6. Design of Infinite Impulse Response (IIR) Filters

6.1 Introduction

Section 1 covered the basics of discrete-time signal and system analysis, Section 2 covered transform-domain analysis, while Section 3 covered analysis in the frequency-domain. Section 4 looked at structures for implementing discrete-time systems – "digital filters" (this Section also looked at the design of some "special" filters like resonators and oscillators). At this stage, we have all of the tools for analysis (and implementation) of digital filters, so we now come to the topic of design of digital filters, i.e. determining a transfer function and/or impulse response based on a desired specification. This section deals with the design of Infinite Impulse Response (IIR) filters, while the next section deals with design of FIR filters. First, however, we look at some "generic" topics that apply to all types of filters (some of this material will already be familiar to you from *EE308 Signals and Communications*).

6.2 Overview of Filter Design

From previous studies, you will be aware that filters can be classified into four main categories:

- Low pass
- High pass
- Band pass
- Band stop

We also have some other "special" filters, which may or may not fall into one of the above categories, e.g. resonators are often classified as band pass, while notch filters could be viewed as a form of band stop filter. There are other "special" filters that we will also come across, e.g. ideal differentiator ("high pass"), ideal integrator ("low pass") etc.

"Ideal" magnitude responses for the four types of filter are shown in Figure 6.1:

Generally speaking, IIR filters are designed by first designing a continuous-time ("analogue") filter that meets the required specifications, i.e. determining an s-domain transfer function, then transforming this s-domain transfer function into a z-domain transfer function (by means of a suitable mathematical transformation), thus producing a digital filter. This methodology allows us to re-use the very comprehensive set of design techniques that exist for analogue filters, e.g. Butterworth, Chebyshev etc. On the other hand, there is no real continuous-time equivalent for FIR filters, so this framework cannot be used for FIR filter design; instead, special design techniques exist for "direct" design of FIR filters.

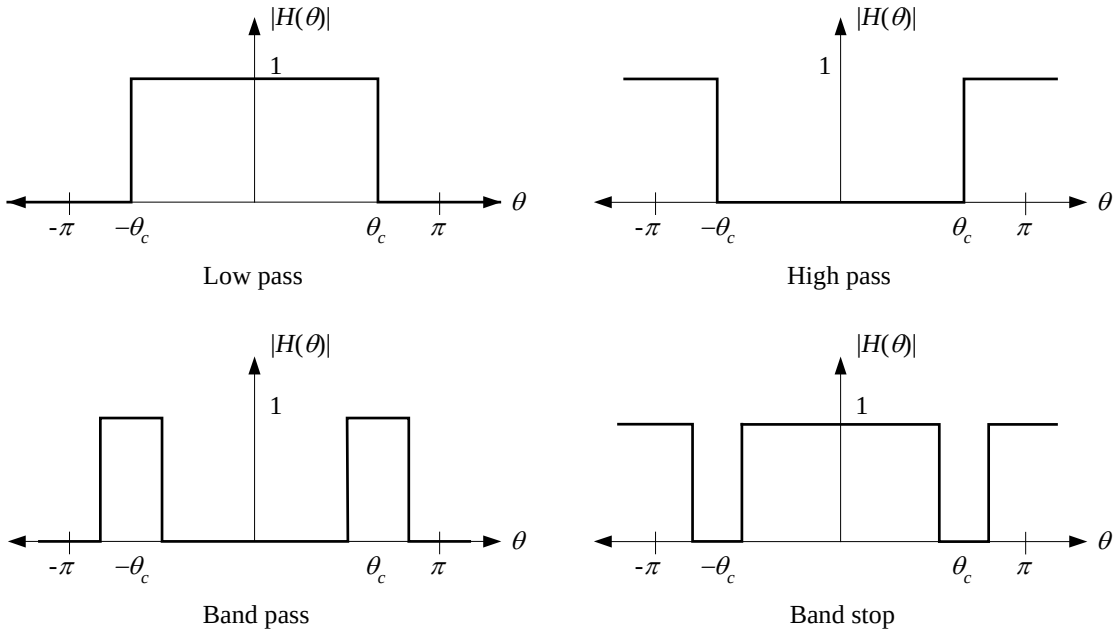

Figure 6.1. "Ideal" magnitude responses for four classes of filter.

The design of a digital filter (whether IIR or FIR) usually starts with a specification of the desired frequency response. In general, this requires specification of magnitude response and phase response characteristics. However, it is often the case that the phase response is not important; in which case, the magnitude response only is specified. The fact that the phase response is unspecified often leaves an extra "degree of freedom" that can be used in optimising the filter, i.e. greater phase distortion could be tolerated if it means that a simpler filter can be used. On the other hand, the application may place some constraints on the phase response, which in turn will influence the filter design process, for example, if the application requires linear phase, then an FIR filter should be used.

The desired filter specification is often described as shown in Figure 6.2:

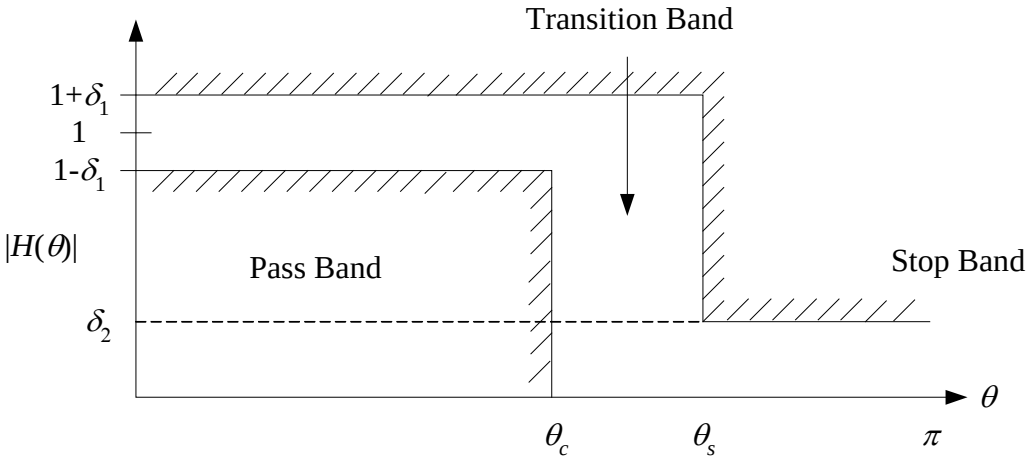

Figure 6.2. Typical magnitude response specification for a low pass filter.

The desired magnitude response for a low pass filter can be written as:

$1-\delta_1 \le |H(\theta)| \le 1+\delta_1 \qquad 0 \le \theta \le \theta_c$ Passband

$|H(\theta)| \le \delta_2 \qquad \theta_s \le \theta \le \pi$ Stopband

where δ_1 is the allowable pass band ripple, δ_2 is the stop band attenuation, θ_c is the cutoff frequency, while the stop band extends from θ_s to π. The region between θ_c and θ_s is the transition band. The magnitude response of the filter must not pass through the shaded areas indicated in Figure 6.2 above. As with analogue filter design, design of digital filters involves the usual trade-offs between filter complexity, width of transition band, tolerable ripple in the pass band etc. (especially so for IIR digital filters, since their design makes use of the same techniques as for analogue filters). Note that while Figure 6.2 refers to a low pass filter, the same principles apply for the other types of filter, and filter design often starts with a "prototype" low pass filter, following which this low pass filter is transformed to a high pass, band pass or band stop filter (see EE308).

The general procedure for digital filter design is as follows:

1. Obtain the performance specifications (usually the frequency response), along with any additional constraints, e.g. if linear phase is required, then an FIR filter will be needed, any constraints on implementation etc.
2. Determine the filter coefficients (the main topic of this Section).
3. Select the filter structure (cascade, parallel etc.). This may be influenced by implementation constraints, e.g. you may already have a second-order IIR filter developed and fully verified in VHDL, so in this case, it makes sense to implement your filter as a cascade of second-order sections, and to re-use the VHDL block to implement each section (thus saving on design and test effort for a "new" implementation).
4. Determine the required number of bits for coefficients and data, if the filter is to be implemented using finite-length binary numbers (as distinct from implementation using floating-point), and verify that the performance requirements can be met.
5. Implement using hardware or software.

In practice, some iteration is usually required, especially between Steps 3 and 5.

The following table highlights some differences between FIR and IIR filters:

FIR Filters	IIR Filters
Transfer function contains only zeros	Transfer function contains both poles and zeros
Can have exactly linear phase response	Cannot have linear phase
Less sensitive to quantisation noise in finite-precision implementation	More sensitive to quantisation noise (because of feedback)
Generally require more coefficients	Generally require fewer coefficients
Complexity is proportional to the length	No direct relationship between the length of the impulse response and filter

of the impulse response	complexity
No analogue counterpart.	Analogue filters can be readily transformed into digital filters

In general, for a given filter specification, an FIR filter will need more coefficients, and hence will be more complex (counterbalancing this is the fact that FIR filters have desirable phase characteristics that cannot be matched by IIR filters).

Example 6.1

The following transfer functions represent two different filters (FIR and IIR), meeting the same magnitude response characteristics:

$$H_1(z) = \frac{0.498182 + 0.927478z^{-1} + 0.498182z^{-2}}{1 - 0.674488z^{-1} - 0.363348z^{-2}}$$

$$H_2(z) = \sum_{k=0}^{11} h_k z^{-k}$$

where $h(0) = 0.546032\text{x}10^{-2} = h(11)$

$h(1) = -0.450687\text{x}10^{-1} = h(10)$

$h(2) = 0.691694\text{x}10^{-1} = h(9)$

$h(3) = -0.553844\text{x}10^{-1} = h(8)$

$h(4) = -0.634284\text{x}10^{-1} = h(7)$

$h(5) = 0.578924 = h(6)$

For each filter, draw a block diagram and write the difference equation. Comment on the relative complexity of the two filters.

Drawing the block diagrams and determining the difference equations is straightforward. In terms of complexity, it can be seen that the FIR filter requires 12 multiplies and 11 additions (though some efficiencies are possible), while the IIR filter requires 5 multiplications and 4 additions. In terms of memory requirements, the FIR filter requires 24 locations for coefficients and data (again, some efficiencies are possible), while the IIR filter requires 8 locations for coefficients and data (using a Direct Form II implementation).

In terms of the methods that may be used for designing digital filters, we have already noted that IIR filters are usually designed by "transforming" an analogue prototype filter from the s-domain to the z-domain, while for FIR filters special techniques have been developed. Techniques for designing IIR filters include the following:

- Impulse Invariant Transformation
- Bilinear Transformation
- Pole-Zero Placement Method

For FIR filters, the design techniques include:

- Windowing

- Frequency Sampling
- Numerical Optimisation

We have already dealt with the Pole-Zero Placement method for designing "simple" IIR digital filters where an "approximate" magnitude response is sufficient; however, where there are greater requirements on the accuracy of the frequency response, this method is not suitable, and methods involving transformation of an analogue filter are more commonly used. Both transformation methods (Impulse Invariant and Bilinear) transform a stable analogue filter to a stable digital filter, while preserving the essential features of the analogue filter's behaviour. Each method is based in some way on a "mapping" between the s-plane and the z-plane.

6.3 Impulse Invariant Transformation

The Impulse Invariant Transformation technique can be used to transform an analogue filter to a digital filter so that the impulse response of the digital filter is a sampled version of the impulse response of the analogue filter, i.e. the objective is to preserve the impulse response (hence the name of the method). In other words:

$$h(n) = h_a(t)\big|_{t=nT}$$

where $h_a(t)$ is the impulse response of the analogue filter, and T is the sampling period, as usual. However, the fact that the impulse responses are "equivalent" does not guarantee that the frequency response of the digital filter will be the same as that of the analogue filter (at least over the frequency range $0 \le f \le f_{samp}/2$). We have already seen in Section 2 that the sampling process results in a mapping between the s-plane and the z-plane whereby "strips" of the left half of the s-plane of width $2\pi/T$ are each mapped into the interior of the unit circle, while strips of the right half of the s-plane are mapped to the exterior of the unit circle. By the same token, we have seen in Section 3 that sampling of an analogue signal results in a spectrum that consists of an infinite set of copies of the analogue spectrum, repeating at intervals of 2π. If the sampling rate is too low, then aliasing can occur. Intuitively, what the Impulse Invariant Transformation is doing is sampling an analogue impulse response at some specified sampling rate, therefore, the effect on the system behaviour will be as described in Sections 2 and 3. In particular, if the sampling rate is too low, then the copies of the "spectrum", i.e. the magnitude response of the filter, will overlap, resulting in aliasing. Therefore, the Impulse Invariant Transformation is not suitable for filters whose magnitude responses have significant amplitude above half the desired sampling frequency, e.g. a high pass filter. The definition of "significant" is somewhat application-dependent; in essence, the designer has to accept that some aliasing will occur (after all, the magnitude response of an analogue filter will never be truly "zero" at any frequency – see *EE308 Signals and Communications*), so it's a question of deciding how much aliasing is tolerable, and selecting the sampling period accordingly.

While the Impulse Invariant Transformation preserves the impulse response of the analogue filter, the design procedure involves starting with the transfer function of an analogue filter, $H(s)$, and transforming this to the z-domain. To see how this mapping comes about, we take the simple case of a single pole in the s-domain:

$$\frac{1}{s+b}, \quad b>0$$

Inverse Laplace transformation of this expression gives us the corresponding time-domain expression:

$$h_a(t) = e^{-bt}$$

If we sample this with some sampling period T, we obtain:

$$h_a(nT) = e^{-bnT} = h(n), \qquad n \geq 0$$

Taking the z-transform of this expression (see the table in Section 2):

$$H(z) = \frac{1}{1-e^{-bT}z^{-1}}$$

Therefore, we can say that the Impulse Invariant transformation performs the following mapping from the s-plane to the z-plane:

$$\frac{1}{s+b} \rightarrow \frac{1}{1-e^{-bT}z^{-1}}, \quad b>0$$

Normally, the design procedure involves re-writing the s-domain transfer function in terms of single poles as indicated above, using the Method of Partial Fractions or some other means, and applying the mapping to each term (with simplification and combination of the terms after mapping). Note that the mapping applies only to s-plane poles; there is no equivalent mapping for zeros.

Example 6.2

Using the Impulse Invariant Transformation, convert the following analogue filter transfer function to a corresponding digital filter:

$$H(s) = \frac{2}{(s+1)(s+3)}$$

Using the Method of Partial Fractions, this can be re-written as:

$$H(s) = \frac{1}{(s+1)} - \frac{1}{s+3}$$

Applying the transformation to each term, we obtain:

$$H(z) = \frac{1}{1-e^{-T}z^{-1}} - \frac{1}{1-e^{-3T}z^{-1}}$$

Some algebraic manipulation of this expression yields:

$$H(z) = \frac{(e^{-T}-e^{-3T})z^{-1}}{1-(e^{-T}-e^{-3T})z^{-1}+e^{-4T}z^{-2}}$$

Example 6.3

Using the Impulse Invariant Transformation, convert the following analogue filter transfer function to a corresponding digital filter:

$$H(s) = \frac{4}{(s+1)(s+2)}$$

Choose a suitable sampling frequency for the resulting digital filter, and hence write the difference equation.

Using the Method of Partial Fractions, this can be re-written as:

$$H(s) = \frac{4}{s+1} - \frac{4}{s+2}$$

Applying the transformation to each term, we obtain:

$$H(z) = \frac{4}{1-e^{-T}z^{-1}} - \frac{4}{1-e^{-2T}z^{-1}}$$

Some algebraic manipulation of this expression yields:

$$H(z) = \frac{4(e^{-T} - e^{-2T})z^{-1}}{1-(e^{-T} + e^{-2T})z^{-1} + e^{-3T}z^{-2}}$$

For an analogue filter of the form $1/(s+a)$, a rule of thumb often used to choose a suitable sampling rate is to choose a value that is 5 to 10 times the cutoff frequency of the analogue filter (given by a in radians/s). In the example above, there are two such terms, hence we must examine the one with the higher cutoff frequency, which is ω_c = 2 radians/s, or $1/\pi$ Hz. Hence, a suitable sampling rate would be $5/\pi$ Hz, which means that T, the sampling period, is $\pi/5$ seconds. Substituting this value into the digital transfer function above yields:

$$H(z) = \frac{0.9955z^{-1}}{1-0.8181z^{-1} + 0.1518z^{-2}}$$

with difference equation:

$$y(n) = 0.9955x(n\text{-}1) + 0.8181y(n\text{-}1) - 0.1518y(n\text{-}2)$$

Matlab contains a function called `impinvar`, which transforms an analogue transfer function (specified in terms of numerator and denominator polynomials in s) into a digital transfer function.

Example 6.4

Use Matlab to design a digital filter (using the impulse invariant transformation) from the analogue filter in Example 5.2 above. Plot the frequency response of the digital filter and compare it with the frequency response of the analogue filter (hint: use the Matlab function `freqs` to calculate the frequency response of an analogue filter). Examine the effect of the sampling frequency on the digital filter by choosing two different values of sampling rate, one equal to three times the cutoff frequency, and one equal to ten times the cutoff frequency.

Note the apparent scaling of the digital frequency response by the factor $f_{samp}=1/T$. This is a consequence of the sampling of the analogue impulse response (see Section 3.2 above); as noted above, this operation preserves the impulse response, but can result in a slightly different frequency response.

6.4 Bilinear Transformation

As noted above, one of the problems associated with the Impulse Invariant Transformation is the fact that significant aliasing may occur if the sampling rate is not sufficiently high. This is because the Impulse Invariant Transformation maps "strips" of the left half of the s-plane into the interior of the unit circle of the z-plane. The Bilinear Transformation avoids this problem by mapping the *entire* left half of the s-plane into the interior of the unit circle in the z-plane (and the entire right half of the s-plane to the exterior if the unit circle in the z-plane).

The transformation is defined as follows. Starting with the mapping from the s-plane to the z-plane we already have:

$$z = e^{sT}$$

we can represent this in the form of a power series, as follows:

$$z = e^{sT} = \frac{e^{\frac{sT}{2}}}{e^{-\frac{sT}{2}}} = \frac{1+\frac{sT}{2}+\frac{1}{2!}\left(\frac{sT}{2}\right)^2+.....}{1-\frac{sT}{2}+\frac{1}{2!}\left(\frac{sT}{2}\right)^2-.....}$$

By dropping the higher-order terms, we obtain the following approximation:

$$z = e^{sT} \approx \frac{1+\frac{sT}{2}}{1-\frac{sT}{2}} \text{ or } z^{-1} \approx \frac{1-\frac{sT}{2}}{1+\frac{sT}{2}}$$

$$z^{-1}\left(1+\frac{sT}{2}\right) = 1-\frac{sT}{2}$$

$$1-z^{-1} = s\left(\frac{T}{2}\right)(z^{-1}+1)$$

$$\therefore s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$$

Because the Bilinear Transformation maps the entire left half of the s-plane into the interior of the unit circle in the z-plane, it can be seen that the entire imaginary axis is mapped onto the unit circle itself, in other words, the entire "analogue" frequency axis from 0 to ∞ is mapped onto a finite length of the unit circle from $\theta = 0$ to $\theta = \pi$ (and correspondingly in the negative frequency direction). Thus, compression or "warping" of the analogue frequency axis takes place during the transformation. This can be seen more clearly by considering how the imaginary axis in the s-plane maps to the unit circle $z=e^{j\theta}$:

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$$

Let $s = j\omega_a$ and $z = e^{j\theta}$:

$$j\omega_d = \frac{2}{T}\frac{1-e^{-j\theta}}{1+e^{-j\theta}} = \frac{2}{T}\frac{e^{-\frac{j\theta}{2}}(e^{\frac{j\theta}{2}} - e^{-\frac{j\theta}{2}})}{e^{-\frac{j\theta}{2}}(e^{\frac{j\theta}{2}} + e^{-\frac{j\theta}{2}})} = \frac{2}{T}\frac{jSin\left(\frac{\theta}{2}\right)}{Cos\left(\frac{\theta}{2}\right)}$$

$$\omega_a = \frac{2}{T}\tan\left(\frac{\theta}{2}\right)$$

Inverting the above equation, we obtain :

$$\theta = 2\tan^{-1}\left(\frac{\omega_a T}{2}\right)$$

In other words, the usual relationship between analogue and digital frequency – $\theta = \omega_a T$ – has been replaced by an inverse tan relationship. From this equation, it can be seen that for low values of ω_a, we have $\theta \approx \omega_a T$, i.e. the linear relationship between analogue and digital frequency is "approximately" preserved. However, for higher values of ω_a, the compressive behaviour of the inverse tan function dominates. A plot of the this nonlinear relationship between analogue and digital frequency is shown in Figure 6.3:

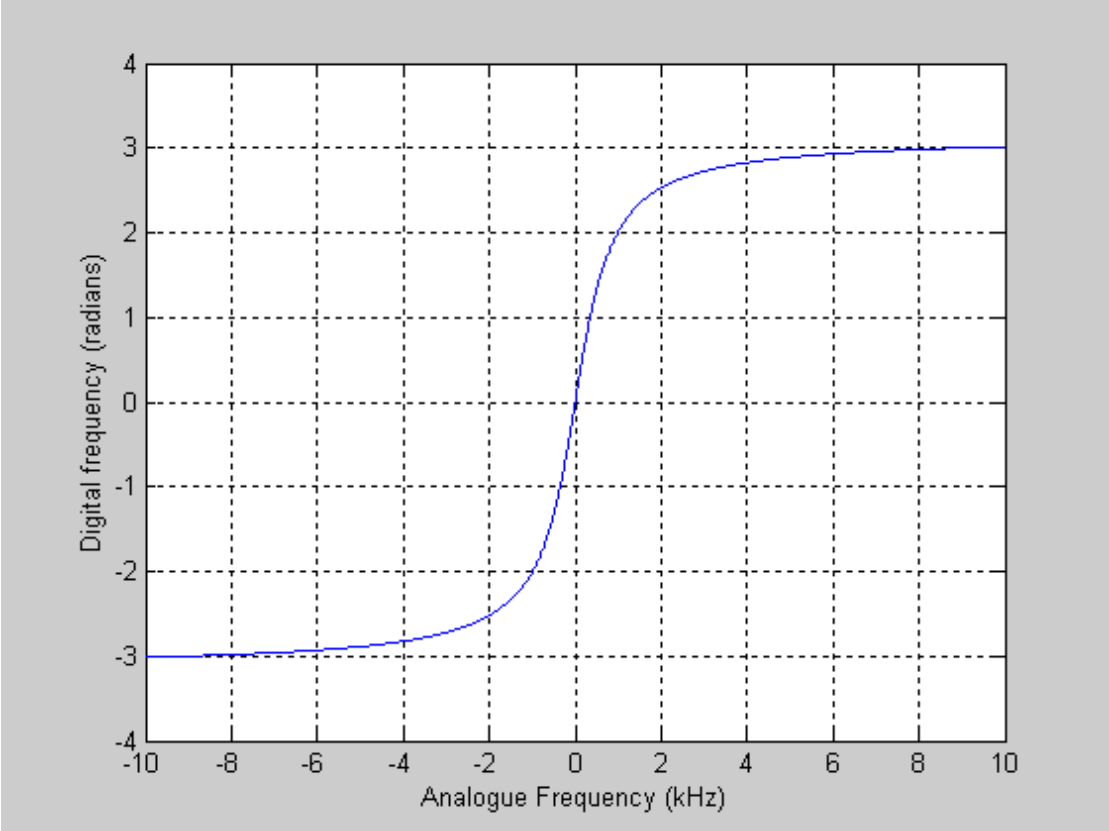

Figure 6.3. Plot of nonlinear relationship between analogue and digital frequency (sampling frequency = 2 kHz).

The problem with this warping is that the "important" frequencies (cut off frequency, start of stop band etc.) will be "moved" during the transformation. Therefore, it is necessary to "pre-warp" the analogue frequency axis before carrying out the transformation – the "pre-warping" combined with the warping caused by the transformation cancel each other out. Therefore, in designing a digital filter using this method, it is necessary to "pre-warp" the significant analogue frequencies before designing the analogue filter.

Example 6.5

A simple analogue low pass filter is given by the following transfer function:

$$H(s) = \frac{\omega_c}{s + \omega_c}$$

where the cut off frequency is given by ω_c. Determine the corresponding digital low pass filter using the Bilinear Transformation, if the desired cut off frequency is 1 kHz and the sampling rate is 8 kHz.

Solution

The transfer function of the analogue low pass filter is:

$$H(s) = \frac{2000\pi}{s + 2000\pi} = \frac{6283.2}{s + 6283.2}$$

If we simply apply the bilinear transformation directly to the analogue filter transfer function, we will obtain a low pass filter, however, the cut off frequency will be warped as follows:

$$\theta_d = 2\tan^{-1}\left(\frac{\omega_c T}{2}\right) = 2\tan^{-1}\left(\frac{2000\pi}{16000}\right)$$

$$\omega_d = \frac{\theta_d}{T} = 5987 \text{ rad/s} \quad \text{or} \quad f_d = 952 \text{ Hz}$$

In other words, the digital low pass filter has a cut off frequency of 962 Hz, instead of the desired 1000 Hz; this is because the frequency axis has been "compressed" by the transformation. We must "pre-warp" the analogue cut off frequency as follows. First, determine the digital cut off frequency in the usual way:

$$\theta_c = 2\pi\frac{f_c}{f_s} = 2\pi\frac{1000}{8000} = \frac{\pi}{4}$$

Then we calculate the "pre – warped" analogue frequency:

$$\omega_a = \frac{2}{T}\tan\left(\frac{\theta_c}{2}\right) = 16000\tan\left(\frac{\pi}{8}\right) = 6627.4 \text{ rad/s}$$

Thus, the desired analogue transfer function will be:

$$H(s) = \frac{6627.4}{s + 6627.4}$$

i.e. the cut off frequency of the analogue filter has been increased to compensate for the compression that will occur during the Bilinear Transformation. The digital filter transfer function is calculated by substituting for *s*, as follows:

$$H(z) = \frac{6627.4}{16000\dfrac{1-z^{-1}}{1+z^{-1}} + 6627.4} = \frac{6627.4(1+z^{-1})}{16000(1-z^{-1}) + 6627.4(1+z^{-1})}$$

$$H(z) = \frac{6627.4(1+z^{-1})}{22627.4 - 9372.6z^{-1}} = \frac{0.2929 + 0.2929z^{-1}}{1 - 0.4142z^{-1}}$$

Difference Equation :

$$y(n) = 0.2929x(n) + 0.2929x(n\text{-}1) + 0.4142y(n\text{-}1)$$

Example 6.6

Design a digital low pass filter with the following specification:

- Cut off frequency = 10 kHz
- Transition band from 10 to 20 kHz
- Sampling rate = 100 kHz
- Stop band attenuation of at least 10 dB
- No ripple in pass band and stop band.

Solution

The requirement to have no ripple indicates that a Butterworth filter is required. The first step is to calculate the desired digital cut off and stop band frequencies, and then the corresponding "pre-warped" analogue frequencies:

$$\theta_c = 2\pi\frac{10000}{100000} = 0.2\pi \qquad \theta_s = 2\pi\frac{20000}{100000} = 0.4\pi$$

$$\omega_a = \frac{2}{T}\tan\left(\frac{\theta_c}{2}\right) = 200000\tan(0.1\pi) = 64983.9$$

$$\omega_s = \frac{2}{T}\tan\left(\frac{\theta_s}{2}\right) = 200000\tan(0.2\pi) = 145308.5$$

Given the pre-warped cut off and stop band frequencies, the next step is to calculate the transfer function of the Butterworth filter. The order of the filter, *N*, is calculated using the following equation (*see EE308 Signals and Communications*):

$$\left|H_a(\omega)\right|^2 = \frac{1}{1+\left(\dfrac{\omega_s}{\omega_c}\right)^{2N}}$$

Solving this gives N=1.367, which is rounded up to N=2, i.e. we require a two-pole Butterworth filter. The poles occur in a complex conjugate pair, and lie in the left half of the s-plane, with values as follows:

$$p_1 = \omega_c e^{j\frac{3\pi}{4}} = 0.6498\text{x}10^5\left(-0.7071 + j0.7071\right)$$

$$p_2 = \omega_c e^{j\frac{5\pi}{4}} = 0.6498\text{x}10^5\left(-0.7071 - j0.7071\right)$$

So, the transfer function is:

$$\begin{aligned} H(s) &= \frac{p_1 p_2}{(s-p_1)(s-p_2)} \\ &= \frac{4.223\text{x}10^9}{s^2 + 0.919\text{x}10^5 s + 4.223\text{x}10^9} \end{aligned}$$

Applying the Bilinear Transformation:

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} = 2\text{x}10^5\frac{1-z^{-1}}{1+z^{-1}}$$

$$H(z) = \frac{0.0675 + 0.1349z^{-1} + 0.0675z^{-2}}{1 - 1.143z^{-1} + 0.4128z^{-2}}$$

Example 6.7

Determine the transfer function for the digital equivalent of the simple, first-order, low pass RC filter. The normalised s-domain transfer function for this filter is:

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

Assume a sampling rate of 150 Hz and a cutoff frequency of 30 Hz.

Solution

First, determine the desired digital cut off frequency, and its "pre-warped" analogue equivalent:

$$\theta_c = 2\pi\frac{f_c}{f_s} = 2\pi\frac{30}{150} = 0.4\pi$$

$$\omega_a = \frac{2}{T}\tan\left(\frac{\theta_c}{2}\right) = 217.95\,\text{rad/s}$$

The denormalised analogue transfer function is obtained by replacing s with s/ω_a:

$$H(s) = \frac{1}{\frac{s}{\omega_a}+1} = \frac{\omega_a}{s+\omega_a}$$

Applying the Bilinear Transformation yields the following digital transfer function:

$$H(z) = \frac{0.4208(1 + z^{-1})}{1 - 0.1584z^{-1}}$$