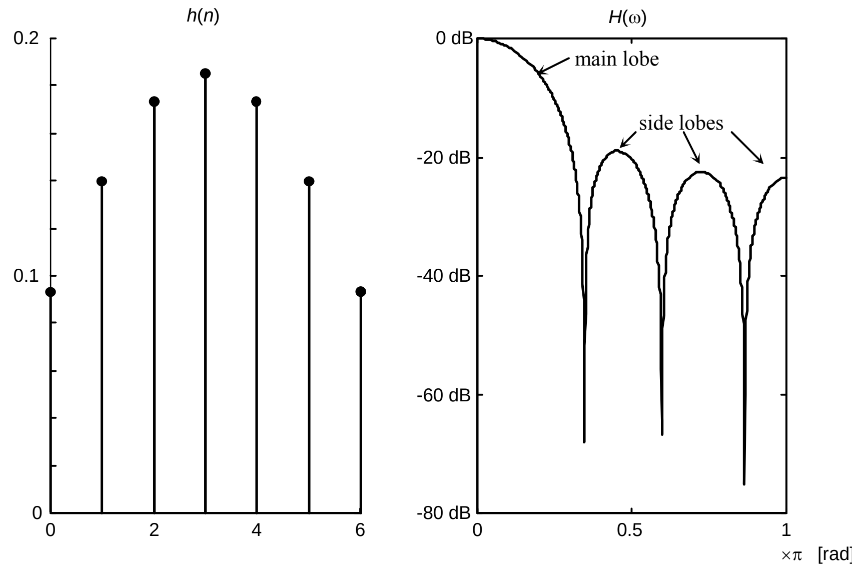
To make a causal digital filter, $h_a(n)$ should be shifted by 3. Finally, a lowpass filter should have unit gain at $\theta = 0$, i.e., $H(0) = \sum_{n=0}^{6} h(n)e^{-j(0)n} = \sum_{n=0}^{6} h(n) = 1$. Because $\sum_{n=-3}^{3} h_a(n) = 1.0787$, the final digital filter is obtained by shifting $h_a(n)$ by 3 to the right and by dividing each coefficient by 1.0787.

$h(0) = 0.0935,\ h(1) = 0.1403,\ h(2) = 0.1734,\ h(3) = 0.1854,$
$h(4) = 0.1734,\quad h(5) = 0.1403,\ h(6) = 0.0935.$

The impulse response and the frequency magnitude response are plotted in Fig. 3.8. Note that the magnitude plot is in dB (decibel). The magnitude response in decibel is given as

$$20\log_{10}|H(\theta)| \text{ [dB]}.$$

The filter gain at d.c. is 1 that corresponds to 0 [dB]. The pass band and the transition band are included in the main lobe, and the stop band consists of the side lobes. The filter gain at the peak of the first side lobe is about −20 [dB]. That is equivalent to say that the attenuation at the first side lobe is about 20 [dB]. This indicates that a signal that needs to be stopped or rejected still have one tenth of the original amplitude. In most applications, having only 20 dB of attenuation is not enough. Unfortunately this problem does not go away even when the order of the LPF is increased. In fact, as the filter order increases, the transition band gets narrower but attenuation in the stop band will not improve.




(a) Impulse response (b) Magnitude response

Figure 3.8 The impulse response and the magnitude response of the LPF designed using a rectangular window.

**Exercise 3.8.1**

Design an 8-point FIR lowpass filter with the cutoff frequency $\theta_c = \pi/2$.

Answer

$h(0) = -0.0721$ $h(1) = -0.1010$ $h(2) = 0.1683$ $h(3) = 0.5048$
$h(4) = 0.5048$ $h(5) = 0.1683$ $h(6) = -0.1010$ $h(7) = -0.0721$

So far we have used a *rectangular window*. The *N*-point rectangular window is defined as

$$w_R(n) = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \text{elsewhere.} \end{cases} \tag{3.29}$$

The window sequence specified by Eq. (3.29) is also referred to as the *boxcar* window. The window is used to truncate a long sequence (often, a infinitely long sequence) to make a finite-length sequence. The following procedure is followed to design an *N*-point FIR lowpass filter using a window.

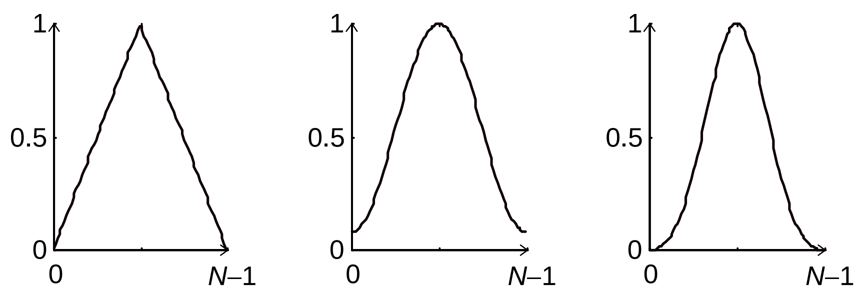
1) Obtain the expression of $h_a(t)$ for the desired normalized frequency $\theta_c$ (= $\omega_c$).
2) Sample $h_a(t)$ at every second at $t = 0, \pm1, \pm2, \cdots$ for the filter with odd number of coefficients or at $t = \pm0.5, \pm1.5, \pm2.5, \cdots$ for the filter with even number of coefficients.
3) Sampled version of $h_a(t)$ is shifted by $(N-1)/2$ to the right..

4) Each of the resulting filter coefficients is multiplied by the corresponding window coefficient.
5) Filter coefficients are normalized so that $|H(\theta)|_{\theta=0} = 1$.

Problem with the rectangular window is that attenuation at the stop band is not large enough. As shown in Fig. 3.8, the attenuation achieved at the first side lobe is about 20 dB. To increase the attenuation at the stop band other types of windows are used. For example, instead of using the rectangular window, the triangular window (or known as Bartlett widow) as shown in Fig. 3.9(a) may be used.

- Triangular (Bartlett) window (for odd $N$):

$$W_T(n) = \begin{cases} \dfrac{2n}{N-1}, & 0 \le n \le \dfrac{N-1}{2} \\ 2 - \dfrac{2n}{N-1}, & \dfrac{N-1}{2} < n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$




(a) Triangular window (b) Hamming window (c) Blackman window

Figure 3.9 Three different types of windows

Two more windows are shown in Fig. 3.9 (b) and (c), respectively.

- Hamming window:

$$w_H(n) = \begin{cases} 0.54 - 0.46\cos[2\pi n/(N-1)], & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.30}$$

- Blackman window:

$$w_B(n) = \begin{cases} 0.42 - 0.5\cos[2\pi n/(N-1)] + 0.08\cos[4\pi n/(N-1)], & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.31}$$

Note that the actual length of the Bartlett and the Blackman windows is $N-2$ as the first and the last window points are zero.

**Example 3.8.2**

Design a 7-point FIR lowpass filter with the cutoff frequency $\theta_c$ of $0.2\pi$ using the triangular window.

Solution

We can use the same coefficients obtained in Example 3.8.1. Let us shift the sequence by 3.

$h_a(0) = 0.1009$, $h_a(1) = 0.1514$, $h_a(2) = 0.1871$, $h_a(3) = 0.2000$,
$h_a(4) = 0.1871$, $h_a(5) = 0.1514$, $h_a(6) = 0.1009$.

Each filter coefficient must be multiplied by each of the triangular window coefficients. The triangular window coefficients are given as follows. ($N = 9$ is used but the first and the last window points are neglected as they are 0.)

$w_T(0) = 0.25$, $w_T(1) = 0.5$, $w_T(2) = 0.75$, $w_T(3) = 1.0$,
$w_T(4) = 0.75$, $w_T(5) = 0.5$, $w_T(6) = 0.25$.

Now intermediate filter coefficients are obtained by multiplying corresponding window coefficients.

$$h_i(0) = h_a(0) \times w_T(0) = 0.1009 \times 0.25 = 0.0252$$
$$h_i(1) = h_a(1) \times w_T(1) = 0.1514 \times 0.5 = 0.0757$$
$$h_i(2) = h_a(2) \times w_T(2) = 0.1871 \times 0.75 = 0.1403$$
$$h_i(3) = h_a(3) \times w_T(3) = 0.2000 \times 1.0 = 0.2000$$
$$h_i(4) = 0.1403, \quad h_i(5) = 0.0757, \quad h_i(6) = 0.0252.$$

A lowpass filter should satisfy the condition: $H(0) = \sum_{n=0}^{6} h(n)e^{-j(0)n} = \sum_{n=0}^{6} h(n) = 1$.

Because $\sum_{n=0}^{6} h_i(n) = 0.6824$, the final digital filter is obtained by dividing $h_i(n)$ by 0.6824.

$h(0) = 0.0369$, $h(1) = 0.1109$, $h(2) = 0.2056$, $h(3) = 0.2931$,
$h(4) = 0.2056$, $h(5) = 0.1109$, $h(6) = 0.0369$.

**Exercise 3.8.2**

Design a 7-point FIR lowpass filter with the cutoff frequency of $0.2\pi$ using the Hamming window.

Answer

$h(n) = \{0.0135, 0.0785, 0.2409, 0.3344, 0.2409, 0.0785, 0.0135\}$

Magnitude response of the three lowpass filters we have just designed using the three different windows are plotted on the same graph as shown in Fig. 3.10.

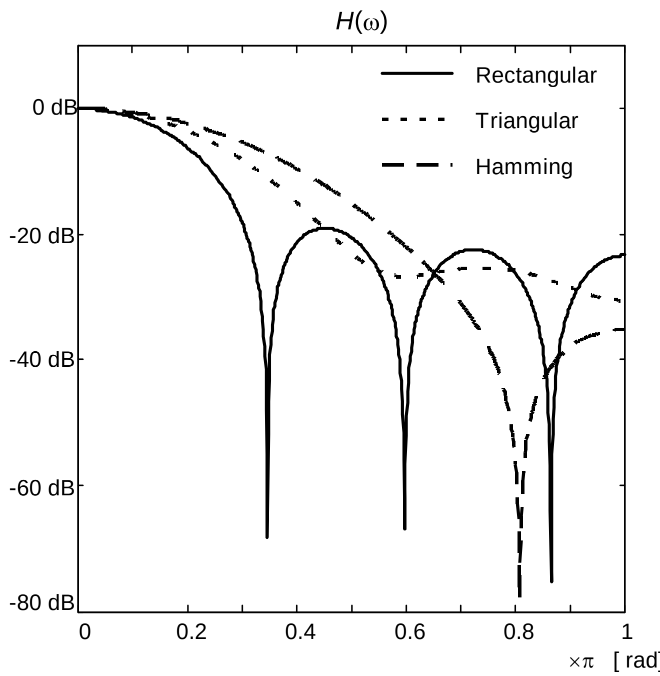



Fig. 3.10 Magnitude response of LPF designed using different windows

There is a tradeoff between the transition band and the attenuation. The rectangular window gives the sharpest transition characteristic and the least attenuation at the stop band. On the other hand, the triangular window gives larger attenuation at the expense of wider transition band. The Hamming window gives a larger attenuation at the stop band and a wider transition band than the triangular window.

Fig. 3.11 shows the frequency characteristics of 20-point long lowpass filters (with the cutoff frequency of $0.2\pi$) designed using three different types of windows. Of the three filters, one designed using the rectangular window of course gives the sharpest transition characteristic, but it has the least attenuation at the stop band. On the other hand, the Blackman window gives about 80 dB of attenuation at the expense of very wide transition band. The filter designed using the Hamming window gives about 50 dB of attenuation at the stop band and has relatively a sharper transition band than the Blackman window. Transition band can be reduced by increasing the number of filter coefficients. However, attenuation cannot be improved by just increasing the number of filter coefficients. Readers may want to try filters with different orders, 40 or 50 for example. Attenuation depends on a type of window that is selected. If a large attenuation in the stop band required such as in digital audio, one may want to choose the Blackman window. For speech processing, Hamming windows are often used. For more types of windows see Ifeacher.[1]

[1] E.C. Ifeacher et al., Digital Signal Processing - A Practical Approach, Addison-Wesley, 1995.

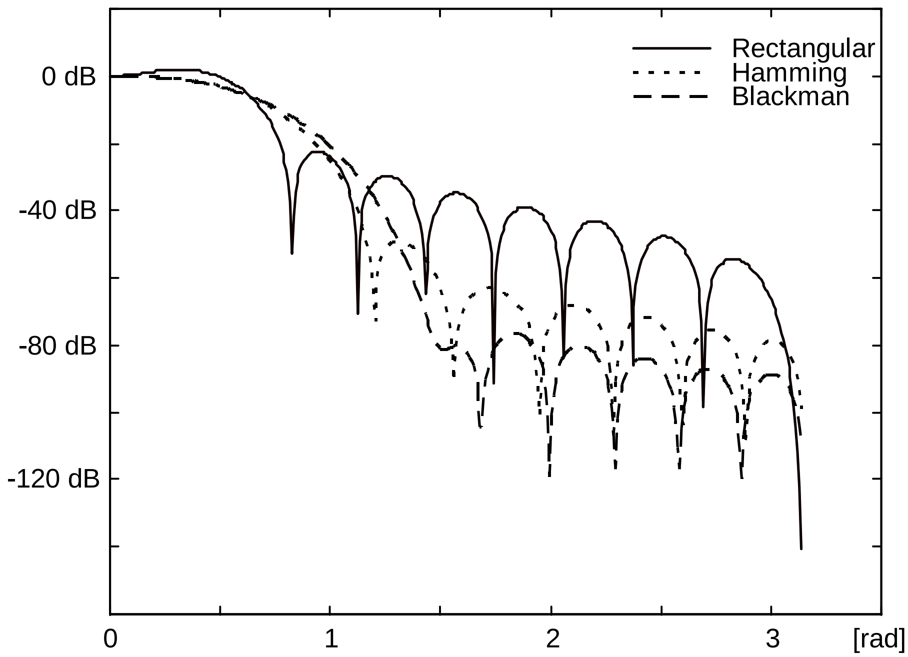



Figure 3.11 Lowpass filter magnitude response (20-point long filters)

### 3.9 FIR Highpass Filter Design

An ideal lowpass filter with the cutoff frequency $\omega_c$ (= $\theta_c$) has the frequency response and the impulse response, respectively.

$$H_l(\omega) = \begin{cases} 1, & -\theta_c \le \omega \le \theta_c \\ 0 & \text{otherwise} \end{cases}$$

$$h_l(t) = \left(\frac{\theta_c}{\pi}\right)\frac{\sin\theta_c t}{\theta_c t}$$

An ideal highpass filter with the cutoff frequency $\omega_c$ (= $\theta_c$) has the frequency response that is expressed in terms of $H_l(\omega)$. (See Fig. 3.12.)

$$H_h(\omega) = 1 - H_l(\omega) \tag{3.32}$$

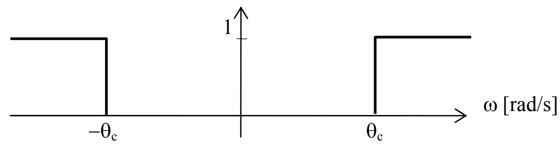



Fig 3.12 Ideal highpass filter frequency response

The impulse response is obtained by taking the inverse Fourier transform of $H_h(\omega)$ so that

$$h_h(t) = \delta(t) - \left(\frac{\theta_c}{\pi}\right)\frac{\sin\theta_c t}{\theta_c t}. \tag{3.33}$$

The same procedure to design FIR lowpass filters using windows can be followed for the highpass filter design. However, one important property of highpass filters needs to be pointed out here. As shown at the end of Section 3.7, a general expression of the frequency response of even-symmetric FIR filters with even number of coefficients is

$$H(\theta) = \pm|H(\theta)|\, e^{j\lambda(\theta)}$$

Where

$$\pm|H(\theta)| = 2\sum_{n=0}^{\frac{N}{2}-1} h(\frac{N}{2}-1-n)\cos\big((n+.5)\theta\big)$$

and

$$\lambda(\theta) = -\frac{N-1}{2}\theta.$$

In this case, the frequency response is zero at $\theta = \pi$ which is the highest frequency. Therefore, an even-symmetric FIR filter with even number of coefficients cannot be a highpass filter.

On the other hand, frequency response of an even-symmetric FIR filter with odd number of coefficients is expressed as

$$H(\theta) = \pm|H(\theta)|\, e^{j\lambda(\theta)}$$

where

$$\pm|H(\theta)| = 2\sum_{n=1}^{\frac{N-1}{2}} h(\tfrac{N-1}{2} - n)\cos(n\theta) + h(\tfrac{N-1}{2})$$

and

$$\lambda(\theta) = -\frac{N-1}{2}\theta.$$

In this case, the frequency response is not zero at $\theta = \pi$. Thus, FIR highpass filters should have odd number of coefficients when they are even symmetric.

**Example 3.9.1**

Design a 7-point FIR highpass filter with the cutoff frequency of $0.2\pi$ using the Hamming window.

Solution

Using Eq. (3.33), $h_h(t)$ sampled at $t = 0, \pm1, \pm2, \pm3$ are

$h_h(-3) = -0.1009$, $h_h(-2) = -0.1514$, $h_h(-1) = -0.1871$, $h_h(0) = 1 - 0.2 = 0.8000$, $h_h(1) = -0.1871$, $h_h(2) = -0.1514$, $h_h(3) = -0.1009$.

Each filter coefficient must be multiplied by the corresponding window coefficient. The Hamming window coefficients are given as follows.

$w_H(0) = 0.08$, $w_H(1) = 0.31$, $w_H(2) = 0.77$, $w_H(3) = 1.0$,
$w_H(4) = 0.77$, $w_H(5) = 0.31$, $w_T(6) = 0.08$.

Now intermediate filter coefficients are obtained by multiplying window coefficients.

$h_i(0) = h_a(0) \times w_H(0) = -0.1009 \times 0.08 = -0.0081$

$h_i(1) = h_a(1) \times w_T(1) = -0.1514 \times 0.31 = -0.0469$

$h_i(2) = h_a(2) \times w_T(2) = -0.1871 \times 0.77 = -0.1441$

$h_i(3) = h_a(3) \times w_T(3) = 0.8000 \times 1.0 = 0.8000$

$h_i(4) = -0.1441$, $h_i(5) = -0.0469$, $h_i(6) = -0.0081$.

A highpass filter should satisfy $H(\pi) = \sum_{n=0}^{6} h(n)e^{-j(\pi)n} = \sum_{n=0}^{6} h(n)(-1)^n = 1$. Because $\sum_{n=0}^{6} h_i(n)(-1)^n = -1.0106$, the final digital filter is obtained by dividing $h_i(n)$ by $-1.0106$.

$h(0) = 0.0080$, $h(1) = 0.0464$, $h(2) = 0.1426$, $h(3) = -0.7917$,
$h(4) = 0.1426$, $h(5) = 0.0464$, $h(6) = 0.1426$.

### 3.10 Half-band LPF and HPF

Let us assume that $H_0(\theta)$ is a half-band lowpass filter. Half-band lowpass means that the cutoff frequency of this filter is $\pi/2$ which is a half of the highest frequency, $\pi$. Now the highpass filter $H_1(\theta)$ is the mirror image of the lowpass filter if there is a mirror at $\theta = \pi/2$. A pair of half-band lowpass and highpass filters are depicted in Fig. 3.13.

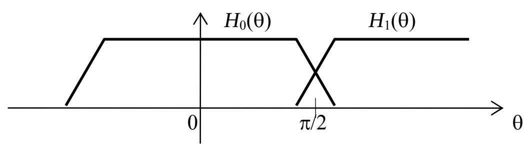



Figure 3.13 Half-band LPF and HPF

As $H_1(\theta)$ is obtained by shifting $H_0(\theta)$ by $\pi$, $H_1(\theta)$ can be expressed as

$$H_1(\theta) = H_0(\theta - \pi). \tag{3.34}$$

Now $H_0(\theta-\pi)$ and $H_1(\theta)$ may be expressed, respectively, as

$$H_0(\theta-\pi) = \sum_{n=0}^{N-1} h_0(n) e^{-j(\theta-\pi)n} = \sum_{n=0}^{N-1} h_0(n) e^{j\pi n} e^{-j\theta n}\ , \tag{3.35}$$

$$H_1(\theta) = \sum_{n=0}^{N-1} h_1(n) e^{-j\theta n}\ . \tag{3.35}$$

The corresponding relation between $h_1(n)$ and $h_0(n)$ in the time domain is

$$h_1(n) = h_0(n) e^{j\pi n} = (-1)^n\ h_0(n). \tag{3.36}$$

**Example 3.10.1**

Design an 8-point FIR highpass filter with the cutoff frequency $\theta_c = \pi/2$.

Solution

From Exercise 3.8.1 which is the 8-point half-band lowpass filter, one can simply change the sign of odd indexed coefficients to obtain a highpass filter.

$h(0) = -0.0721$ $h(1) = 0.1010$ $h(2) = 0.1683$ $h(3) = -0.5048$
$h(4) = 0.5048$ $h(5) = -0.1683$ $h(6) = -0.1010$ $h(7) = 0.0721$

Note that the highpass filter we designed here is anti-symmetric.

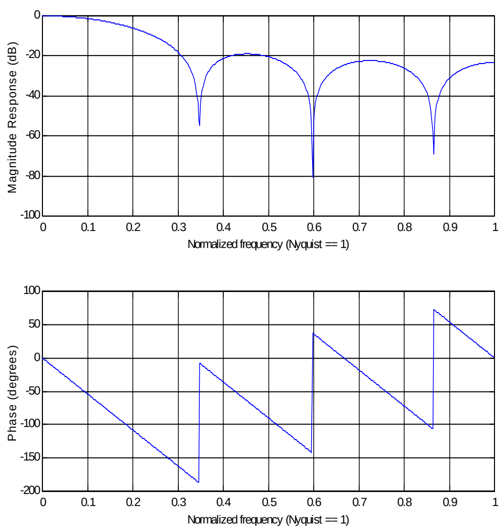
## MATLAB Examples

1. Design a 7-point FIR filter with the cutoff frequency of $0.2\pi$ using the rectangular window (the rectangular window is called the boxcar window in MATLAB).

```
» h = fir1(6,0.2,boxcar(7))
h =
       0.0935   0.1403   0.1734   0.1854   0.1734   0.1403   0.0935
```

2. Plot the frequency response of the filter above.

```
» freqz(h,1)
```




3. Design a 15-point highpass filter with the cutoff frequency of $0.6\pi$ using the Hamming window.

```
» h = fir1(14,0.6,'high',hamming(15));
» freqz(h,1)
```

4. The lowpass IIR filter $h(n) = 0.9^n u(n) = \{1, 0.9, 0.81, \cdots\}$has discrete-time Fourier transform $H(\theta) = 1/(1-0.9e^{-j\theta})$. Normalize all plots so that the gain at dc is 1, i.e. $H(0) = 1$.

   (a) Plot $|H(\theta)|$.

```
H = 1 ./(1 - 0.9*exp(-j*2*pi*.01*(0:99)));     % Evaluation of H(θ) at
                                               %  100 points between 0 and 2π
H = abs(H);                                    % Magnitude of H
H = H/H(1);                                    % For normalization
```

(b) Truncate $h(n)$ to length 10 to obtain $h_{10}(n)$. Compute and plot $|H_{10}(\theta)|$. Observe and explain the phenomenon resulting from the discontinuity of the rectangular window.

```
h10 = 0.9 .^(0:9);              % Select first ten samples of h(n)
H10 = abs( fft(h10, 100) );     % Pad 90 zeros to h10 to make 100-point
                                % DFT that will make a plot smoother
H10 = H10/H10(1);               % Normalization
```

(c) Use a rectangular window of length 20. Compute and plot $|H_{20}(\theta)|$ on the same graph. Compare the ripples in $|H_{10}(\theta)|$ and $|H_{20}(\theta)|$.

```
h20 = 0.9 .^(0:19);             % Select first twenty samples of h(n)
H20 = abs( fft(h20, 100) );
H20 = H20/H20(1);
```

(d) To reduce the ripple, repeat using the Hamming window. Compare the ripples to those in (b) and (c).

```
w10 = hamming(10);        % Get 10-point hamming window coefficients
for n=1:10,
    hh10(n) = h10(n)*w10(n);            % Loop to get a windowed sequence
end
HH10 = abs( fft(hh10, 100) );
HH10 = HH10/HH10(1);
```

### 3.11 Implementation of FIR filters

In general, IIR filters are more efficient, more flexible and need less computational operations than FIR filters. Because IIR filters have both poles and zeros, they can model most frequency characteristics better than FIR filters which have zeros only. For example, sharp peaks can be modeled better with poles and steep valleys can be modeled better with zeros. IIR filter design techniques are extensively discussed in "Digital Signal Processing" authored by Oppenheim and Schafer[2]. However, only FIR filters give linear phase. To simulate one pole, several zeros may be used. In other words, by increasing the order of FIR filters any desired frequency response can be achieved. Of course, higher order FIR filters have more computational complexity than IIR filters. However, IIR filters have major drawbacks. Because IIR filters are recursive, error in the output sample may propagate and this results in numerical instability. As the speed of microprocessors increases, the computational complexity is no longer a problem in most applications. Hence, the FIR filters are more widely used than IIR filters.

Only multiplications and additions are required to implement digital FIR filters. That is a reason why many powerful microprocessors suitable for real-time signal processing are designed so that they can execute one multiplication, one addition and some other data transfer commands in one clock cycle. They are called the DSP (digital signal processor) chips.

Suppose an input signal is very long. A buffer may be used to store temporarily one present and $M$ past input samples as shown in Fig. 3.14.

---

[2] A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, Prentice-Hall, 1975.

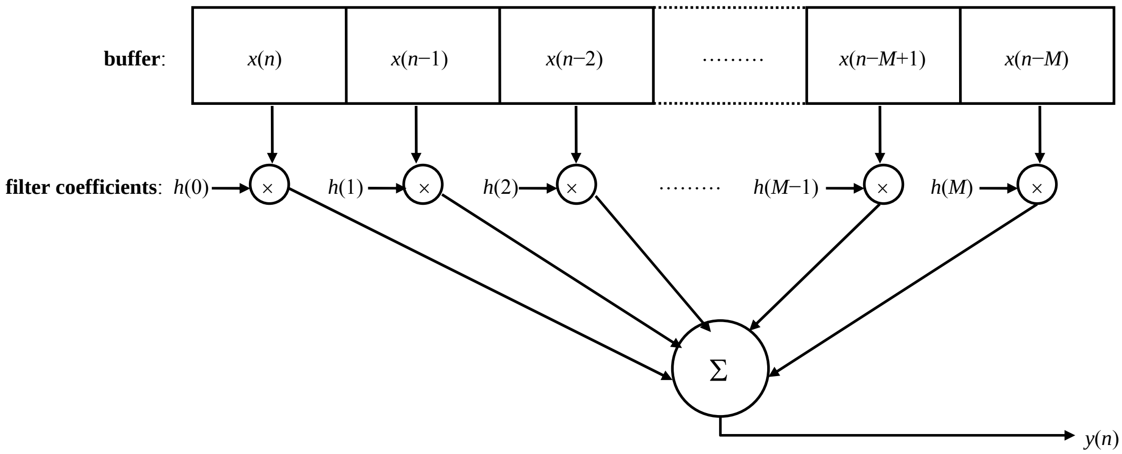



Fig. 3.14 Implementation of a digital FIR filter.

Samples stored in the buffer are multiplied by the corresponding filter coefficients and the results are summed up to yield a present output sample. When the next input sample is available, it is entered in the buffer to become $x(n)$ and samples that were already in the buffer are shifted by one to the right to be ready to compute the next output sample.

The following C program may be used to implement an FIR filter.

```
#include     <stdio.h>
#define      N  16
void   main()
{
   int n;
   double  buf[N], h[N], output;
   FILE   *in, *out;
   in = fopen("signal.dat", "r");
   out = fopen("out.dat", "w");
   h[0] =-0.0024;   h[1] =-0.0042;            // LPF coefficients
   h[2] = 0.0095;   h[3] = 0.0200;
   h[4] =-0.0380;   h[5] =-0.0696;
   h[6] = 0.1374;   h[7] = 0.4472;
   for (n=0; n<=7; n++) h[N-1-n] = h[n];      // Copy h[8] - h[15]
   for (n=0; n<N; n++) buf[n] = 0.;           // Initialize buffer with zeros
   while ( !feof(in) )                  // As long as there is data in the file,
   {                                    // the filtering process is continued.
   fscanf (in, "%lf\n", &buf[0]);    // Get current sample and put it in buffer
      output = 0.;                                      // Initialize output
      for (n=0; n<N; n++) output = output + h[n]*buf[n];   // Current output
      fprintf (out, "%f\n", output);                    // Print output
      for (n=N-1; n>0; n--) buf[n] = buf[n-1];          // Update buffer
   }
   fclose(in);
   fclose(out);
}
```

The numbers of multiplications or additions required to compute each output sample is $M$. In the program above, filter coefficients $h(n)$ for $n = 8, 9, \cdots, 15$ were copies of $h(n)$ for $n = 7, 6, \cdots, 0$. In the case of a symmetric filter, the program to compute the current output sample (the seventh line from the bottom) in the program above can be modified as:

```
for (n=0; n<(N/2); n++) output = output + h[n]*(buf[n] + buf[N-1-n]);
```

Now the number of additions remains the same but the number of multiplications is reduced to $M/2$. The number of storage locations to save filter coefficients is also reduced to $M/2$. Another advantage of using the modified program is that no matter how filter coefficients are quantized, linear phase is still guaranteed.

## 3.12 Conclusions

In an LTI discrete-time system, an input-output relation is expressed in terms of a *difference equation* that consists of multiplications and summations. A $z$-transform is used to analyze discrete-time systems. In this chapter, the $z$-transform has been defined and the relationship between the $z$-transform and the Laplace transform was discussed. Also, the discrete-time systems are classified as FIR or IIR system according to the length of an impulse response. Several methods to compute frequency response of a discrete-time system were introduced. Design of digital FIR filters was explained and practical considerations on implementation of digital filters were discussed.

## Computer Assignment 3.1

1. Write a C program to generate a signal and store it in `signal.dat`:
   $s(n) = 2\cos[2\pi(.05)n] + \sin[2\pi(.2)n]$
   for $n = 0, 1, \ldots, 255$.

2. Find the discrete Fourier transform $S(k)$ using a C program with your own function `void dft()` whose input is $s(n)$ and output are the real and imaginary parts of
   $$S(k) = \sum_{n=0}^{N-1} s(n)e^{-j2\pi nk/N}$$
   for $k = 0, 1, \ldots, 255$ and plot the magnitude and phase spectra. Choose $N$ to be 256.

3. Write a C program to design 16-point FIR lowpass filters with the cutoff frequency of $0.2\pi$ using the rectangular, Hamming, and Blackman windows. Compute and plot the magnitude and the phase response of each filter. Try $N = 1000$.

4. Write a C program to implement the FIR lowpass filter. Use $s(n)$ of part 1 as the input and find the output of the filter. Plot the input and the output.

## C Program Example 3.1

The following C program is to compute coefficients of the16-point FIR lowpass filter (with $\theta_c = 0.5\pi$) using three different windows. The magnitude response of each filter is computed.

```
/*******************************************
*    Lowpass Fiter Design Using Windows     *
*******************************************/
#include      <stdio.h>
#include      <math.h>
#define   PI   3.1415926536
#define   N    100
void           magdft(double x[], double mag[]);
void main()
{
    int n;
    double hrec[16], hham[16], hbla[N], x[N], mag[N], arg;
    double norm1, norm2, norm3;
    FILE   *filter, *out1, *out2, *out3;
    filter = fopen("filter.dat", "w");
    out1 = fopen("rect.dat", "w");
    out2 = fopen("hamm.dat", "w");
    out3 = fopen("blac.dat", "w");
    for (n=0; n<=7; n++)                     /* LPF design with rectangular window */
    {
        arg = (.5 + n) * 0.5 * PI;           /* argument for sinc function */
        hrec[8+n] = .5 * sin(arg)/arg;       /* 0.5*sinc(arg) */
        hrec[7-n] = hrec[8+n];               /* Use even symmetric property */
    }
    for (n=0; n<=15; n++)                    /* Compute LPF coefficients */
    {
        hham[n] = hrec[n]*(0.54 - 0.46*cos(2*PI*n/15) );                /* Hamming */
        hbla[n] = hrec[n]*(.42 - .5*cos(2*PI*n/15) + .08*cos(4*PI*n/15) ); /* Blackman */
    }
    norm1 = norm2 = norm3 = 0.;
    for (n=0; n<=15; n++)                    /* Find the frequency response at dc */
    {
        norm1 = norm1 + hrec[n];
        norm2 = norm2 + hham[n];
        norm3 = norm3 + hbla[n];
    }
    for (n=0; n<=15; n++)                    /* Normalize filter coefficients */
    {
        hrec[n] = hrec[n]/norm1;
        hham[n] = hham[n]/norm2;
        hbla[n] = hbla[n]/norm3;
    }
    fprintf (filter, "\tRectangular\tHamming\t\tBlackman\n");
    for (n=0; n<=15; n++) fprintf(filter, "%d\t%f\t%f \t%f\n",n,hrec[n],hham[n],hbla[n]);

    for (n=16; n<N; n++) x[n] = 0.0;         /* Pad extra zeros out */
    for (n=0; n<=15; n++) x[n] = hrec[n];    /* Copy rectangular filter coefficients */
    magdft (x, mag);                  /* Magnitude response of LPF rectangular window */
    for (n=0; n<N; n++) fprintf (out1, "%f\n", mag[n]);

    for (n=0; n<=15; n++) x[n] = hham[n];    /* Copy Hamming filter coefficients */
    magdft (x, mag);                  /* Magnitude response of LPF using Hamming */
    for (n=0; n<N; n++) fprintf (out2, "%f\n", mag[n]);
    for (n=0; n<=15; n++) x[n] = hbla[n];    /* Copy Blackman filter coefficients */
    magdft (x, mag);                        /* Magnitude response of LPF using Blackman */
    for (n=0; n<N; n++) fprintf (out3, "%f\n", mag[n]);

    fclose(filter);
    fclose(out1);
    fclose(out2);
    fclose(out3);
}
```

```
void    magdft(double x[], double mag[])
{
    int n, k;
    double  XR[N], XI[N];
    for (k=0; k<N; k++)
    {
        XR[k]=0; XI[k]=0.;
        for (n=0; n<N; n++)
        {
            XR[k] = XR[k] + x[n]*cos(2.*PI*k*n/N);
            XI[k] = XI[k] - x[n]*sin(2.*PI*k*n/N);
        }
        mag[k] = XR[k]*XR[k] + XI[k]*XI[k];
        mag[k] = 10.*log10(mag[k]);
    }
}
```

Filter coefficients (filter.dat) are given below.

| | Rectangular | Hamming | Blackman |
|---|---|---|---|
| 0 | -0.031779 | -0.002409 | 0.000000 |
| 1 | -0.036668 | -0.004162 | -0.000580 |
| 2 | 0.043335 | 0.009536 | 0.003153 |
| 3 | 0.052965 | 0.019971 | 0.010040 |
| 4 | -0.068098 | -0.037954 | -0.025333 |
| 5 | -0.095337 | -0.069573 | -0.056707 |
| 6 | 0.158896 | 0.137361 | 0.127400 |
| 7 | 0.476687 | 0.447230 | 0.442026 |
| 8 | 0.476687 | 0.447230 | 0.442026 |
| 9 | 0.158896 | 0.137361 | 0.127400 |
| 10 | -0.095337 | -0.069573 | -0.056707 |
| 11 | -0.068098 | -0.037954 | -0.025333 |
| 12 | 0.052965 | 0.019971 | 0.010040 |
| 13 | 0.043335 | 0.009536 | 0.003153 |
| 14 | -0.036668 | -0.004162 | -0.000580 |
| 15 | -0.031779 | -0.002409 | 0.000000 |

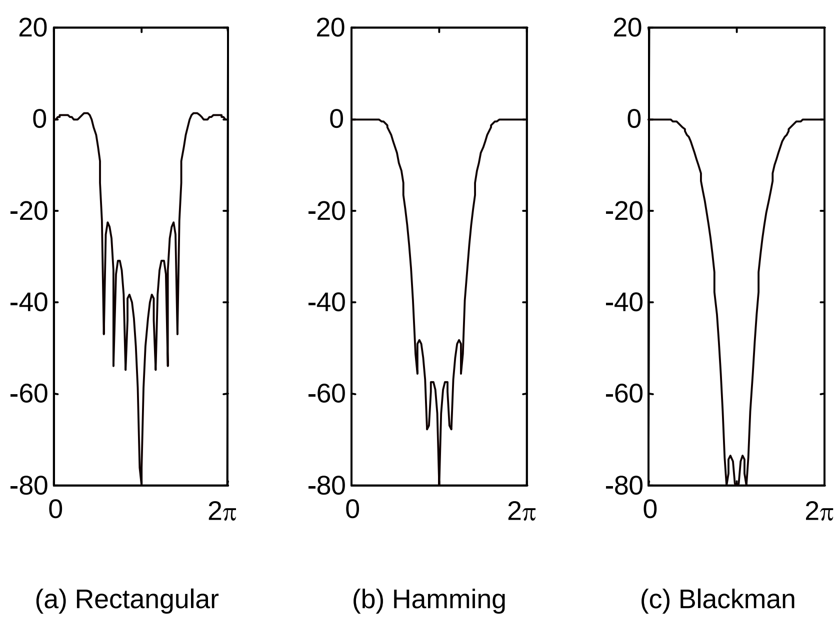



Figure P.3.1  Various FIR lowpass filters.

**C Program Example 3.2**

The following C program computes the outputs of both the lowpass and highpass filters at the same time. The half-band lowpass filter (Hamming window) was obtained from the previous example. Because the lowpass and the highpass filters share the same filter coefficients and only the odd-indexed coefficients have different signs, the number of multiplications can be reduced. In this program, we decompose the convolution into two summations: one for the even indices and the other for the odd indices.

$$\text{Lowpass out for each } n = \sum_{m=0}^{M} h(m)x(n-m)$$

$$= \sum_{m \text{ even}} h(m)x(n-m) + \sum_{m \text{ odd}} h(m)x(n-m)$$

$$= \text{even}(n) + \text{odd}(n)$$

Now the highpass filter output becomes

$$\text{Highpass out for each } n = \text{even}(n) - \text{odd}(n).$$

```c
#include      <stdio.h>
#define    N   16
#define    N1  15

void   main()
{
    int n;
    double  buf[N], h[N], even, odd, low, high;
    FILE   *in, *out1, *out2;
    in = fopen("signal.dat", "r");
    out1 = fopen("lpf.dat", "w");
    out2 = fopen("hpf.dat", "w");
    h[0] =-0.0024;     h[1] =-0.0042;        /* LPF coefficients */
    h[2] = 0.0095;     h[3] = 0.0200;
    h[4] =-0.0380;     h[5] =-0.0696;
    h[6] = 0.1374;     h[7] = 0.4472;
    for (n=0; n<=7; n++) h[N1-n] = h[n];             /* Copy h[8] - h[15] */
    for (n=0; n<=N1; n++) buf[n] = 0.;               /* Initialize buffer with zero */
    while ( !feof(in) )                      /* As long as there is data in the file, */
    {                                        /* the filtering process is continued.   */
        fscanf (in, "%lf\n", &buf[0]);
        even = odd = 0.;
        for (n=0; n<N; n=n+2) even = even + h[n]*buf[n];     /* Even indexed coefficients*/
        for (n=1; n<N; n=n+2) odd = odd + h[n]*buf[n];       /* Odd indexed coefficients */
        low = even + odd;                                    /* LPF output */
        high = even - odd;                                   /* HPF output */
        fprintf (out1, "%f\n", low);
        fprintf (out2, "%f\n", high);
        for (n=N1; n>0; n--) buf[n] = buf[n-1];              /* Update buffer */
    }
    fclose(in);
    fclose(out1);
    fclose(out2);
}
```

## PROBLEMS

1. Find the $z$-transform of the following and sketch the pole-zero plot.

   (a) $x(n) = 0.8^n u(n)$

   (b) $x(n) = [0.5^n + (-0.4)^n]u(n)$

   (c) $x(n) = -0.5^n u(n-2) + (-0.4)^n u(n)$

   (d) $x(n) = 0.8^n [u(n) - u(n-10)]$

   **Ans.** (a) $\dfrac{1}{1-0.8z^{-1}}$ (b) $\dfrac{2-0.1z^{-1}}{(1-0.5z^{-1})(1+0.4z^{-1})}$ (c) $\dfrac{1-0.5z^{-1}-0.25z^{-2}-0.1z^{-3}}{(1-0.5z^{-1})(1+0.4z^{-1})}$
   (d) $\dfrac{1-0.8^{10}z^{-10}}{1-0.8z^{-1}}$

2. Find the inverse $z$-transform of the following sequence.

   (a) $X(z) = \dfrac{1}{z-0.8}$, ROC: $|z| > 0.8$

   (b) $X(z) = \dfrac{1-0.5z^{-1}}{1+0.75z^{-1}+0.125z^{-2}}$, ROC $|z| > 0.5$

   (c) $X(z) = \dfrac{1-0.5z^{-1}}{1-0.25z^{-1}}$, ROC: $|z| > 0.25$

   **Ans.** (a) $0.8^{n-1}u(n-1)$ (b) $[4(-0.5)^n - 3(-0.25)^n]u(n)$ (c) $\delta(n) - 0.25^n u(n-1)$

3. Find the inverse $z$-transform of $H(z) = \dfrac{4}{1-.75z^{-1}+.125z^{-2}}$.

4. Find the $z$-transform of a double-sided sequence $p(n) = e^{-0.1|n|}$ and the region of convergence.

**Ans.** $\dfrac{1}{1-e^{-0.1}z^{-1}}-\dfrac{1}{1-e^{0.1}z^{-1}}$ , ROC: $e^{-0.1}<|z|<e^{0.1}$

5. A continuous-time signal $x(t) = 5e^{-0.4t}u(t)$ where $u(t)$ is the unit step function.
   (a) Find the Laplace transform of $x(t)$. Find the pole.
   (b) The signal $x(t)$ is sampled at every 0.2[s]. Find the $z$-transform of the discrete-time signal. Find the pole.

6. Consider an LTI discrete-time system whose input-output relationship is described by the following difference equation.

   $$y(n) = 0.8y(n-1) + 3x(n).$$

   (a) Find the transfer function of the system.
   (b) Find the impulse response of the system.

7. An LTI system's input-output relation is described by $y(n) = 0.8y(n-1) + 2x(n-1)$.
   (a) Find the impulse response of the system for $n = 1, 2, 3, 4, 5$ by inputting a unit impulse input.
   (b) Find the transfer function $H(z)$ of the system.
   (c) Find the expression of the impulse response $h(n)$.

8. The LTI discrete-time system's input-output relation is given by

   $$y(n) = -1.25y(n-1) - .375y(n-2) + x(n).$$

   (a) Find the impulse response of the system for $n = 0, 1, 2, 3$ by inputting a unit impulse.
   (b) Find the impulse response by taking the inverse $z$-transform of the transfer function.

   **Ans.** (a) $h(0) = 1$, $h(1) = -1.25$, $h(2) = 1.1875$, $h(3) = -1.0156$
   (b) $3(-.75)^n - 2(-.5)^n u(n)$

9. A causal LTI system has the frequency response

   $$H(\omega) = \frac{10}{1 + j0.1\omega}.$$

   Sketch the Bode plot.

10. Find and sketch the frequency response of $H(z) = 1 - z^{-1}$. What kind of filter is this?

11. Find and sketch the frequency response of $H(z) = (1 - z^{-1})/(1 - .25z^{-1})$. What kind of filter is this?

**Ans.** $|H(0)| = 0$, $|H(\pi/4)| = 0.909$, $|H(\pi/2)| = 1.372$, $|H(\pi)| = 1.6$

12. Find and sketch the magnitude and phase responses of a system whose transfer function is specified by
(a) $H(z) = 1 + z^{-1}$
(b) $H(z) = 1 - 2z^{-1} + z^{-2}$.

13. Find and sketch the magnitude response for the 3-point moving average filter with $h(0) = 0.5$, $h(-1) = h(1) = 0.25$. Does it have zero phase, constant phase, or linear phase? Does this filter cause any delay? Is this filter causal?

14. Sketch the magnitude response of $H(z) = 1 + 2z^{-1} + z^{-2}$. What kind of filter is this?

15. An impulse response of a filter is given by
$h(n) = -0.25\delta(n) + 0.5\delta(n-1) - 0.25\delta(n-2)$.
(a) Carefully sketch the magnitude response.
(b) Carefully sketch the phase response.
(c) Find the 3-dB cutoff frequency.
(d) What kind of filter is it?
(e) Find the delay of the filter.

16. Design a 4-point FIR lowpass filter with the cutoff frequency of $0.3\pi$ using a Hamming window.

**Ans.** {0.0350, 0.4650, 0.4650, 0.0350}

17. Design a 5-point FIR highpass filter with the cutoff frequency of $0.7\pi$ using a Blackman window.

**Ans.** {−0.0000, −0.1843, 0.6314, −0.1843, −0.0000}