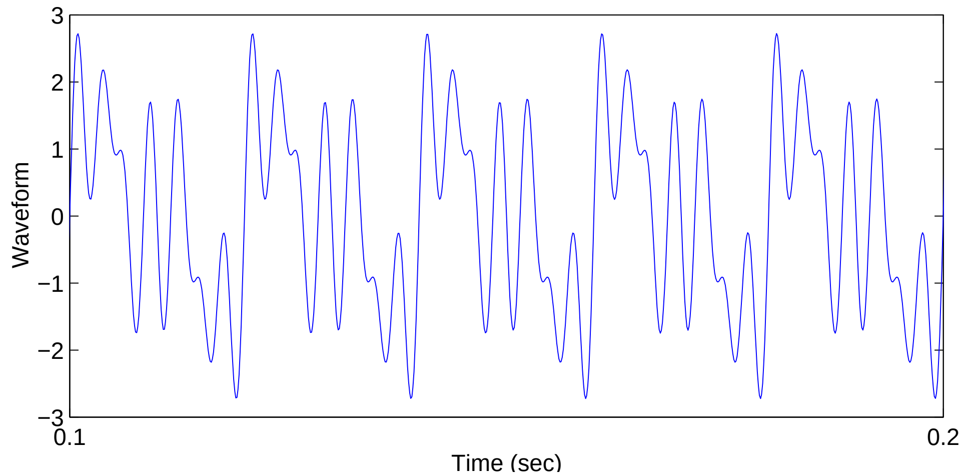

Figure 3.1: A sinusoidal signal with four frequency components: $f_1 = 50$, $f_2 = 100$, $f_3 = 250$, $f_4 = 350$Hz.

Lesson 3: Introduction to Filtering and Feedforward Filters

Introduction

Required Reading:
Ch. 4, 12 (§9)

In this lesson, I will introduce the concept of *filtering* and the operation of basic feedforward filters. By the end, you should understand some important terms related to filters, for example, *frequency response*, *phase response*, *transfer function* and *zeros* of a *transfer function*. You should be able to implement simple digital filters on a computer and use them to solve some simple signal processing problems.

The concept of filtering should not be new to you. For example, when you saw that the content to be covered here is from chapter 4, you would pay attention to this chapter and ignore the others. Your mind performed a *bandpass* filter with the pass band being chapter [4]. Some other kinds of filters are low pass, high pass, and bandstop. Using the same example, a low pass filter allows you to attend to all the chapters in the book, from the beginning to the end of the pass band, say chapter 4; the high pass on the other hand will pass only the chapters beyond a certain chapter, say beyond chapter 5; the bandstop is the one that allows you to pay attention to all the chapters *except* the stop band, say [4].

In the signal processing domain, filters exclude and/or include signal *frequencies*. For example, consider a signal $x(t)$ (where t is time) with four sinusoidal components. It has frequencies at $f_1 = 50$, $f_2 = 100$, $f_3 = 250$, and $f_4 = 350$:

$$x = \sin(2\pi t f_1) + \sin(2\pi t f_2) + \sin(2\pi t f_3) + \sin(2\pi t f_4) \tag{3-1}$$

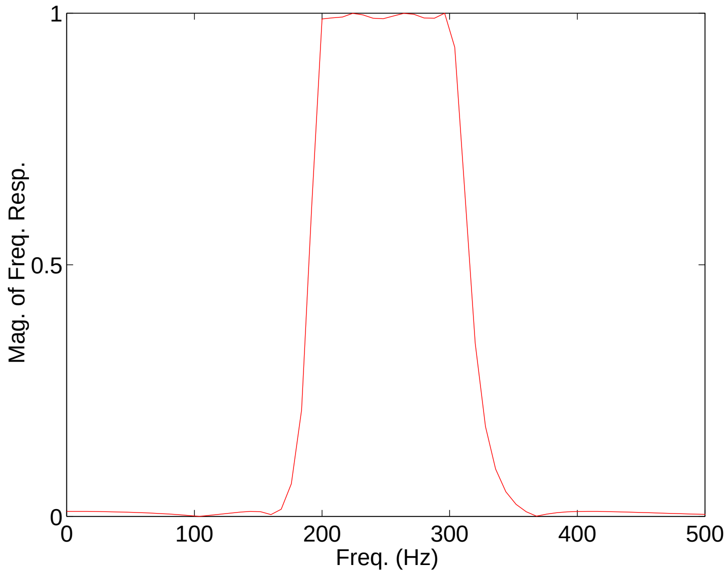

Figure 3.2: Filter's frequency response. The pass band is [200 300] Hz.

Figure 3.1 shows a plot of about 0.1 second of this signal. Its four frequencies are shown as green peaks in figure 3.4. Let's say that we would like to have a filter to keep the $f_3 = 250$Hz sinusoid and get rid of the f_1, f_2, and f_4 sinusoids. This filter can be visualized as in figure 3.2. Here, the horizontal axis is frequency and the vertical axis is magnitude of the filter's response. The filter is designed to have a frequency passband between 200 and 300 Hz, which means that only the signal's frequency components within this band are allowed to pass. Figure 3.3 shows this filter's output — in other words, the filtered signal — plotted along time. Another way to look at the filtered signal is its frequency components, which are shown in figure 3.4 (magenta). This latter graph clearly shows that, after filtering, the signal is very close to a 250 Hz sinusoid, exactly as expected.

At this point, you should have an idea of what filters do. Next, I will explain *how* filters work.

Feedforward Filters

Delaying a phasor

In lesson 1 we learned the term *phasor*, a complex sinusoid expressed as $e^{j\omega t}$. We also saw that this representation makes the math simpler for adding sinusoids, at least. A phasor's magnitude is one, its frequency is ω, and it angle is ωt, where t is time. It moves around the unit circle (such as textbook chapter 4, figure 1.1) counter-clockwise along time (solid arrow). If we delay it by τ sec, we can write this as:

$$e^{j\omega(t-\tau)} = e^{-j\omega\tau}e^{j\omega t} \tag{3-2}$$

This is the product of two phasors at the same frequency. This doesn't do anything except

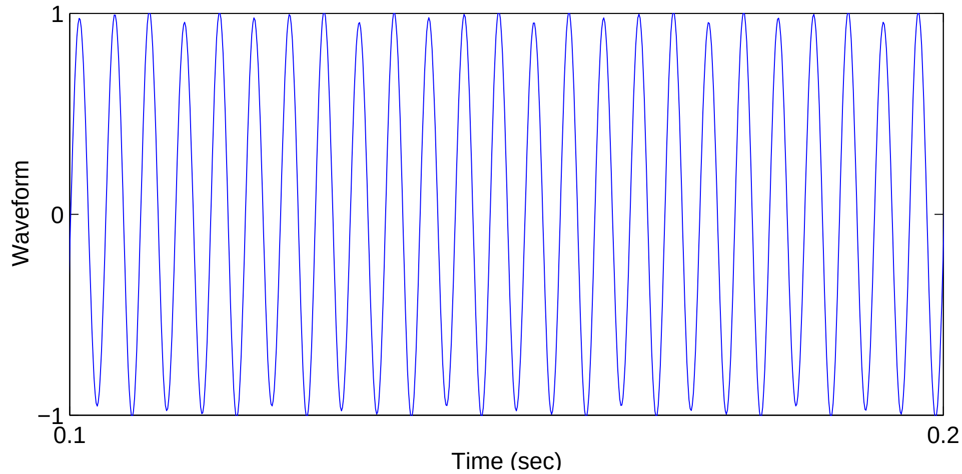

Figure 3.3: The signal (3-1) after filtering out frequencies 50, 100 and 350Hz.

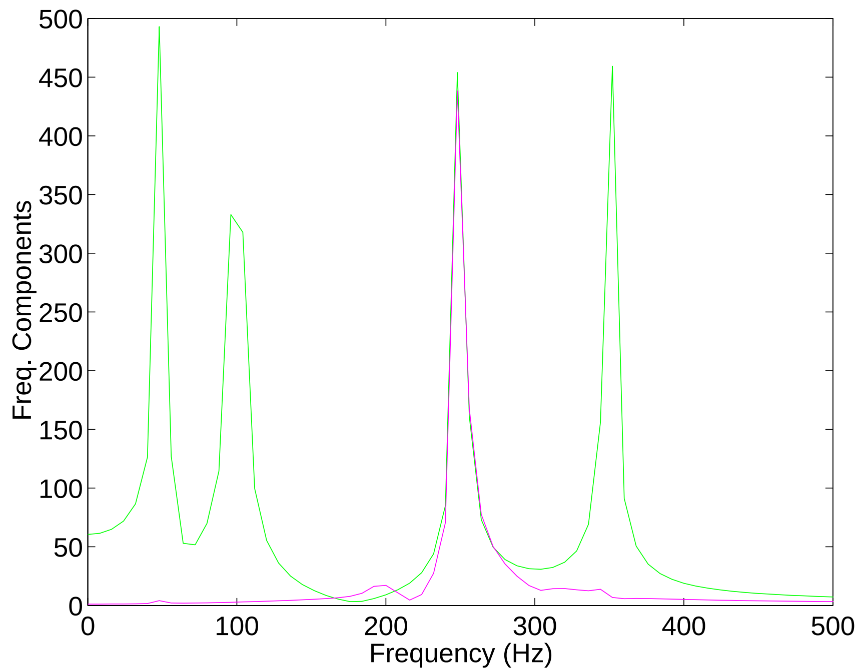

Figure 3.4: Spectrum of four-sinusoid signal (green) and its filtered version (magenta).

rotate the phasor by $-\omega\tau$ (i.e., it doesn't change its magnitude).

A simple feedforward filter

Filters combine delayed versions of signals. We have already seen that signals are made up of phasors. A feedforward filter's block diagram is shown in textbook figure 2.1, chapter 4 (for the remainer of this lesson, all figures are from chapter 4). Its output is the summation of its input and a scaled (or *weighted*) version of its input delayed by τ. This can be written as, given a input signal x_t, its delayed one $x_{t-\tau}$ and coefficient a_1, the output y_t is

$$y_t = x_t + a_1 x_{t-\tau} \tag{3-3}$$

When the input is a phasor, equation (3-3) becomes:

$$y_t = e^{j\omega t} + a_1 e^{j\omega(t-\tau)} \tag{3-4}$$

Notice that the input and the output have the same frequency, ω. This addition is illustrated in text book figure 2.2, where the angle between x_t and $x_{t-\tau}$ is $-\omega\tau$.
We can factor out the $e^{j\omega t}$ in equation (3-4) to obtain:

$$\begin{aligned} y_t &= [1 + a_1 e^{-j\omega\tau}]e^{j\omega t} && \text{(3-5)} \\ &= H(\omega)e^{j\omega t} && \text{(3-6)} \\ &= H(\omega)x_t && \text{(3-7)} \end{aligned}$$

where

$$H(\omega) = \frac{y_t}{x_t} = 1 + a_1 e^{-j\omega\tau} \tag{3-8}$$

is called the filter's *frequency response*. Remember that τ is a constant delay; what can vary here (besides t, which is the same for both x_t and y_t) is the frequency of the input, ω. So, $H(\omega)$ is a complex function of frequency. Frequency response is generally written in a polar form with its magnitude response $|H(\omega)|$ and phase response $\theta(\omega)$.

$$H(\omega) = |H(\omega)|e^{-j\theta(\omega)} \tag{3-9}$$

The filter multiplies its input by its magnitude response $|H(\omega)|$ and shifts its phase by the phase response $\theta(\omega)$. These two functions define the filter's behavior. For this particular example,

$$\begin{aligned} |H(\omega)| &= |1 + a_1 e^{-j\omega\tau}| && \text{(3-10)} \\ &= |1 + a_1^2 + 2a_1 \cos(\omega\tau)|^{\frac{1}{2}} && \text{(3-11)} \end{aligned}$$

Since $|\cos(\omega\tau)| \leq 1$, the maximum value $|H(\omega)|$ can reach is $(1 + a_1)$, which occurs when the angle $\omega\tau = n2\pi, n = 0, 2, 4, ...$ (zero or even multiples of π). Why is this? (*Popup answser: These are the values of* $\omega\tau$ *for which* $\cos\omega\tau = \pm 1$, *respectively.*) Similarly, the minimum of

$|H(\omega)|$ is $1 - a_1$, which happens when $\omega\tau = n\pi, n = 1, 3, 5...$ (odd multiples of π). Textbook figure 2.3 shows the magnitude response versus frequency ω when $a_1 = 0.99$ and $\tau = 167\mu$s (μs is microseconds, or $10^{-6}s$). Converting ω (radians/second) to Hz by $\omega = 2\pi f$, that figure describes a filter with passbands centered at 0, 6, and 12 kHz and notches at 3, 9, and 15 kHz. You can easily see how the input signal's frequency components (remember that all periodic functions can be expressed as sums of complex sinusoids) will be altered by the filter: when they are within the filter's passbands, they will be passed through; they will be reduced in magnitude or filtered out when their frequency matches the low magnitude response. Figure 3.2 is another example of a bandpass filter with passband [20 30] Hz. The input signal's frequencies out of this band are filtered out. This is shown in figure 3.4: in the output signal, only the frequency component at $f = 25Hz$ is left.

Self-Test Exercise

1. Use Euler's formula and the definition of the magnitude of a complex vector to derive (3-11) from (3-10). (*Popup Answer: Subsituting* $e^{-j\omega\tau} = \cos\omega\tau + j\sin\omega\tau$, *we obtain* $|H(\omega)| = |1 + a_1(\cos\omega\tau + j\sin\omega\tau)| = |1 + a_1\cos\omega\tau + ja_1\sin\omega\tau|$. *The magnitude of a complex number is the square root of the absolute value of the sum of the squares of its real and imaginary parts, so* $|H(\omega)| = |(1 + a_1\cos\omega\tau)^2 + a_1^2\sin^2\omega\tau|^{\frac{1}{2}}$. *Computer the two squared terms, remember that* $\sin^2 x + \cos^2 x = 1$, *and you're home free.*)

Digital Filters

Electrical engineers have spent a lot of time developing different kinds of filter transfer functions for different classes of filters. You may run across names like Butterworth or Chebyshev. The motivation behind this has been to produce filters with good properties (flatness of the passband, steepness of the rolloff from the passband, minimal phase distortion) that is still implementable in analog hardware (operational amplifiers, resistors, and capacitors). However, once we digitize a signal, we can filter it in a computer because filtering is a mathematical operation and that's what computers do. By directly implementing a filter's transfer function, we can implement digital filters that may be difficult or impossible to implement in analog hardware.

To be implemented on a computer, an analog filter must be discretized in its variables, to yield a *digital filter*. There are two important points here related to digital filters compared to analog filters:

1. Time is expressed as an integer times a *sampling period*.

2. Only frequencies below the *Nyquist frequency* can be represented accurately.

For the first point, digitized time t or τ is in form of:

$$t = nT_s \tag{3-12}$$

or,

$$\tau = nT_s \tag{3-13}$$

where n is an integer, and T_s is the time interval between samples or the sampling period, whose units are sec/sample. When the *sampling frequency* is f_s (samples/second), $T_s = 1/f_s$. So the phasor becomes $e^{j\omega\tau} = e^{j\omega n T_s}$, with its exponent having units of radians/sec $\times$ samples $\times$ sec/samples = radians.

It's rather cumbersome to have to carry around T_s or f_s in all our equations. Additionally, if we keep these variables we will always need to note and remember the signal in question's Nyquist frequency. To simplify our notation, let's redefine our digital ω to be the analog ωT_s (with units of radians/sample) and replace n (the sample counter) with a digital time t. Yes, it would probably be better to use different symbols for these two, but I'll stick to these to be consistent with the textbook. Our phasor then becomes $e^{j\omega t}$, and has the same form as the analog version. However, remember that when we are speaking of digital filters ω has different units.

For the second point, recall from the previous lesson that the Nyquist frequency is:

$$f_{Nyquist} = \frac{f_s}{2} \tag{3-14}$$

Only frequencies below $f_{Nyquist}$ will be accurately represented after sampling, beyond it they are aliased to frequencies below $f_{Nyquist}$. It is important to restrict the frequency range to $f_{Nyquist}$ so you can get correct results.

Again, for convenience, the digital frequency can be normalized by f_s,

$$\frac{f}{f_s} \tag{3-15}$$

and

$$\frac{f_{Nyquist}}{f_s} = \frac{f_s/2}{f_s} = \frac{1}{2} \tag{3-16}$$

Since the analog $f \leq f_s/2$, the digital f is a fraction that ranges between zero and 1/2 — a fraction of the digital sampling rate. We can multiply it by f_s to convert it back to the analog units of Hz.

If the normalized $f_{Nyquist}$ is 1/2, the normalized $\omega_{Nyquist}$ is

$$\omega_{Nyquist} = 2\pi f_{Nyquist} = 2\pi \times 1/2 = \pi \tag{3-17}$$

Textbook figure 3.1 shows the same example as textbook figure 2.3 for magnitude of frequency response, but here τ is one T_s delay, normalized as 1 according to (3-13). So equation (3-11) becomes:

$$|H(\omega)| = |1 + a_1^2 + 2a_1 \cos(\omega)|^{\frac{1}{2}} \tag{3-18}$$

Since $\omega_{Nyquist} = \pi$, $|H(\omega_{Nyquist})| = (1 - a_1)$. This response corresponds to the first notch in textbook figure 2.3; a notch at digital $f = 1/2$ (one-half the sampling rate). The filter has a broad band from 0 until the notch, so it can pass all the frequencies from 0 to below 1/2 — this filter is a lowpass filter. Figure 3.4 is a bandpass filter and textbook figure 4.1 is a bandstop one (which removes a particular range of frequencies in its band [0.22 0.32] and passes all others).

Delay as an Operator

We're still on our quest to make the mathematics of filter design and analysis as simple as possible. Generally, when filter is feedforward with many delay terms, the input x_t to output y_t relation can be written as:

$$y_t = \sum_{k=0}^{K} a_k x_{t-k} \tag{3-19}$$

When x_t is the phasor $e^{j\omega t}$ this becomes

$$\begin{aligned} y_t &= \sum_{k=0}^{K} a_k e^{j\omega t - kj\omega} && \text{(3-20)} \\ &= e^{j\omega t} \sum_{k=0}^{K} a_k e^{-kj\omega} && \text{(3-21)} \\ &= x_t \sum_{k=0}^{K} a_k e^{-kj\omega} && \text{(3-22)} \end{aligned}$$

Now that you're comfortable with the concept of multiplication by the phasor $e^{-kj\omega}$ being a delay, let's get rid of it. Seriously, though, it's a lot of writing; this phasor is acting as an *operator* which, when applied to another phasor, delays it. We can use another symbol for convenience sake. In this case, we define an *operator* z as follows:

$$z = e^{j\omega} \tag{3-23}$$

This is a symbol that represents the application of an action on an object (an operation, so z is an operator). So a k time delay $e^{-kj\omega}$ can be written as

$$z^{-k} = e^{-kj\omega} \tag{3-24}$$

Here z^{-k} is a k time delay operator, where $k = 0, 1, 2, \ldots$ denotes the $0, 1, 2, \ldots k, \ldots$ time step delay. Operators are a general concept in mathematics, which can be used in many other circumstances to simplify notation.

Example 1: Consider a "square" operator S, which squares the thing on which it operates. When it operates on a variable x, we get:

$$Sx = x^2$$

Example 2: The transpose operator T is commonly used in linear algebra. It *transposes* the matrix to which it is applied, exchanging its rows and columns. For example, when applied to the matrix $\mathbf{A}$,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

the result is

$$\mathsf{T}\mathbf{A} = \mathbf{A}^{\mathsf{T}} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Now we're ready to analyze how a digital filter responds to an input. Let $X = \{x_1, x_2, \ldots, x_t, \ldots\}$ be the entire digital input signal: the ordered set of all samples. $X|_t = x_t$ is the signal X's value at time t. We use the same notation to produce Y for the ordered set of output samples. Instead of using just one sample (x_t and y_t) as in equation (3-22), let's substitute X and Y and the delay operator z^{-k} to obtain an equation that describes the action of a filter on an entire digital signal:

$$\begin{aligned} Y &= \sum_{k=0}^{K} a_k z^{-k} X && \text{(3-25)} \\ &= [a_0 + a_1 z^{-1} + \cdots + a_k z^{-k} + \cdots]X && \text{(3-26)} \end{aligned}$$

The benefit of using the delay operator is that it makes the task of factoring out the entire signal X simple. If we call the expression in the square brackets $\mathcal{H}(z)$,

$$\begin{aligned} \mathcal{H}(z) &= \sum_{k=0}^{K} a_k z^{-k} && \text{(3-27)} \\ &= a_0 + a_1 z^{-1} + \cdots + a_k z^{-k} + \cdots && \text{(3-28)} \end{aligned}$$

we get

$$Y = \mathcal{H}(z)X \tag{3-29}$$

$\mathcal{H}(z)$ is also an operator, which transfers the input signal X to the output signal Y, so it is called the filter's *transfer function*. The relation between the analog frequency response and digital transfer function is

$$H(e^{j\omega}) = H(\omega) \tag{3-30}$$

(where we continue to adhere to the textbook's convention of using exactly the same symbols for both, regardless of how confusing it might be). Textbook figure 5.1 describes the situation where $k = 1$,

$$Y = \mathcal{H}(z)X = [a_0 + a_1 z^{-1}]X \tag{3-31}$$

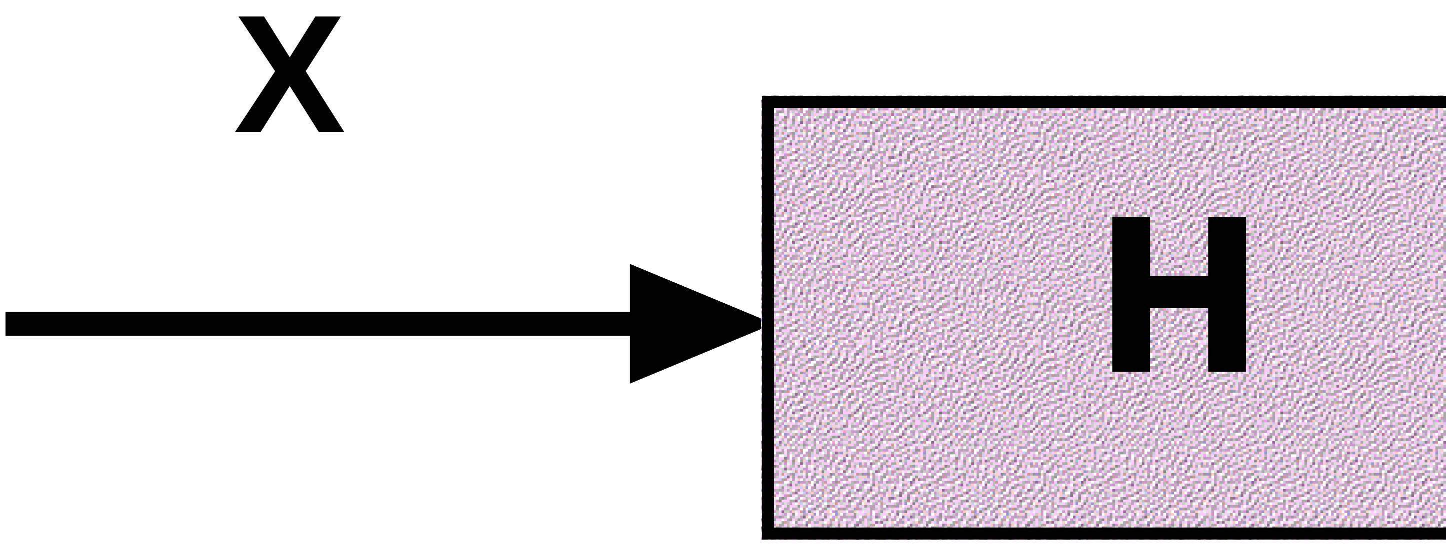

Figure 3.5: Treat transfer function as a black box.

Equation (3-31) says that the output equals the weighted sum of the input signal and the input signal delayed by one time step (one sampling interval).

We can treat the transfer function as a black box that does everything that the filter needs to do. We only to pay attention to the input and output, as in figure 3.5. This give us a way to combine different filters, simply by composing block diagrams. Textbook figure 5.2 describes two simple filters $\mathcal{G}(z)$ and $\mathcal{H}(z)$ connected in series. $\mathcal{G}(z)$ has input X and output W, and $\mathcal{H}(z)$ takes $\mathcal{G}(z)$'s output as its input and outputs Y. Starting from the output of this system, this can be written

$$Y = \mathcal{H}(z)W = \mathcal{H}(z)[\mathcal{G}(z)X] = [\mathcal{H}(z)\mathcal{G}(z)]X \tag{3-32}$$

This suggests that the combined transfer function is $\mathcal{H}(z)\mathcal{G}(z)$. In fact, we can interchange the order

$$\mathcal{H}(z)\mathcal{G}(z) = \mathcal{G}(z)\mathcal{H}(z) \tag{3-33}$$

This gives us a way to represent digital filtering as just multiplication by a polynomial (and the product of two polynomials is just another polynomial).

Self-Test Exercises

1. Write equation (3-22) for $k = 0, 1, 2, 3$. (*Popup Answer: For* $k = 0$, $y_t = a_0 x_t$*;* $k = 1$, $y_t = x_t(a_0 + a_1 e^{-j\omega})$*;* $k = 2$, $y_t = x_t(a_0 + a_1 e^{-j\omega} + a_2 e^{-2j\omega})$*;* $k = 3$, $y_t = x_t(a_0 + a_1 e^{-j\omega} + a_2 e^{-2j\omega} + e^{-3j\omega})$.)

2. Given the signal $x(t) = \sin t$ and the derivative operator $D = \mathrm{d}/\mathrm{d}t$, what is $Dx(t)$? (*Popup Answer:* $Dx(t) = \cos t$.)

3. When

$$\begin{aligned} \mathcal{G}(z) &= a_0 + a_1 z^{-1} \\ \mathcal{H}(z) &= b_0 + b_1 z^{-1} \end{aligned}$$

with a_0, a_1, b_0, and b_1 are constants, show that $\mathcal{H}(z)\mathcal{G}(z) = \mathcal{G}(z)\mathcal{H}(z)$. (*Popup Answer: Multiplying* $\mathcal{H}(z)\mathcal{G}(z)$*, we obtain* $(a_0 + a_1 z^{-1})(b_0 + b_1 z^{-1}) = a_0 b_0 + (a_0 b_1 + a_1 b_0) z^{-1} + a_1 b_1 z^{-2}$*. Multiplying* $\mathcal{G}(z)\mathcal{H}(z)$*, we get* $(b_0 + b_1 z^{-1})(a_0 + a_1 z^{-1}) = b_0 a_0 + (b_0 a_1 + b_1 a_0) z^{-1} + b_1 a_1 z^{-2}$*. Because constants are commutative, these two expressions are equal. In other words, series combination of filters is commutative because multiplication of polynomials is commutative.*)

The z-plane

Like the textbook, I just used z as a delay operator without much comment as to why I picked the same symbol as I've used for the complex plane. The operator $z = e^{j\omega}$ is obviously a complex variable in the z-plane. For any value of ω, it lies on a circle of radius one, at an angle of ω relative to the positive real axis — it is the polar representation of a complex number. As ω varies from zero to 2π (or $-\pi$ to $+\pi$, if you prefer not to consider $\omega > \omega_{Nyquist}$), its path is the unit circle in the z-plane. Since the Nyquist frequency $\omega_{Nyquist} = \pi$, we are only interested the top of half of the circle running from $\omega = 0$ to $\omega = \pi$ shown in textbook figure 6.1.

Let's examine a simple digital filter with one time delay:

$$y_t = x_t - a_1 x_{t-1} \tag{3-34}$$

Just for convenience, I have used subtraction instead of summation (or, equivalently you can think that I used a negative weight on the delayed signal). Using the delay operator, this becomes,

$$Y = [1 - a_1 z^{-1}]X = \mathcal{H}(z)X \tag{3-35}$$

Obviously,

$$\mathcal{H}(z) = 1 - a_1 z^{-1} = (1 - a_1 z^{-1})\frac{z}{z} = \frac{z - a_1}{z} \tag{3-36}$$

When z is on the unit circle ($z = e^{j\omega}$), the magnitude of $\mathcal{H}(e^{j\omega})$ is the same as the magnitude of $H(\omega)$, so the magnitude of the frequency response is

$$|H(\omega)| = |\mathcal{H}(z)| = \frac{|z - a_1|}{|z|} = |z - a_1| \tag{3-37}$$

because $|z| = 1$ when we're on the unit circle. You can see that although $z = 0$ makes the denominator of $\mathcal{H}(z)$ zero, we're not considering that case — we've already said that we're on the unit circle; that $|z| = 1$. The value that makes the denominator of $\mathcal{H}(z)$ zero is called a *pole*, which we will talk about in the next lesson.

Equation (3-37) tells us that the magnitude of the transfer function is the distance between z and a_1 in the complex plane. Since z is a vector from the origin to the unit circle (see textbook figure 6.2) and a_1 is a constant, which can be any number here (the figure shows the case of a_1 real and $a_1 \leq 1$), $|H(\omega)|$ is equal to the length of vector from a_1 to the unit circle where z points. As this length becomes shorter, $|H(\omega)|$ becomes smaller. We can see that is the case when z nears a_1.

The other thing the equation (3-37) tells us is that when $z = a_1$, $H(\omega) = 0$, so a_1 is a *root* of $H(\omega)$, also called a *zero*, which makes the magnitude of the frequency response reach its minimum. Obviously, when the zero (a_1) is near $\omega = 0$, this results in a high pass filter because it doesn't pass frequencies near zero (low frequencies). Similarly, when the zero (a_1) is near $\omega = \pi$ we obtain a low pass filter: high frequency components are filtered out.

Example 3: Consider the general two-time-step delay feedforward filter,

$$y_t = x_t + a_1 x_{t-1} + a_2 x_{t-2}$$

where a_1 and a_2 are real constants. The transfer function is

$$\begin{aligned} \mathcal{H}(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} \\ &= \frac{z^2 + a_1 z + a_2}{z^2} \end{aligned}$$

The magnitude response is

$$|\mathcal{H}(z)| = \frac{|z^2 + a_1 z + a_2|}{|z^2|} \tag{3-38}$$

It has poles at zero; however, as we already know, $|z^2| = 1$, so they don't affect the magnitude response. So $|\mathcal{H}(z)|$ becomes:

$$|\mathcal{H}(z)| = |z^2 + a_1 z + a_2|$$

The zeros of this magnitude response are merely the roots of any polynomial of order two,

$$z_{1,2} = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2}}{2} \tag{3-39}$$

The possible types of zeros are:

- When $a_1^2 > 4a_2$, there are two real zeros.
- When $a_1^2 = 4a_2$, there are a pair of zeros at $-a_1$.
- When $a_1^2 < 4a_2$, a pair of complex conjugate zeros result.

Since these are the roots of the magnitude response, we can factor the magnitude response as:

$$|\mathcal{H}(z)| = |(z - z_1)(z - z_2)| \tag{3-40}$$

We can write the zeros in polar form,

$$z_i = r_i e^{j\omega_{0_i}}, \quad i = 1, 2 \tag{3-41}$$

where $r_i > 0$ are the radii where the zeros are located, and ω_{0_i} their angles. Depending on the angles ω_{0_i}, the zeros can be either real or complex. For example, when $\omega_{0_i} = 0$, the zero lies on the z-plane's real axis, and when $\omega_{0_i} = \pi/2$, it is on the imaginary axis. Actually, when filter's coefficients are real, if one of the zeros is a complex number, the other one will be its conjugate mate (if one of them is real, the other will be real, too). In that last case, $\omega_0 = -\pi/2$ would be the other zero's angle. So, for this filter and real coefficients, the pair of complex zeros can be written as

$$z_{1,2} = re^{\pm j\omega_0} \tag{3-42}$$

Figures 3.6 and 3.7 show two different sets of zero locations. The filter's magnitude response for those two sets of zeros are presented in figures 3.8 and 3.9.

Phase Response

So far we have been talking about the magnitude response of $H(\omega)$. In the last topic of this lesson, let's talk about is its phase response. We already know that $H(\omega)$ can be expressed in polar form, with its magnitude and angle

$$H(\omega) = |H(\omega)|e^{j\theta(\omega)} \tag{3-43}$$

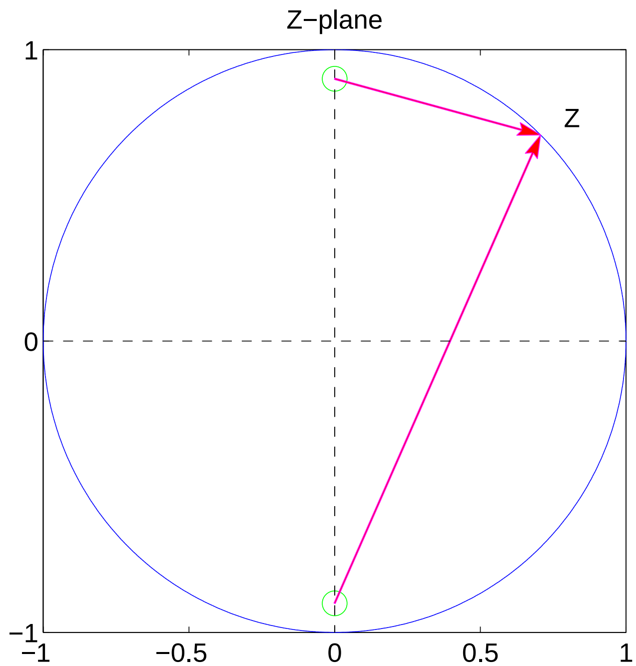

Figure 3.6: Two zero feedforward filter. $r = 0.9$, $|\omega_0| = \pi/2$

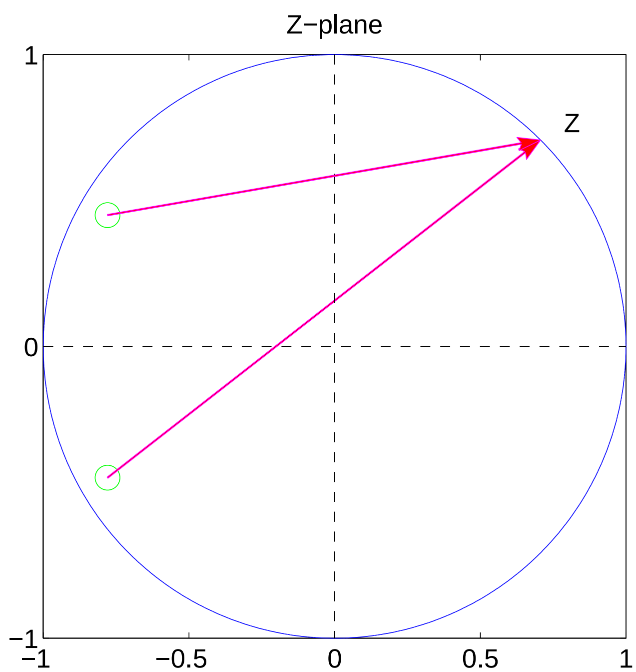

Figure 3.7: Two zero feedforward filter. $r = 0.9$, $|\omega_0| = 5\pi/6$

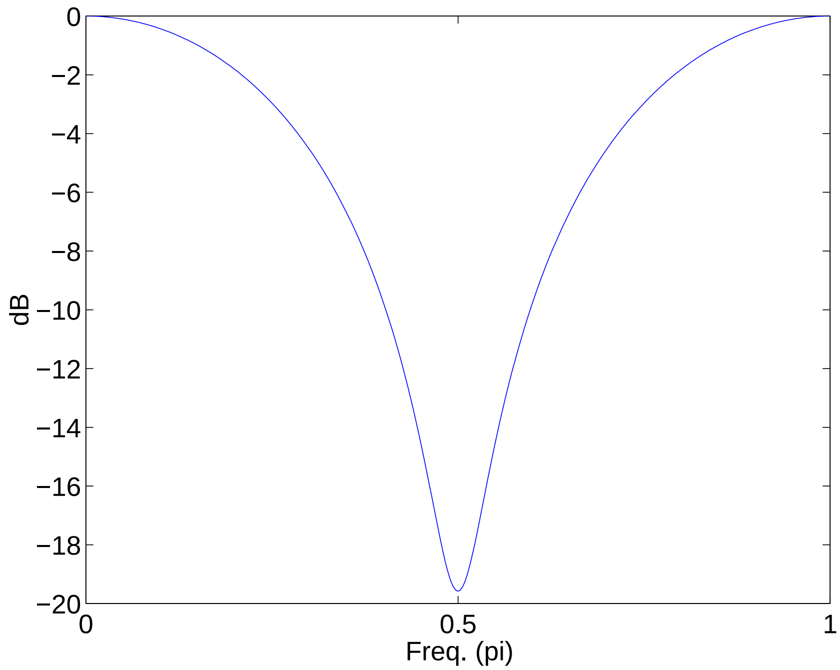

Figure 3.8: Magnitude of frequency response for the filter in figure 3.6.

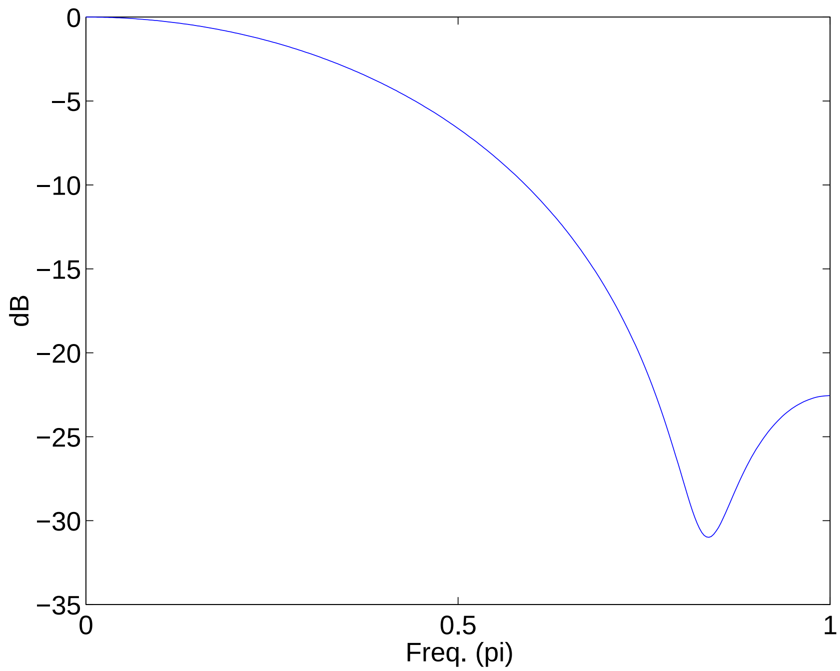

Figure 3.9: Magnitude of frequency response for the filter in figure 3.7.

$|H(\omega)|$ is the magnitude response and $\theta(\omega)$ is the phase response. The phase response can be computed as

$$\theta(\omega) = \arctan\left(\frac{\mathrm{Im}[H(\omega)]}{\mathrm{Re}[H(\omega)]}\right) \tag{3-44}$$

When the input signal is a phasor $e^{j\omega t}$, the filter's output is

$$y_t = |H(\omega)|e^{j(\omega t+\theta(\omega))} \tag{3-45}$$

So, what a filter does to a phasor (one frequency of the input) is to change the input's magnitude at that frequency by multiplying by $|H(\omega)|$ and shift its phase by the phase response $\theta(\omega)$.

Example 4: Let's examine the previous example (from the discussion of the z-plane):

$$y_t = x_t + a_1 x_{t-1} + a_2 x_{t-2}$$

Its transfer function is

$$\mathcal{H}(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$$

Therefore the frequency response is

$$\begin{aligned} H(\omega) &= 1 + a_1 e^{-j\omega} + a_2 e^{-2j\omega} \\ &= 1 + a_1(\cos\omega - j\sin\omega) + a_2(\cos 2\omega - j\sin 2\omega) \\ &= (1 + a_1\cos\omega + a_2\cos 2\omega) - j(a_1\sin\omega + a_2\sin 2\omega) \end{aligned}$$

When the magnitude response of a filter is plotted, the y axis scale is usually expressed in decibels (dB), a logarithmic scale. To obtain this, we note that the square of the magnitude response is

$$|H(\omega)|^2 = [1 + a_1\cos\omega + a_2\cos 2\omega]^2 + [a_1\sin\omega + a_2\sin 2\omega]^2 \tag{3-46}$$

A quantity is converted to dB by taking twenty times the logarithm (base ten). $|H(\omega)|$ can be converted to dB as

$$\begin{aligned} |H(\omega)|_{dB} &= 20\log_{10}|H(\omega)| = 10\log_{10}|H(\omega)|^2 \\ &= 10\log_{10}\left\{[1 + a_1\cos\omega + a_2\cos 2\omega]^2 + [a_1\sin\omega + a_2\sin 2\omega]^2\right\} \end{aligned}$$

Which is a trivial matter to compute for any value of ω (on a computer) for plotting purposes. The phase response is

$$\theta(\omega) = \arctan\left[\frac{-(a_1\sin\omega + a_2\sin 2\omega)}{1 + a_1\cos\omega + a_2\cos 2\omega}\right] \tag{3-47}$$

Consider a special case: $a_1 = 0$ and $a_2 = 1$. Substituting these values into equations (3-46) and (3-47), the squared magnitude becomes

$$\begin{aligned}
|H(\omega)|^2 &= (1+\cos 2\omega)^2 + \sin^2 2\omega \\
&= 1 + 2\cos 2\omega + \cos^2 2\omega + \sin^2 2\omega \\
&= 1 + 2\cos 2\omega + 1 \\
&= 2(1+\cos 2\omega) \\
&= 2(1 + 2\cos^2\omega - 1) \\
&= 4\cos^2\omega
\end{aligned}$$

The square root of this is

$$|H(\omega)| = 2|\cos\omega|$$

With the substitution, the phase response becomes

$$\theta(\omega) = \arctan\left(\frac{-\sin 2\omega}{1+\cos 2\omega}\right) \tag{3-48}$$

If we remember the double-angle formulae,

$$\begin{aligned}
\sin 2\omega &= \frac{2\tan\omega}{1+\tan^2\omega} \\
\cos 2\omega &= \frac{1-\tan^2\omega}{1+\tan^2\omega}
\end{aligned}$$

and substitute them into equation (3-48), we get

$$\begin{aligned}
\theta(\omega) &= \arctan\left(\frac{-2\tan\omega}{2}\right) \\
&= \arctan(-\tan\omega) \\
&= \begin{cases} -\omega & 0 \le \omega < \pi/2 \\ \pi - \omega & \pi/2 < \omega \le \pi \end{cases}
\end{aligned}$$

In other words *both* the magnitude and phase responses are functions of ω — they change the input in a frequency-dependent way. According to equation (3-39), the two zeros of the transfer function are at:

$$z_{1,2} = \frac{\pm\sqrt{-4}}{2} = \pm\frac{2j}{2} = \pm j$$

which are a pair of complex zeros on the imaginary axis. Using polar coordinates they are

$$z_{1,2} = e^{\pm j\pi/2} \tag{3-49}$$

with $r = 1$. The final result for the transfer function is

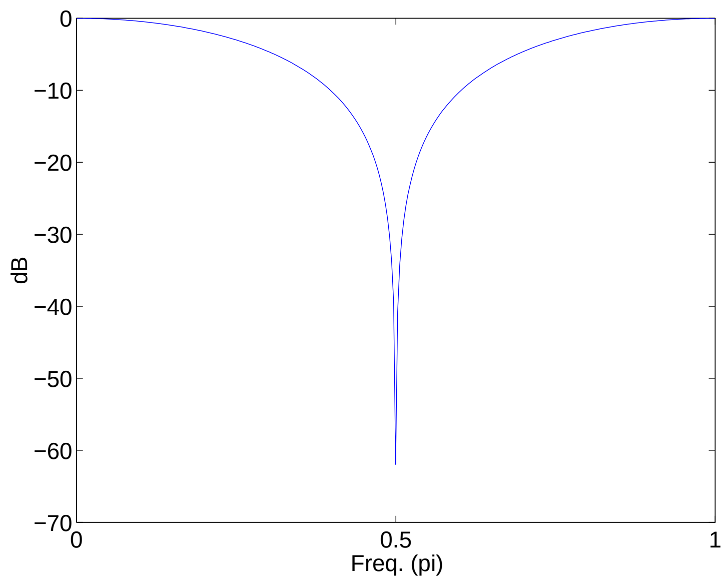

Figure 3.10: Magnitude response of two time delay feedforward filter, with $a_1 = 0, a_2 = 1$ in (3-51). Its zeros are at $e^{\pm j\pi/2}$.

$$H(\omega) = |H(\omega)|e^{j\theta(\omega)} = 2\cos\omega e^{-j\omega} \tag{3-50}$$

When the input is a phasor (a single frequency, $e^{j\omega t}$), the output is

$$y_t = 2\cos\omega e^{j\omega(t-1)} \tag{3-51}$$

The effect of the filter on the input signal is to delay it by one sampling interval (remember that for convenience we aren't writing T_s there) and to multiply it by $2\cos\omega$. Notice that the delay is independent of its frequency. When all the frequency components of a signal are delayed by an equal amount, say the filter has *no phase distortion*.

The magnitude response and phase response for this special case are shown in figure 3.10 and figure 3.11.

Self-Test Exercises

1. Prove $|z^2| = 1$ in equation (3-38). (*Popup Answer:* $|z^2| = |e^{2j\omega}|$. *From Euler's formula, this is* $|\cos 2\omega + j\sin 2\omega|$. *Since the magnitude of a complex number in rectangular form is the square root of the sum of the squares of its real and imaginary components, and* $\cos^2 2\omega + \sin^2 2\omega = 1$, $|z^2| = 1$.)

2. Starting with the factored magnitude response in equation (3-40), derive expressions for a_1 and a_2 in terms of z_1 and z_2. (*Popup Answer: The factored magnitude response is* $|\mathcal{H}(z)| = |(z - z_1)(z - z_2)|$. *Multiplying the two terms out yields* $|\mathcal{H}(z)| = |z^2 - zz_1 - zz_2 + z_1z_2| = |z^2 - (z_1 + z_2)z + z_1z_2|$. *Because* $|\mathcal{H}(z)| = |z^2 + a_1z + a_2|$, $a_1 = -(z_1 + z_2)$ *and* $a_2 = z_1z_2$.)

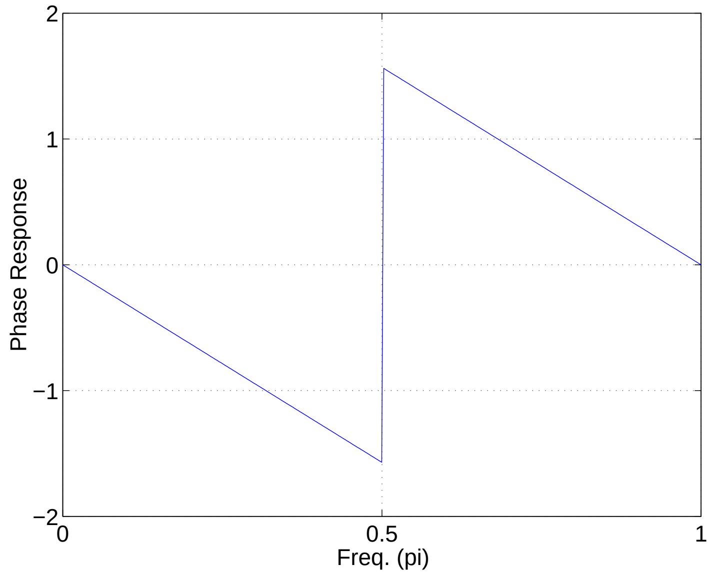

Figure 3.11: Phase response of the same frequency response of figure 3.10.

Implementing Digital Filters

Implementing feedforward digital filters is really quite straightforward. Let's first look at how we would implement the two time delay feedforward filter $y_t = x_t + a_1 x_{t-1} + a_2 x_{t-2}$, when $a_1 = 0, a_2 = 1$, in MATLAB. I'll present segments of a MATLAB script to do the computation and plot some results with my comments on it interspersed below:

```
% exp_ff_filter.m
% feedforward filter : y_t=x_t+a1x_{t-1}+a2x_{t-2}
% when a1=0, a2=1;

clf;
clear

% --- Generate an input signal ---
Fs=100;                 % Sampling frequency Fs Hz
N=100;                  % make N samples
Delay = 2;              % delay
dt = 1/Fs;              % time interval
t=(1:N)*dt;             % time axis
f1= 5;                  % input signal's frequency
f2= 25;
x=sin(2*pi*t*f1)+sin(2*pi*t*f2);      % input signal
```

At this point, I've generated a discrete signal `x` by first producing a vector `t` containing all the time points at which the sampling should occur (100 samples at 100Hz = 1 second). The signal has two frequency components: one at 5Hz and one at 25Hz (both well below the Nyquist frequency).

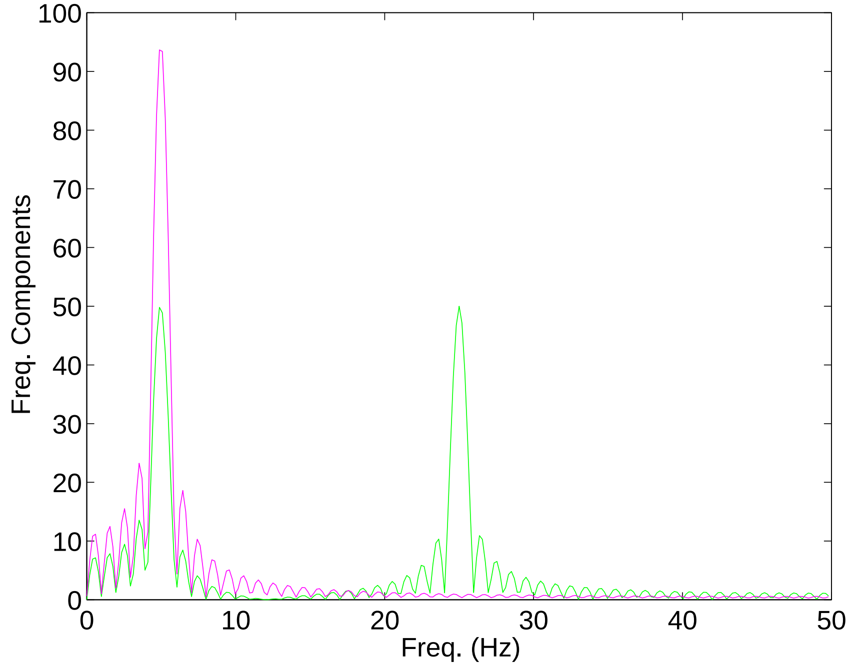

Figure 3.12: Figure 3.13's spectrum.

```
% --- Filtering ---
y = zeros(size(x));
y(1+Delay:N) = x(1+Delay:N) + x(1:N-Delay);
y = real(y);
```

In this particular example, I have the twin luxuries of taking advantage of MATLAB's built-in vector operations and being able to hold the entire input signal in a vector. I initialized `y` to zero to allocate memory at one time (MATLAB will automatically reallocate memory to expand or contract vectors and matrices, but that can be very time consuming). Because there is a two time step delay, I can't compute a value for `y(1)` or `y(2)`, so those stay zero. Since I'm only going to plot the magnitude response, I only keep the real part of `y`.

```
% --- plot magnitude of the input and output signals' spectra ---
spx=fft(x,512);          % original signal's fft
spy=fft(y,512);
fstep=(Fs/2)/256:        % frequency step
f=(0:255)*fstep;         % frequency axis
plot(f,abs(spx(1:256)'), 'g', f,abs(spy(1:256)'), 'm');
                         %plot spx and spy in the same plot
xlabel('Freq. (Hz)', 'Fontsize', 16);
ylabel(Freq. Components, 'Fontsize', 16);
set(gca, 'Fontsize', 16);
input('Press key to continue')
```

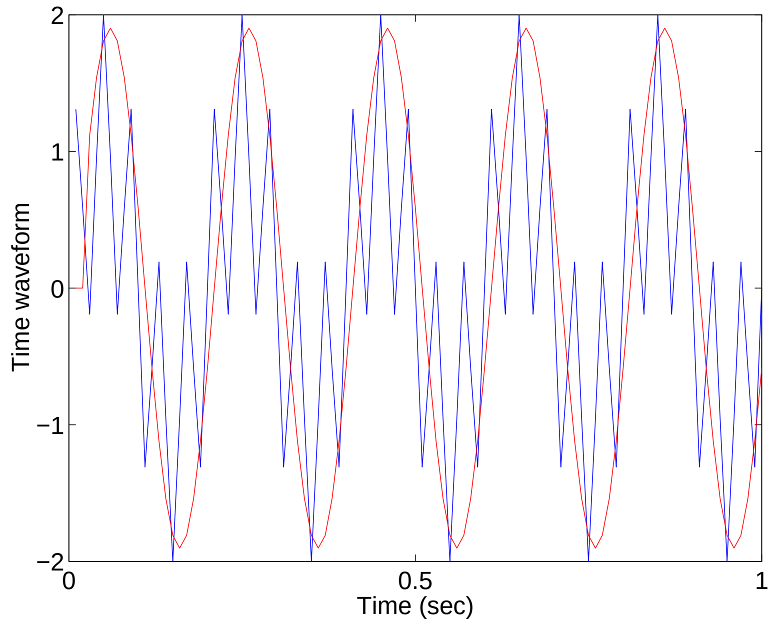

Figure 3.13: Two frequency component sine wave and its filtered version, using the feedforward filter $y_t = x_t + x_{t-2}$, $f_1 = 5Hz$ and $f_2 = 25Hz$. Zeros are at $z0 = 0.99e^{\pm j\pi/2}$.

I want to see what happened to the signal's frequency components, and so I use the built-in MATLAB function `fft()`. I'll talk about the Fourier transform and FFT in lesson 5. See the MATLAB manuals (especially the graphics guide) to see what the `plot()` and `set()` are doing. Figure 3.12 is the plot this produces.

```
% --- Plot orignal and filtered signal ---
plot(t, x, 'b', t, y, 'r' );
xlabel('Time (sec)', 'FontSize', 16);
ylabel('Time waveform', 'FontSize', 16);
set(gca, 'FontSize', 16);
```

And finally, I plot the input and output. They're shown in figure 3.13.

How would one implement this in a language like C, C++, Java, etc? Let's not worry about the plotting issue: that would certainly have to be dealt with, but it would involve either getting a graphics library, writing one's own, or using an external plotting package (ideally, one targeted at scientific and engineering applications, rather than one written for business). All the elementary vector and matrix operations provided by MATLAB can be implemented as loops. I will discuss the `fft()` function in lesson 5. The only remaining issue is that of keeping the entire input and output signal in memory.

In general, it is not possible to keep the entire input and output signal in memory. In fact, many (if not most) digital signal processing applications involve real-time processing, so the computer system really needs to be viewed as just a stage in a processing pipeline, with the input flowing in and the output flowing out (just like the block diagrams used to illustrate filters). We can therefore only devote a relatively small amount of memory to hold the part of the input and output needed for the current computation. If n delayed samples of the input are needed to compute each output value, we will need storage for n input samples. For a feedforward filter, there is no need to save output values; they can be sent out as soon as they are computed.

What abstract data type (ADT) should hold the delayed inputs? (*Popup Answer: a queue.*) As each input sample comes in, it displaces the oldest input sample from our buffer of n values. This is clearly a FIFO ADT — a *queue*. As you should remember, a queue is very efficiently and simply implemented in an array, especially when the queue size is fixed. Furthermore, you should remember how to treat the array as though it were circular using indexing mod the queue size.

Assignment 3

1. Use the same method as textbook equation (7.6) to recompute the results (3-50) and (3-51) of Example 4. It should be easier.

2. For the filter whose magnitude response is described in (3-40), plot the location of a pair of complex conjugate zeros at $r = 0.8$ and $|\omega_0| = \pi/4$. Using MATLAB, compute and plot the filter's magnitude response. Submit your code with your figures.

3. Text book chapter 4 problem 4.4.

4. Text book chapter 4 problem 4.9, but using MATLAB to plot out the results. Submit your code with your results.

5. Text book chapter 4 problem 4.10.