4. Digital Filter Structures

4.1 Introduction

Previous sections have covered the basic tools for analysing discrete-time signals and systems, including time-domain representation, z-domain representation, and frequency-domain analysis. In the case of discrete-time systems (digital filters), we have studied the difference equation description of a filter, from which the z-transform (transfer function) can readily be obtained. The transfer function, in its pole-zero form, provides us with a number of insights into system behaviour, including stability of the system, as well as the general form of the impulse response. From the transfer function, the system frequency response can easily be determined. Alternatively, the frequency response can be obtained directly from the system impulse response, through application of the Fourier transform, though this is impractical in the case of filters with “long” impulse response filters.

In this section, we continue our study of digital filters. Now that we have covered the basic tools for obtaining the frequency response of a system, we will go back and look some more at the effect that pole-zero placement has on the frequency response (in Section 2 we looked at the effect of pole-zero placement on impulse response). In addition, we will cover the various “architectures” for implementing digital filters, including high order filters (though the actual *design* of these filters will be covered later in the course). Finally, we will look at some “special” cases of second-order digital filters that crop up frequently in DSP.

4.2 Effect of Pole-Zero Placement on Frequency Response

Suppose we have a system whose transfer function is represented as:

$$H(z) = \frac{\sum_{k=0}^{K} a_k z^{-k}}{1+\sum_{n=1}^{N} b_n z^{-n}} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \ldots + a_K z^{-K}}{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + \ldots + b_N z^{-N}}$$

If we calculate the poles and zeros (through factorisation of the numerator and denominator polynomials), we can rewrite the transfer function as:

$$H(z) = z^{N-K} \frac{a_0 (z-z_1)(z-z_2)(z-z_3)\ldots(z-z_K)}{(z-p_1)(z-p_2)(z-p_3)\ldots(z-p_N)}$$

We have already established that the frequency response of a system may be obtained from the transfer function, by making the substitution z = $e^{j\theta}$. If we make this substitution in the “factored” form of the transfer function, we obtain:

$$H(\theta) = e^{j(N-K)\theta} \frac{a_0 (e^{j\theta}-z_1)(e^{j\theta}-z_2).....(e^{j\theta}-z_K)}{(e^{j\theta}-p_1)(e^{j\theta}-p_2).....(e^{j\theta}-p_N)}$$

If we now calculate the magnitude of $H(\theta)$, we obtain:

$$\left|H(\theta)\right| = \left| e^{j(N-K)\theta} \frac{a_0 (e^{j\theta}-z_1)(e^{j\theta}-z_2).....(e^{j\theta}-z_K)}{(e^{j\theta}-p_1)(e^{j\theta}-p_2).....(e^{j\theta}-p_K)} \right|$$

$$|H(\theta)| = a_0 \left|e^{j(N-K)\theta}\right| \frac{\left|e^{j\theta} - z_1\right|\left|e^{j\theta} - z_2\right| \ldots\ldots \left|e^{j\theta} - z_K\right|}{\left|e^{j\theta} - p_1\right|\left|e^{j\theta} - p_2\right| \ldots\ldots \left|e^{j\theta} - p_N\right|}$$

$$|H(\theta)| = a_0 \frac{\prod_{i=1}^{K} \left|e^{j\theta} - z_i\right|}{\prod_{k=1}^{N} \left|e^{j\theta} - p_k\right|}$$

Note that $e^{j\theta}$ represents a point on the unit circle, at angle θ with respect to the positive real axis, while z_i and p_k represent the system zeros and poles which may be located anywhere in the z-plane (not necessarily on the unit circle, though of course for a stable system, the points p_k must lie *inside* the unit circle). Therefore, the terms $(e^{j\theta} - p_k)$ and $(e^{j\theta} - z_i)$ represent vectors in the z-plane, drawn from $e^{j\theta}$ to the corresponding poles and zeros, and the terms $\left|e^{j\theta} - p_k\right|$ and $\left|e^{j\theta} - z_i\right|$ represent the magnitude of these vectors. Hence, the magnitude response of the filter at a given frequency θ may be obtained graphically, and is equal to a_0 times the product of the distances from each zero to the point $e^{j\theta}$ divided by the product of the distance from each pole to the point $e^{j\theta}$. This is exactly analogous to the manner in which the frequency response of continuous-time systems may be obtained (except in that case, the vectors ran between the poles and zeros, and points on the imaginary axis). A similar argument holds for the phase angle, and results in the observation that the phase response of the filter at a given frequency θ is equal to the sum of the angles of the vectors from the zeros to the point $e^{j\theta}$ minus the sum of the angles of the vectors from the poles to the point $e^{j\theta}$. This is illustrated graphically for the arbitrary second-order system whose pole-zero map is shown in Figure 4.1.

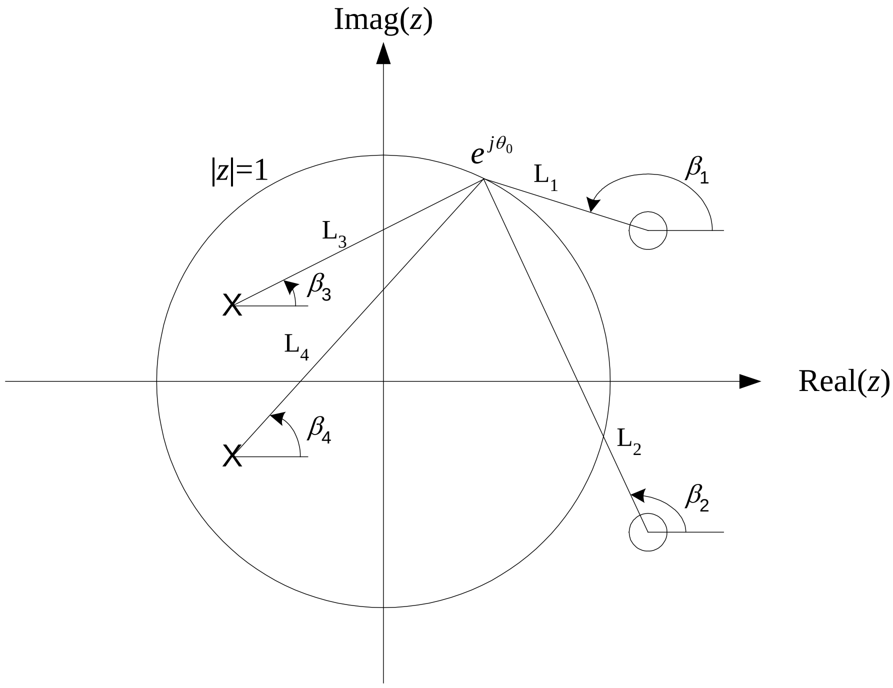

Figure 4.1. Illustration of “graphical” interpretation of frequency response for an arbitrary second-order system.

In the diagram above, the magnitude response at the arbitrarily chosen frequency θ_0 is given by:

$$|H(\theta_0)| = \frac{L_1 L_2}{L_3 L_4}$$

while the phase response is given by:

$$\phi(\theta_0) = \beta_1 + \beta_2 - \beta_3 - \beta_4$$

This graphical interpretation of the frequency response illustrates more clearly the effect that poles and zeros have on the frequency response, particularly around frequencies that correspond to the poles and zeros; these frequencies are often referred to as the "natural" frequencies of the system. For example, the following points can be made:

- The effect of a pole or zero on the frequency response increases the closer it is to the unit circle. In the extreme case of a zero actually on the unit circle, the magnitude response at the zero frequency is equal to zero, while the phase response jumps by π radians. Alternatively, if a pole is located on the unit circle, the magnitude response equals infinity, and again, the phase response jumps by π radians.
- If all of the zeros of a system are inside the unit circle, the system is said to be minimum phase, because zeros inside the unit circle contribute less to the phase response than zeros outside the unit circle. By "reflecting" the zeros of a filter in the unit circle, we can change the phase characteristics of the system, while leaving the magnitude response unchanged. Reflection means that a zero at $z = re^{j\theta_0}$ becomes a zero at $z = \left(1/r\right)e^{j\theta_0}$.

Exercise 4.1

Plot the magnitude and phase responses, and hence find the group delay, of the filer with zeros and poles at the following locations:

$z_1 = 1.2 + j0.7, z_2 = 1.2 - j0.7$

$p_1 = 0.7 + j0.4, p_2 = 0.7 - j0.4$

Do the same calculations for the case where the zeros are reflected in the unit circle.

- An all-pass filter is a filter for which $|H(\theta)| = 1$, for all values of θ. This is achieved by ensuring that all poles have a corresponding zero, located at the reflection of the pole in the unit circle. Note, however, that the phase response of such a filter may be such that significant distortion can still occur. For example, Figure 3.8(c) shows the effect of non-linear phase on a given signal; the filter used for Figure 3.8(c) is actually an all-pass filter.

Exercise 4.2

For the filter in Exercise 4.1, choose zeros such that the filter becomes an all-pass filter, and calculate and plot the frequency response.

- A linear phase filter is one that contains only zeros, where the zeros occur in pairs that are reflected in the unit circle (there may also be zeros on the unit circle itself). An all-zero system is an FIR filter.

Exercise 4.3

For the system in Exercise 4.1, replace the poles with zeros such that the resulting filter has linear phase; calculate and plot the frequency response.

4.3 Non-Recursive Filter Structures

As noted above, the transfer function of a system may be expressed as:

$$H(z)=\frac{\sum_{k=0}^{K}a_k z^{-k}}{1+\sum_{n=1}^{N}b_n z^{-n}}=\frac{a_0+a_1 z^{-1}+a_2 z^{-2}+a_3 z^{-3}+\ldots+a_K z^{-K}}{1+b_1 z^{-1}+b_2 z^{-2}+b_3 z^{-3}+\ldots+b_N z^{-N}}$$

However, there are many ways in which such a system may actually be implemented. In the simpler case of an FIR filter (N=0), the usual filter structure used is a so-called “transversal” filter, which is illustrated in Figure 4.2:

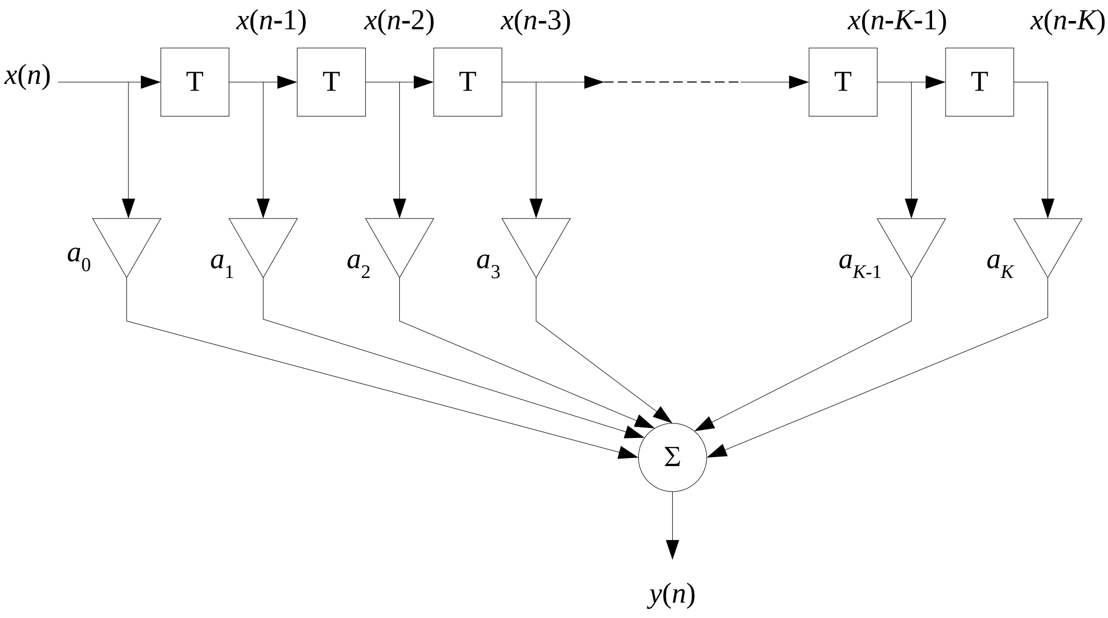

Figure 4.2. Transversal filter structure for implementing an FIR filter.

This architecture is also referred to as a “tapped delay line”. In hardware terms, each of the delay blocks in the transversal filter could be implemented by means of a multi-bit shift register, or perhaps a memory location in an embedded RAM, while the gain blocks would be implemented by a multiplier (perhaps a single multiplier shared among all of the required coefficients).

For IIR filters, there are a number of alternative architectures that can be used; these will be considered in the next sub-section.

4.4 Recursive Filter Structures

4.4.1 Introduction

A number of trade-offs exist in choosing a specific IIR filter structure. For example, the various structures differ in terms of their “processing” requirements (number of memory elements required, speed of multipliers etc.), as well as in their sensitivity to quantisation of coefficients and data (remember that in DSP implementations, numbers will generally be represented with a limited number of bits).

A general IIR filter transfer function may be written as:

$$H(z) = \frac{\sum_{k=0}^{K} a_k z^{-k}}{1+\sum_{n=1}^{N} b_n z^{-n}} = \underbrace{\sum_{k=0}^{K} a_k z^{-k}}_{\text{zeros}} \underbrace{\frac{1}{1+\sum_{n=1}^{N} b_n z^{-n}}}_{\text{poles}}$$

$$= \underbrace{\frac{1}{1+\sum_{n=1}^{N} b_n z^{-n}}}_{\text{poles}} \underbrace{\sum_{k=0}^{K} a_k z^{-k}}_{\text{zeros}}$$

These “alternative” forms of the transfer function are functionally equivalent, however, in one case, the zeros are implemented first, followed by the poles, while in the other case, the reverse is true.

4.4.2 Direct Form I

This architecture is equivalent to implementing the zeros first, and is illustrated in Figure 4.3 for the simple case of a second-order system with the following transfer function:

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}}$$

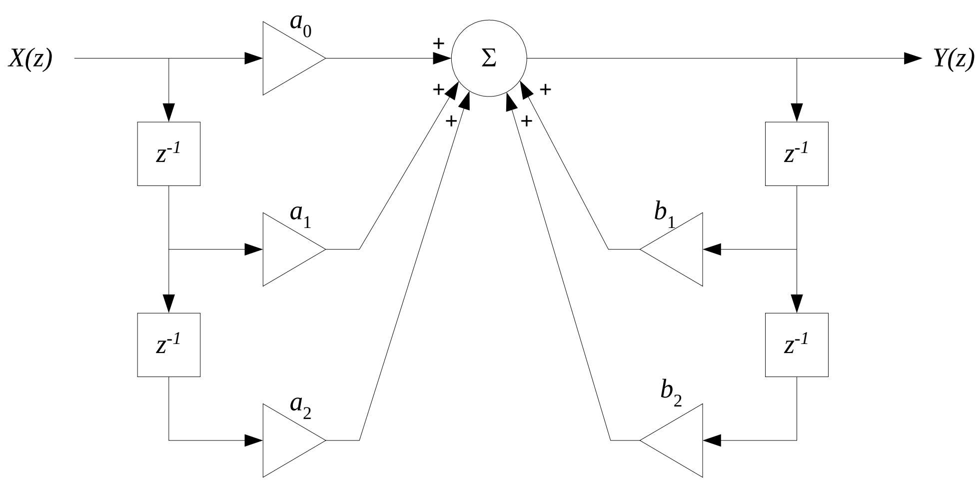

Figure 4.3. Direct Form I implementation of an IIR filter.

As can be seen from the diagram, this structure requires four memory locations to hold the previous values of the input and output (the "state" of the filter), as well as a single "accumulator" to add up the various products that are generated in the calculation of the output.

4.4.3 Direct Form II

This architecture is illustrated in Figure 4.4:

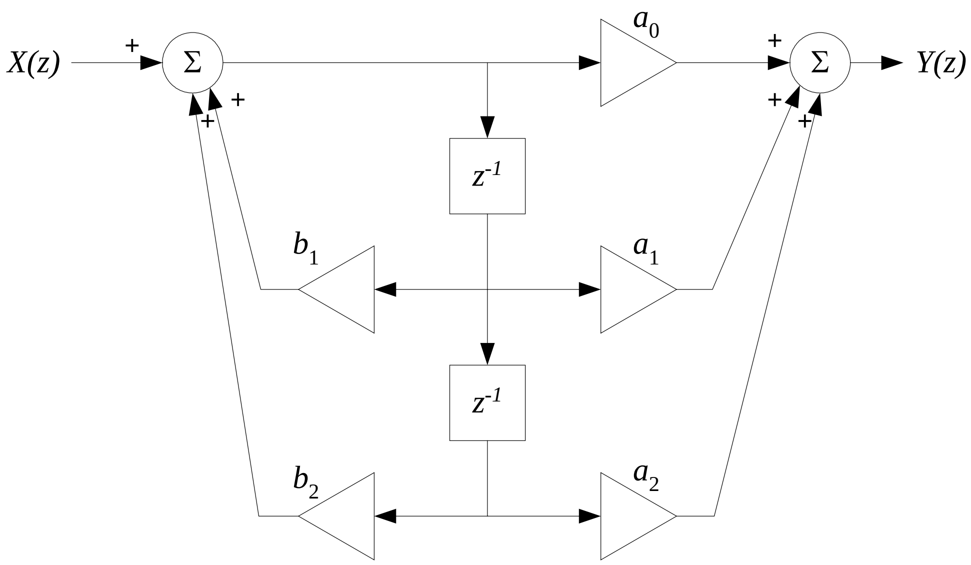

Figure 4.4. Direct Form II implementation of an IIR filter.

In this case, only two memory elements are required, but two "accumulators" are needed. Note that in this case, the orders of the numerator and denominator are the same; this results in a system with the minimum number of memory elements. Such a system is called a *canonic* network.

4.4.4 Transpose Form

The transpose form of an IIR filter is shown in Figure 4.5; in this case, three "accumulators" and two memory elements are needed.

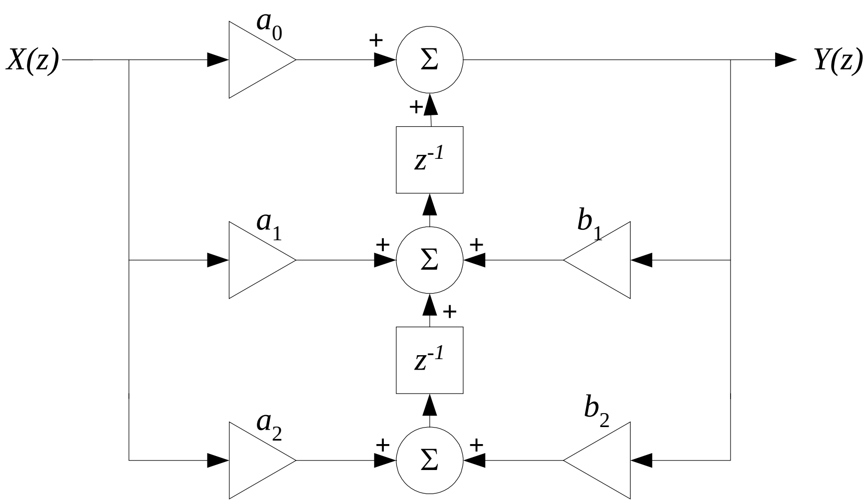

Figure 4.5. Transpose form implementation of an IIR filter.

4.5 Implementing High Order IIR Filters

4.5.1 Introduction

The above example was a relatively simple filter, with an order of 2. However, in practice, digital filters may be much more complex than this case. Such filters are generally not implemented using a single "stage" as illustrated above; instead they are decomposed into a number of second-order and first-order (if required) stages that are connected either in cascade or in parallel. "Generic" first-order and second-order sections are represented by the following transfer functions:

$$H(z) = \frac{c_0 + c_1 z^{-1}}{1 + d_1 z^{-1}}$$

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

(sometimes, the signs of the denominator coefficients are negative)

There are a number of reasons for carrying out such decomposition of a higher-order filter:

- Filters are generally much easier to analyse and design when they are represented as second-order or first-order sections, because it is much easier to see how these simpler sections influence system characteristics like impulse response and frequency response.
- From an implementation point of view, a DSP designer may have already designed a software routine or VHDL block to implement a first- or second-order filter, so this software or hardware can be reused to implement higher-order filters, thus reducing the design and (especially) test efforts.
- When implemented using limited numbers of bits for coefficients and data, high-order filters (especially with feedback) are usually much more sensitive to quantisation effects when implemented with a single section, than when implemented with multiple simpler sections.

Note that the individual second-order sections may be implemented using one of the structures discussed in Section 4.4.

4.5.2 Cascade Structure

In the cascade structure, a filter transfer function is represented as follows:

$$H(z) = \prod_{i=1}^{L} H_i(z) = H_1(z)H_2(z)\ldots H_L(z)$$

where $H_i(z)$ is either a second-order of first-order section. The overall order of the filter can be easily ascertained from the number and type of the individual sections (a first-order section is needed only if the "original" filter order is odd). Determination of the cascade form of a filter requires factorisation of the numerator and denominator of the "original" high-order transfer function into second-order and first-order sections.

Example 4.1

Suppose we want to implement the following third-order filter in cascade form:

$$H(z) = \frac{23 + 40z^{-1} + 36z^{-2} + 19z^{-3}}{10 + 9z^{-1} + 8z^{-2} + 3z^{-3}}$$

This can be factorised to obtain:

$$\begin{aligned} H(z) &= \frac{(1 + z^{-1})(23 + 17z^{-1} + 19z^{-2})}{(2 + z^{-1})(5 + 2z^{-1} + 3z^{-2})} \\ &= \frac{(1 + z^{-1})}{(2 + z^{-1})} \frac{(23 + 17z^{-1} + 19z^{-2})}{(5 + 2z^{-1} + 3z^{-2})} = \frac{(0.5 + 0.5z^{-1})}{(1 + 0.5z^{-1})} \frac{(4.6 + 3.4z^{-1} + 3.8z^{-2})}{(1 + 0.4z^{-1} + 0.6z^{-2})} \end{aligned}$$

$$\text{Hence, } H(z) = H_1(z)H_2(z), \;\; H_1(z) = \frac{(0.5 + 0.5z^{-1})}{(1 + 0.5z^{-1})}, \;\; H_2(z) = \frac{(4.6 + 3.4z^{-1} + 3.8z^{-2})}{(1 + 0.4z^{-1} + 0.6z^{-2})}$$

Note that the individual transfer functions have been scaled so that the "b_0" coefficient in the denominator is equal to 1.

4.5.3 Parallel Structure

In this case, the transfer function is implemented as follows:

$$H(z) = Gz^{-k} + C + \sum_{i=1}^{L} H_i(z) = Gz^{-k} + C + H_1(z) + H_2(z) + H_3(z) + ... + H_L(z)$$

(this is a more generals case of the decomposition we studied in Section 2, when obtaining inverse z-Transforms)

Again, the individual $H_i(z)$ sections are either first-order or second order. The first term Gz^{-k} will exist only if the order of the numerator is greater than the order of the denominator. In this case, the numerator should be divided by the denominator to obtain Gz^{-k}. The remainder will be such that the order of the numerator will be less than or equal to the order of the denominator. In this case, the remainder can be expanded into parallel form using the method of partial fractions. The C term will be zero if the order of the numerator is less than the order of the denominator. For example, suppose we want to obtain the parallel form of the third-order transfer function from Example 4.1:

$$H(z) = \frac{23 + 40z^{-1} + 36z^{-2} + 19z^{-3}}{10 + 9z^{-1} + 8z^{-2} + 3z^{-3}}$$

The orders of the numerator and denominator are equal, so we know that the transfer function will be of the form:

$$H(z) = C_0 + H_1(z) + H_2(z)$$

where $H_1(z)$ is first-order and $H_2(z)$ is second-order. The C_0 term is obtained from the ratio of the two terms containing the order of both the numerator and denominator polynomials (i.e. 3):

$$C_0 = \frac{19}{3}$$

In order to represent the transfer function in the form of partial fractions, we need to factorise the denominator (already done from the previous example):

$$H(z)=\frac{23+40z^{-1}+36z^{-2}+19z^{-3}}{10+9z^{-1}+8z^{-2}+3z^{-3}}=\frac{23+40z^{-1}+36z^{-2}+19z^{-3}}{(2+z^{-1})(5+2z^{-1}+3z^{-2})}$$

Hence, we can write $H(z)$ in partial fraction form as:

$$H(z)=\frac{19}{3}+H_1(z)+H_2(z)=\frac{19}{3}+\frac{A}{2+z^{-1}}+\frac{B+Cz^{-1}}{5+2z^{-1}+3z^{-2}}$$

$$=\frac{19(2+z^{-1})(5+2z^{-1}+3z^{-2})+A(3)(5+2z^{-1}+3z^{-2})+(B+Cz^{-1})(3)(2+z^{-1})}{3(2+z^{-1})(5+2z^{-1}+3z^{-2})}$$

$$H(z)=\frac{(190+15A+6B)+z^{-1}(171+6A+6C+3B)+z^{-2}(152+9A+3C)+57z^{-3}}{3(2+z^{-1})(5+2z^{-1}+3z^{-2})}$$

Comparing this with the original transfer function, we obtain:

$$H(z)=\frac{23+40z^{-1}+36z^{-2}+19z^{-3}}{(2+z^{-1})(5+2z^{-1}+3z^{-2})}=\frac{69+120z^{-1}+108z^{-2}+57z^{-3}}{3(2+z^{-1})(5+2z^{-1}+3z^{-2})}$$

This gives us three simultaneous equations:

$$190+15A+6B=69$$

$$171+6A+3B+6C=120$$

$$152+9A+3C=108$$

The solution of these equation yields:

A = -5, B = -23/3 and C = 1/3

Thus, the transfer function can be written in parallel form as:

$$\begin{aligned}H(z)\quad&=\frac{19}{3}+H_1(z)+H_2(z)\\&=\frac{19}{3}-\frac{5}{2+z^{-1}}-\frac{23-z^{-1}}{3(5+2z^{-1}+3z^{-2})}\\&=\frac{19}{3}-\frac{2.5}{1+0.5z^{-1}}-\frac{\frac{23}{15}-\frac{1}{15}z^{-1}}{1+0.4z^{-1}+0.6z^{-2}}\end{aligned}$$

Note: Expressing a high-order transfer function in cascade or parallel form can be quite tedious to do "on paper", however, each method essentially involves manipulation of polynomials (factorisation etc.), so any computer-based tools which assist in this procedure can be used. For example, the Matlab function "*roots*" can be used to obtain the roots of a polynomial, which can then be used to obtain the cascade form (complex conjugate pairs of roots should be "combined" to give second-order sections with real coefficients). Alternatively, the function "*residuez*" may be used to obtain a partial fraction expansion of a transfer function (again, complex conjugate terms may be combined to give second-order terms with real coefficients).

Exercise 4.4

Use Matlab to obtain the cascade and parallel implementations of the third-order transfer function used in the above examples.

4.6 Resonators

4.6.1 Introduction

In the previous sub-sections, we have examined some issues relating to implementation of digital filters (without being too concerned about how to design these filters, which will be covered later in the course). In particular, we have seen how a high-order filter may be decomposed into a parallel or cascade set of second-order (and first-order) sections (usually IIR). At the same time, a second-order filter is a useful “filter” in its own right (albeit a fairly simple one), and can be readily designed using a few simple equations.

Hence, the second-order IIR filter could be regarded as a fundamental building block of complex DSP systems, and worthy of some more study – this is the subject of this sub-section. As we have seen, a second-order system with real coefficients in the denominator often corresponds to a pair of complex conjugate poles, and these poles cause a “peak”, or “resonance” in the magnitude response of the filter (as is the case for continuous-time systems as well). Hence, second-order IIR filters with complex conjugate poles are often referred to as *resonators*.

4.6.2 Resonator Design

As noted above, a resonator is a filter with a distinct peak in its magnitude response, at a particular frequency close to the pole frequency, θ_0 (often referred to as the “centre frequency”). To obtain the equations for designing a resonator, we write the poles in polar form as:

$$p_1 = re^{j\theta_0} \text{ and } p_2 = re^{-j\theta_0}$$

In addition to the pair of complex conjugate poles, resonators also have a double zero at the origin; however, these zeros have no effect on the magnitude response of the filter, though they do affect the phase response.

Hence, the filter transfer function can be written as:

$$\begin{aligned}
H(z) &= \frac{z^2}{(z - re^{j\theta_0})(z - re^{-j\theta_0})} \\
&= \frac{z^2}{z^2 - r(e^{j\theta_0} + e^{-j\theta_0})z + r^2} \\
&= \frac{z^2}{z^2 - 2r\cos(\theta_0)z + r^2}
\end{aligned}$$

Dividing numerator and denominator by the highest power of z yields:

$$\begin{aligned} H(z) &= \frac{1}{1-2r\cos(\theta_0)z^{-1}+r^2z^{-2}} \\ &= \frac{1}{1+b_1z^{-1}+b_2z^{-2}} \end{aligned}$$

Hence, given knowledge of the filter pole frequency and radius, we can calculate the coefficients using the following equations:

$$b_2 = r^2$$
$$b_1 = -2r\cos(\theta_0) = -2\sqrt{b_2}\cos(\theta_0)$$

Note that for stability, the pole radius r must be less than 1. As noted before, the closer the poles are to the unit circle, the narrower and higher is the corresponding peak in the magnitude response; another way of looking at this is that the "bandwidth" of the peak gets smaller. Hence, r controls the "sharpness" of the peak. An empirical equation that is often used to choose a value for r is as follows:

$$r \approx 1 - \frac{\Delta f}{f_{samp}}\pi$$

where Δf is the desired "bandwidth" of the resonance, and f_{samp} is the sampling frequency. The transfer function is often scaled by the term $1+b_1+b_2$, to ensure that the gain of the filter at DC (0 Hz) is equal to 1. Hence, the final transfer function can be written as:

$$H(z) = \frac{1+b_1+b_2}{1+b_1z^{-1}+b_2z^{-2}}$$

and the difference equation is:

$$y(n) = (1+b_1+b_2)x(n) - b_1y(n-1) - b_2y(n-2)$$

Exercise 4.5

Draw the block diagram of a resonator.

Exercise 4.6

Show how the presence or absence of zeros at the origin of a resonator influences the frequency response.

Example 4.2

Suppose we want to design a resonator with the following specification:

- Centre (pole) frequency = 1200 Hz
- Bandwidth = 75 Hz
- Sampling rate = 9.6 kHz
- DC gain = 1

The first step is to calculate the pole frequency (in radians) and the pole radius:

$$\theta_0 = 2\pi \frac{f_0}{f_{samp}} = 2\pi \frac{1200}{9600} = \frac{\pi}{4}$$

$$r \approx 1 - \frac{\Delta f}{f_{samp}}\pi = 1 - \frac{75}{9600}\pi = 0.9754$$

Then, the coefficients are calculated as follows:

$$b_2 = r^2 = 0.9515$$

$$b_1 = -2r\cos(\theta_0) = -2(0.9754)\cos\left(\frac{\pi}{4}\right) = -1.3794$$

$$1 + b_1 + b_2 = 1 - 1.3794 + 0.9515 = 0.5721$$

Hence, the transfer function and difference equation can be written as:

$$H(z) = \frac{0.5721}{1 - 1.3794z^{-1} + 0.9515z^{-2}}$$

$$y(n) = 0.5721x(n) + 1.3794y(n-1) - 0.9515y(n-2)$$

The frequency response and pole-zero map of this filter are shown in Figures 4.6 and 4.7 respectively:

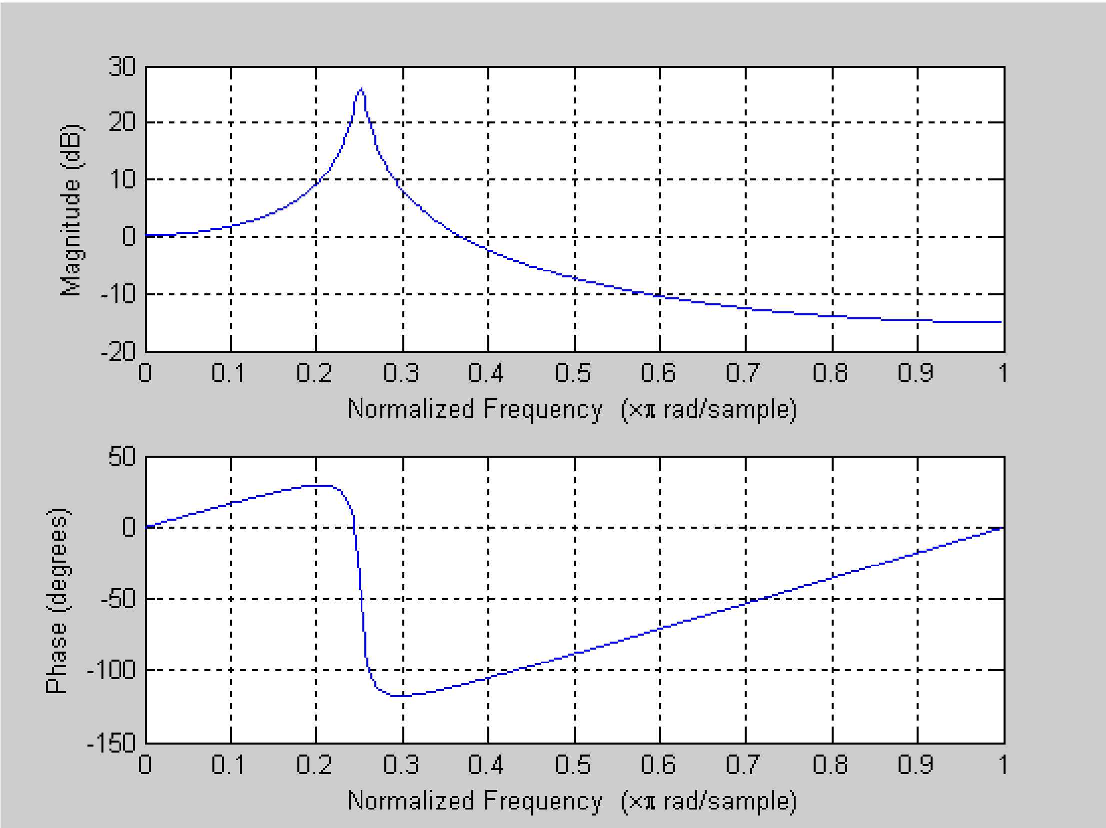

Figure 4.6. Frequency response of resonator from Example 4.2.

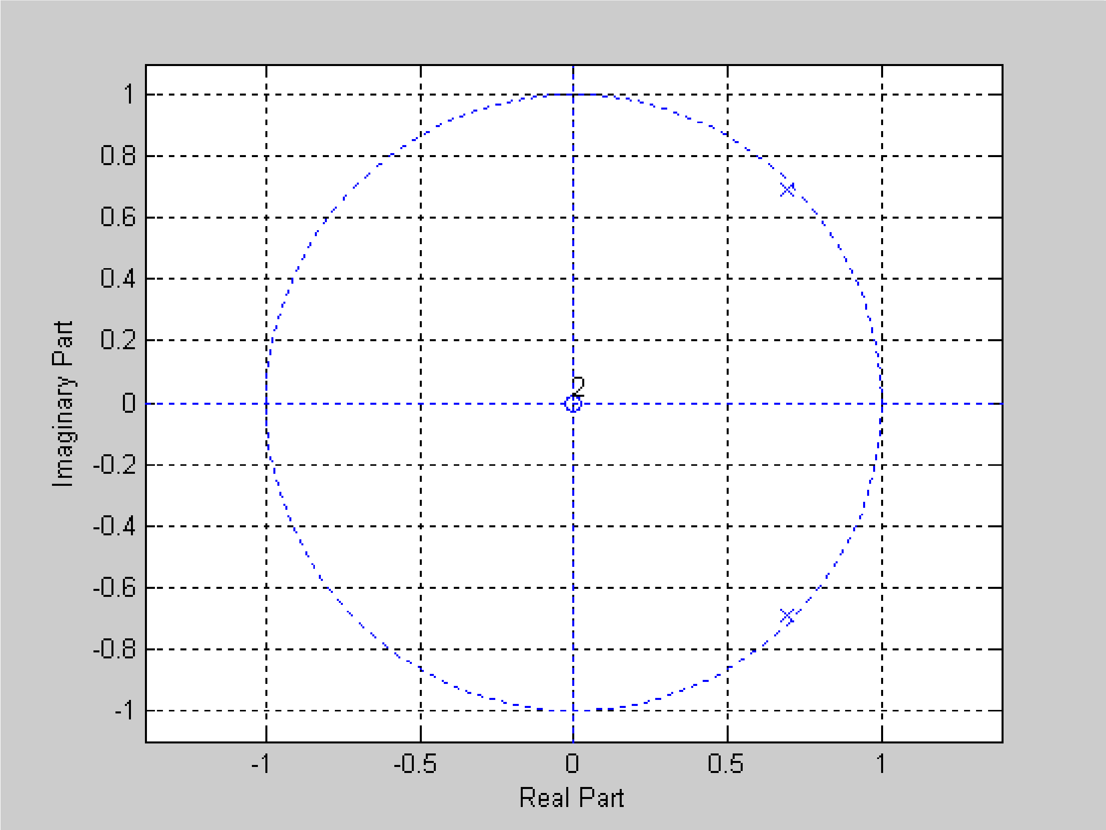

Figure 4.7. Pole-zero map of resonator from Example 4.2.

4.7 Stability of Second-Order Filter

Sometimes it is necessary to determine the stability of a second-order filter from the coefficients b_1 and b_2. This can also be useful if, for example, we have a high-order filter that we have factored into a set of second-order (and possibly first-order) sections. In essence, we need to check the locations of the poles in each section.

We already know that for stability, the poles of a second-order filter must lie inside the unit circle, i.e. the pole radius must be less than 1. This implies that coefficient b_2 must be less than 1, since $b_2 = r^2$. However, this strictly applies only in the case of complex conjugate poles. Let's look at the different possibilities that may arise.

Given a transfer function:

$$H(z) = \frac{1 + b_1 + b_2}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

we can easily determine the poles in terms of the coefficients by getting the roots of the denominator polynomial, i.e. the poles will be located at the following points in the z-plane:

$$z = \frac{-b_1 \pm \sqrt{b_1^2 - 4b_2}}{2}$$

The poles will form a complex conjugate pair if:

$$b_1^2 - 4b_2 < 0$$

and this is the situation we have examined so far with the "resonator".

However, the possibility also exists that the same transfer function will have a pair of real poles – in this case, however, it will not actually be a "resonator". For a stable filter with real poles, the same requirement that the poles lie inside the unit circle still applies, so the real poles must have values between +1 and –1:

$$-1 < \frac{-b_1 \pm \sqrt{b_1^2 - 4b_2}}{2} < 1$$

$$\Rightarrow -2 + b_1 < \pm\sqrt{b_1^2 - 4b_2} < 2 + b_1$$

$$\Rightarrow -2 + b_1 < -\sqrt{b_1^2 - 4b_2} \text{ and } \sqrt{b_1^2 - 4b_2} < 2 + b_1$$

$$\Rightarrow (-2 + b_1)^2 > b_1^2 - 4b_2 \text{ and } b_1^2 - 4b_2 < (2 + b_1)^2$$

$$\therefore b_1 - b_2 - 1 < 0 \text{ and } b_1 + b_2 + 1 > 0$$

These equations define two lines in the "coefficient plane" whose co-ordinates are b_1 and b_2. The equations of these lines are:

$$b_2 = b_1 - 1 \qquad \text{and} \qquad b_2 = -b_1 - 1$$

The region formed by the intersection between these two equations and the equation $b_1^2 - 4b_2 = 0$ (which is the equation for a parabola) defines the region of allowable combinations of the coefficients b_1 and b_2 in order to ensure stability – from it's shape, this region is called the "stability triangle", and is illustrated in Figure 4.8. A second-order filter is stable only if the point (b_1, b_2) lies inside this triangle.

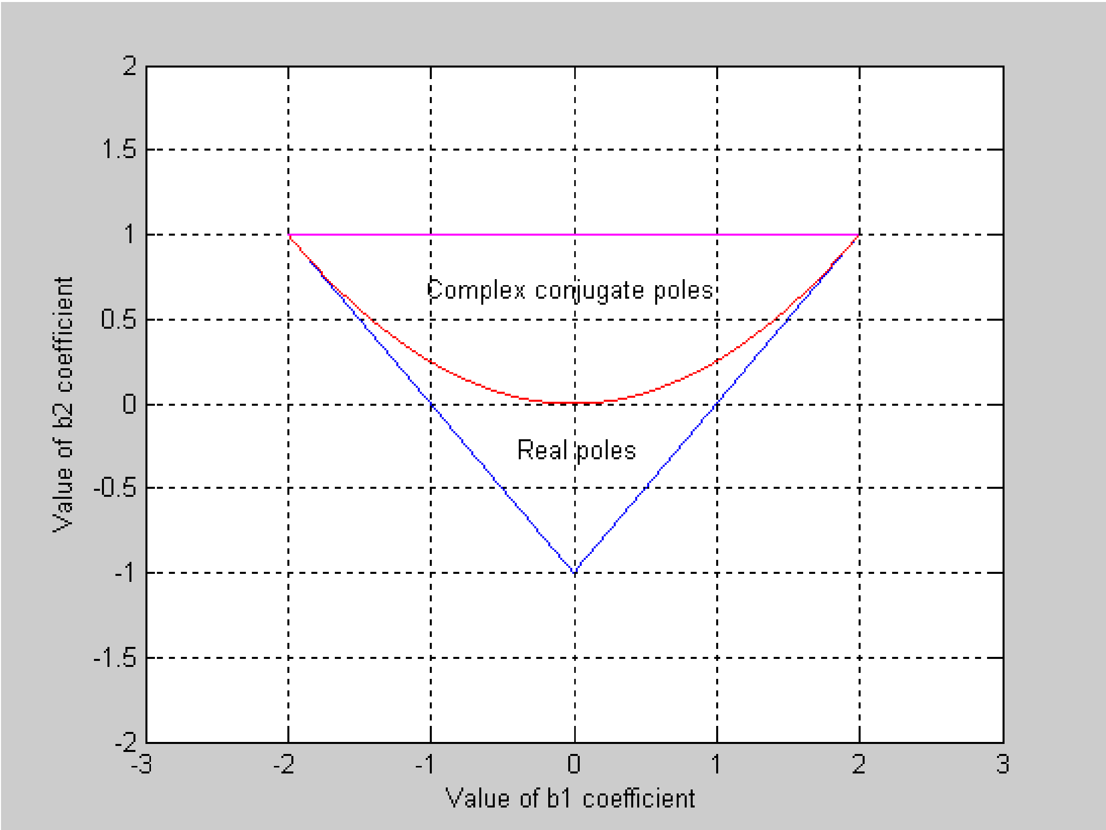

Figure 4.8. Stability triangle for second-order filter.

The points below the parabola correspond to real and distinct poles. Points on the parabola correspond to real and equal (double) poles, while points above the parabola correspond to complex conjugate poles.

4.8 Simple Filter Design using Pole-Zero Placement

As noted above, the design process for a resonator basically involves selecting the pole frequency and radius according to the requirements of the problem, and then calculating the digital filter coefficients using simple equations. This is simply a special case of a more general filter design technique called “pole-zero placement”. As the name suggests, the technique basically involves directly picking the locations of poles and zeros to meet the requirements of the application. Of course, in the case of zeros, instead of a peak in the magnitude response, we get a “notch” in the magnitude response close to the zero frequency. In particular, a zero placed on the unit circle results in complete rejection of any signal components at the zero frequency.

The general procedure for calculating the numerator coefficients in this case is the same as for calculating denominator coefficients, except that stability is not dependent on zero locations, so the limitations that apply to pole locations do not hold for zeros.

Example 4.3

Using pole-zero placement, design a filter with the following specification:

- Complete rejection of DC inputs
- Narrow pass band at 200 Hz, with bandwidth of 15 Hz
- Sampling rate = 500 Hz

We calculate the pole terms as before:

$$\theta_0 = 2\pi \frac{f_0}{f_{samp}} = 2\pi \frac{200}{500} = 0.8\pi$$

$$r \approx 1 - \frac{\Delta f}{f_{samp}}\pi = 1 - \frac{15}{500}\pi = 0.9057$$

The coefficients are calculated as follows:

$$b_2 = r^2 = 0.8204$$

$$b_1 = -2r\cos(\theta_c) = -2(0.9057)\cos\left(0.8\pi\right) = 1.4654$$

For the zero, the requirement for complete rejection of DC means we need a zero at $z = 1$. Therefore, the numerator polynomial will be $(z - 1)$. Hence, the overall transfer function of the filter will be:

$$\begin{aligned} H(z) &= \frac{z-1}{z^2 + 1.4654z + 0.8204} \\ &= \frac{z^{-1} - z^{-2}}{1 + 1.4654z^{-1} + 0.8204z^{-2}} \end{aligned}$$

The magnitude response and pole-zero map are plotted in Figure 4.9.

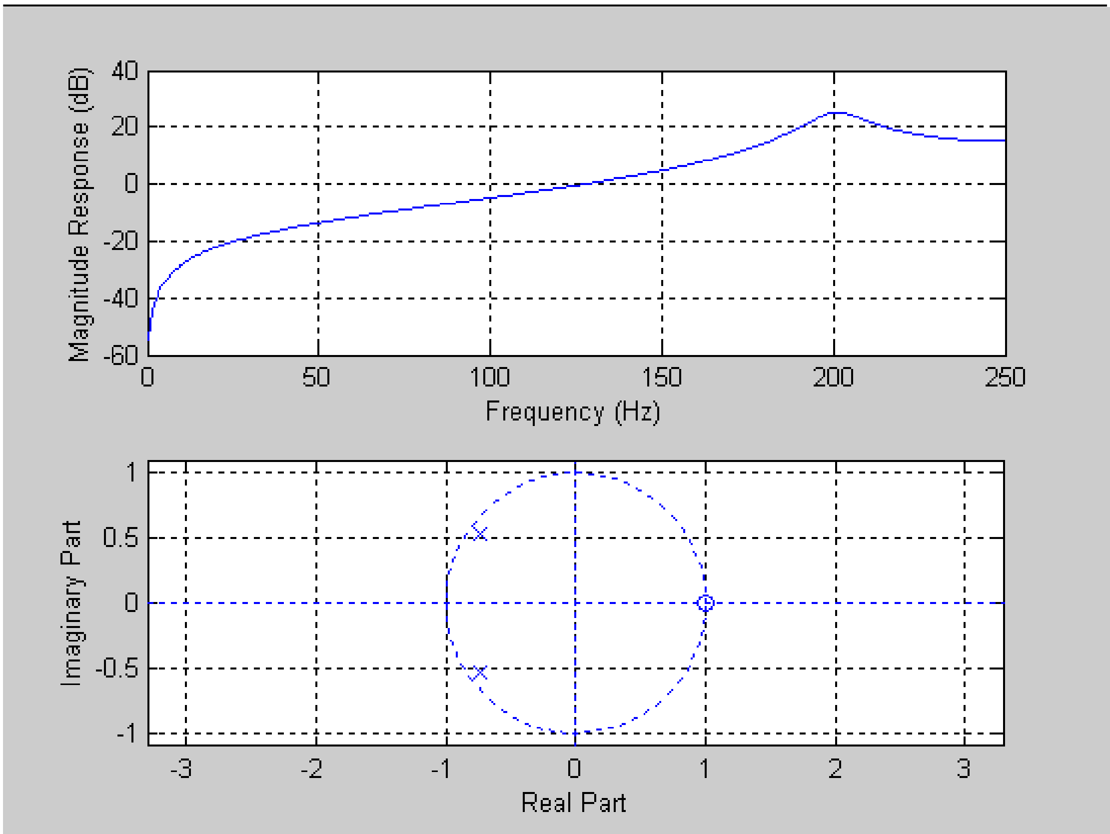

Figure 4.9. Magnitude response (upper) and pole-zero map (lower) of filter from Example 4.3.

In some applications, we require complete rejection of a very narrow band of frequencies in the input signal; a very common example of this is in instrumentation or biomedical engineering applications (e.g. ECG measurements), where the desired signal can often be contaminated by interference from the mains (50 Hz). This suggests that we need to design a notch filter, with a zero at 50 Hz, to effect complete rejection of the unwanted interference. However, if we simply place conjugate zeros at the interference frequency, they will have a very significant influence on the frequency response, and may well remove much of the "wanted" signal as well (i.e. the notch will be quite "broad"). To combat this, we normally place a pair of complex conjugate poles at the same frequency as the zeros, and close to the unit circle. The effect of these poles is to "cancel out" the zeros at frequencies other than those close to the zero frequency. At the same time, at frequencies close to the zero frequency, the zero will dominate, and will effect rejection of the signal at this frequency. In other words, the addition of poles tends to make the notch "narrower". The procedure for determining the pole radius in this case uses the same empirical equation as was used to determine pole radius in previous examples.

Example 4.4

A biomedical instrumentation application requires the removal of mains hum from a signal that is sampled at 300 Hz. Design a notch filter to achieve this. A notch of width 10 Hz should be sufficient.

As before, we need to calculate the pole and zero frequencies, and pole radius (note that since the zeros sit on the unit circle, the zero radius is equal to 1):

$$\theta_0 = 2\pi \frac{f_0}{f_{samp}} = 2\pi \frac{50}{300} = 0.333\pi$$

$$r \approx 1 - \frac{\Delta f}{f_{samp}} \pi = 1 - \frac{10}{300} \pi = 0.8953$$

The denominator coefficients are:

$$b_2 = r^2 = 0.8015$$
$$b_1 = -2r\cos(\theta_c) = -2(0.8953)\cos(0.333\pi) = -0.8969$$

while the numerator coefficients are:

$$a_2 = r^2 = 1$$
$$a_1 = -2r\cos(\theta_c) = -2(1)\cos(0.333\pi) = -1.0018$$

Hence, the transfer function is given by:

$$H(z) = \frac{1 - 1.0018z^{-1} + z^{-2}}{1 - 0.8969z^{-1} + 0.8015z^{-2}}$$

The magnitude response and pole-zero map are plotted in Figure 4.10.

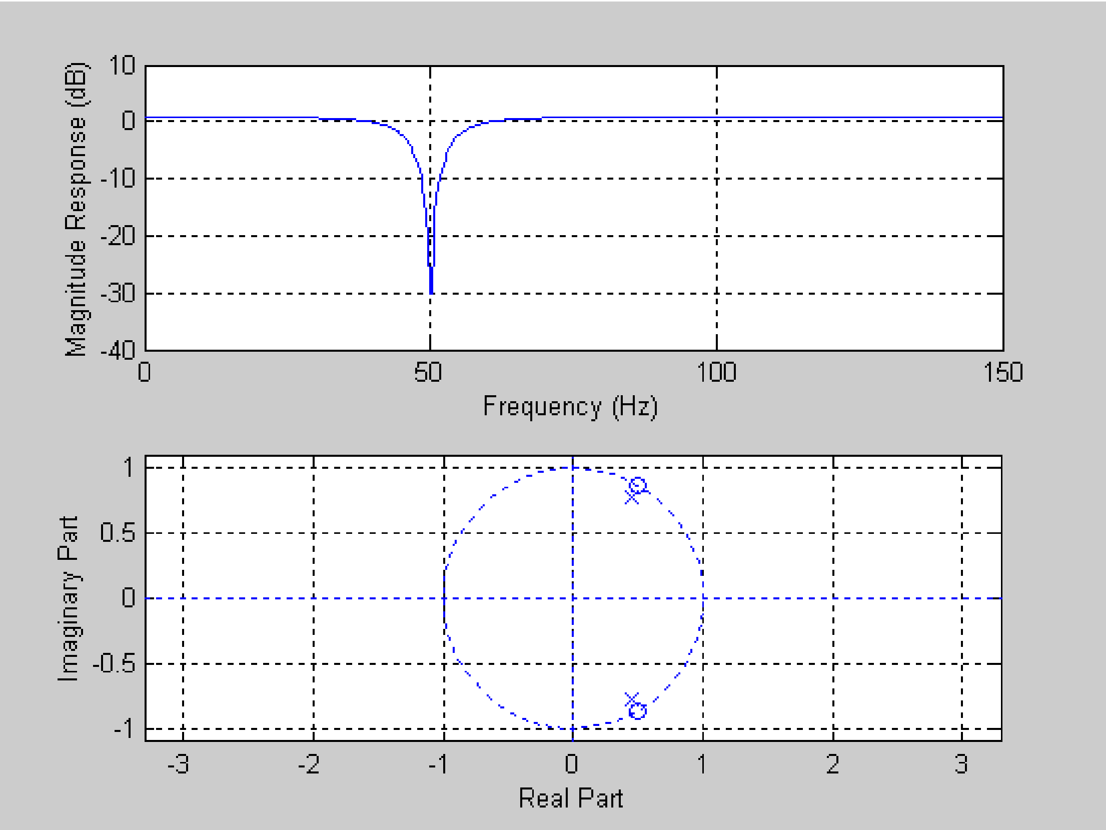

Figure 4.10. Magnitude response (upper) and pole-zero map (lower) of filter from Example 4.4.

<u>Exercise 4.7</u>

Plot the frequency response and pole-zero map for the case where no poles are included in the transfer function.

4.9 Oscillators

In this sub-section, we will examine another "special" form of second-order filter, specifically, one that has its poles sitting on the unit circle. As a consequence of this, the filter is marginally stable, and its output oscillates at a frequency corresponding to the pole frequency. One form of equation for the output signal of such an oscillator is:

$$p(n) = A\cos(n\theta_0)$$

where A is the amplitude of the cosine output, and θ_0 is its (digital) frequency in radians (a form using sine instead of cosine is also possible). From the table of z-Transforms in Section 2, we know that the z-transform of $p(n)$ is:

$$P(z) = \frac{1-\cos(\theta_0)z^{-1}}{1-2\cos(\theta_0)z^{-1}+z^{-2}}$$

If we view this as the oscillator "transfer function", and take the inverse z-Transform, we obtain the following difference equation:

$$y(n) = [x(n) - \cos(\theta_0)x(n-1)] + 2\cos(\theta_0)y(n-1) - y(n-2)$$

However, by definition, an oscillator will "oscillate" without any input signal, so we set the terms involving $x(n)$ to zero to obtain:

$$y(n) = 2\cos(\theta_0)y(n-1) - y(n-2)$$

A block diagram of this system is shown in Figure 4.11.

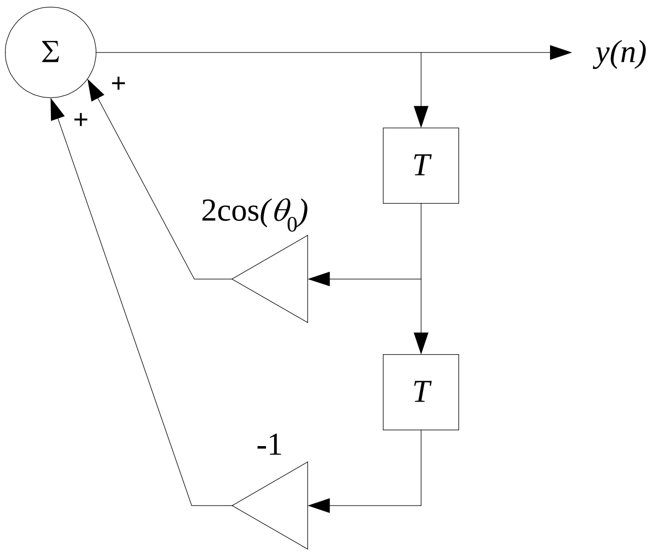

Figure 4.11. Block diagram of an oscillator.

To obtain the required self-sustaining oscillation given by $p(n)$ above, we need to calculate initial conditions. Assuming that the first output sample corresponds to n=0, we use the following initial conditions:

$$y(-1) = A\,cos(-\theta_0)$$

$$y(-2) = A\,cos(-2\theta_0)$$

Note that $y(-1)$ and $y(-2)$ represent $y(n-1)$ and $y(n-2)$ for $n = 0$. We can also choose a different starting value for n in order to meet some requirements e.g. if we wanted the generated sinusoid to have a particular starting phase, we would simply calculate what proportion of one period of the waveform this corresponds to, convert this to samples, and use this value as the time index for the first output, and choose the initial conditions correspondingly.

Also, note that the usual causality assumption for digital filters is not made in this case, and the digital filter memory needs to be appropriately initialised in order to generate a self-sustaining oscillation. Intuitively, this makes sense because, with no input signal, how could the filter produce an output without recursion and appropriate values in the filter memory?

From Figure 4.11, it can be clearly seen that an oscillator is a particular case of a resonator, with poles sitting on the unit circle (i.e. $r = 1$). Its transfer function is given by:

$$H(z) = \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

where $b_2 = 1$, and the coefficient b_1 is related to the oscillation frequency by:

$$b_1 = -2\cos(\theta_0)$$

This is the same equation as for a resonator, of course, but with $r = 1$ (again, note how the denominator coefficients change sign when going from the transfer function to the difference equation).

<u>Exercise 4.8</u>

Calculate the coefficients of an oscillator whose frequency of oscillation is 100 Hz (the sampling frequency is 10 kHz). Write Matlab code to calculate the first 1000 samples of the oscillator output signal, and verify that the frequency of oscillation is as expected.

<u>Exercise 4.9</u>

Design an oscillator (including initial conditions) to produce

$$p(n) = A\sin(n\theta_0)$$

Hint: Start with the z-Transform of $\sin(\theta_0)$.

An important application of oscillators is in digital modulators for communication systems. In *EE308 Signals and Communications*, you studied the process of modulation, by which an information signal is modulated or shifted to a higher frequency to facilitate transmission. This usually involves multiplication of the information signal by a sine or cosine carrier. In many communication systems, both sine and cosine carriers are needed at the same time. A structure to achieve this in digital communications systems can be derived as follows.

From basic trigonometry, we know that:

$$\cos(n+1)\theta = \cos(n\theta)\cos(\theta) - \sin(n\theta)\sin(\theta)$$

If we let $y(n) = \cos(n\theta)$ and $x(n) = \sin(n\theta)$, we obtain

$$y(n+1) = \cos(\theta)y(n) - \sin(\theta)x(n)$$

Delaying both sides of this equation by one sample interval yields:

$$y(n) = \cos(\theta)y(n-1) - \sin(\theta)x(n-1)$$

Similarly:

$$\begin{aligned} & \sin(n+1)\theta && = \cos(n\theta)\sin(\theta) + \sin(n\theta)\cos(\theta) \\ \Rightarrow \quad & x(n+1) && = \sin(\theta)y(n) + \cos(\theta)x(n) \\ \Rightarrow \quad & x(n) && = \sin(\theta)y(n-1) + \cos(\theta)x(n-1) \end{aligned}$$

Combining these two equations results in the “coupled” oscillator structure shown in Figure 4.12.

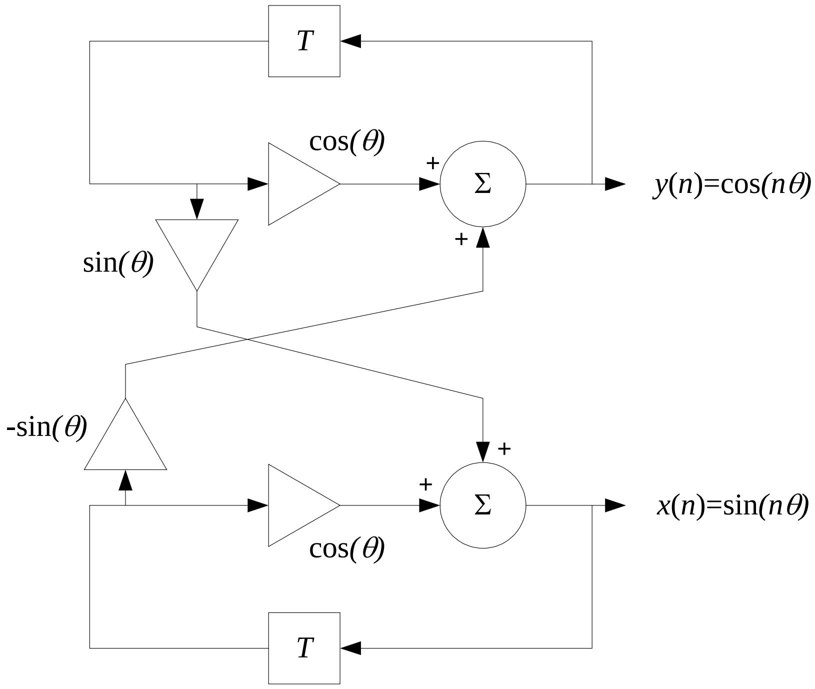

Figure 4.12. Coupled oscillator to produce sine and cosine carriers.

<u>Exercise 4.10</u>

Determine suitable initial conditions for the coupled oscillator in Figure 4.12.

Exercise 4.11

For an oscillator whose output is $p(n)=\sin(n\theta_0)$, where θ_0 is determined by the coefficient b_1, show that a small change in oscillation frequency Δf_0 is related to a change Δb_1 in the coefficient b_1 by the following expression:

$$\Delta f_0 = \frac{-f_{samp}\Delta b_1}{4\pi\cos\left(2\pi\frac{f_0}{f_{samp}}\right)}$$

where f_0 is the frequency of oscillation in Hz, and f_{samp} is the sampling frequency.