EE445 – Digital Signal Processing

IIR Filter Design

Dr Brian Deegan
Autumn 2022

Design of Infinite Impulse Response (IIR) Filters

- Introduction
  - Section 1 - discrete-time signal and system analysis
  - Section 2 - transform-domain analysis
  - Section 3 - analysis in the frequency-domain.
  - Section 4 - structures for implementing discrete-time systems – "digital filters"
- Now we deal with design of digital filters, i.e. determining a transfer function and/or impulse response for a desired specification

Design of Infinite Impulse Response (IIR) Filters

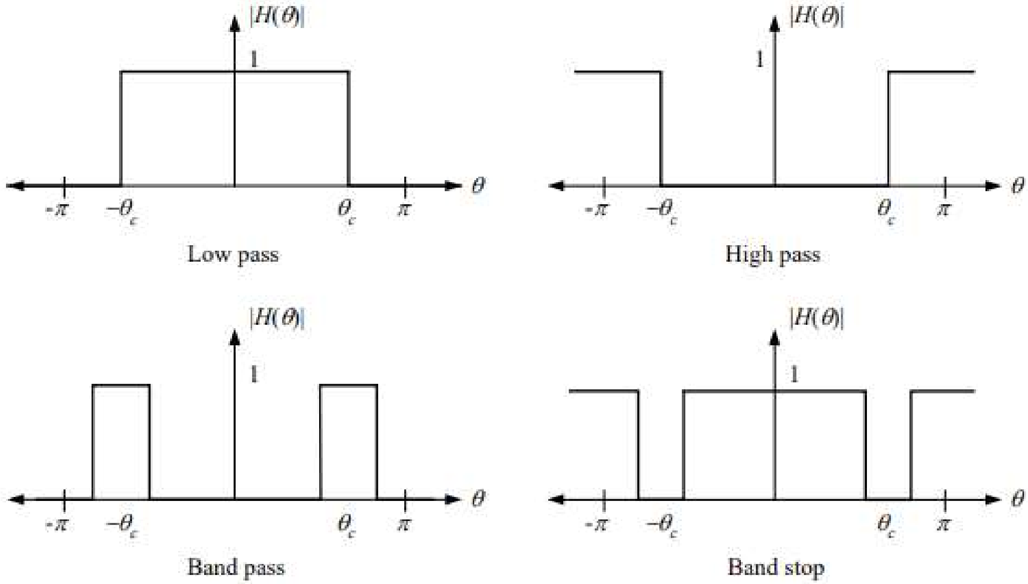
- This section - design of Infinite Impulse Response (IIR) filters
- Next section deals with design of FIR filters.
- First, however, we look at some "generic" topics (some of which were covered in EE357 and elsewhere)

Overview of Filter Design

- Filters can be classified into four main categories:
  - Low pass
  - High pass
  - Band pass
  - Band stop
- We also have some other “special” filters, 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.

Overview of Filter Design

- "Ideal" magnitude responses for the four types of filter are shown below



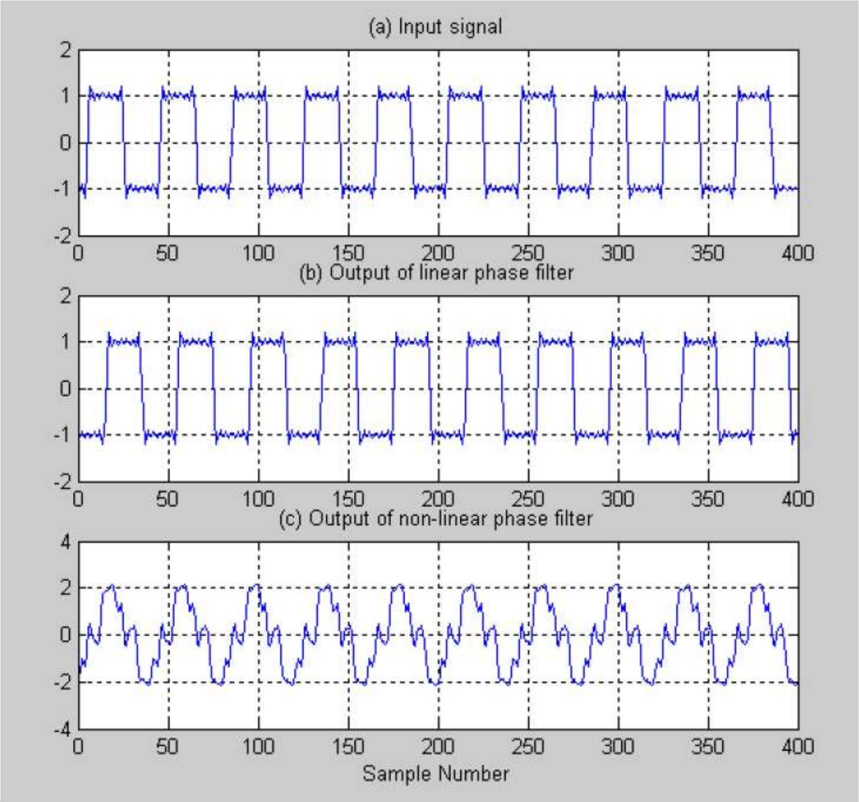
Overview of filter design

- IIR filters usually 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
- Mathematical transformation required for this.
- Allows us to re-use the very comprehensive set of design techniques that exist for analogue filters, e.g. Butterworth, Chebyshev etc
- However, there is no real continuous-time equivalent for FIR filters - special design techniques exist for "direct" design of FIR filters.

Overview of filter design

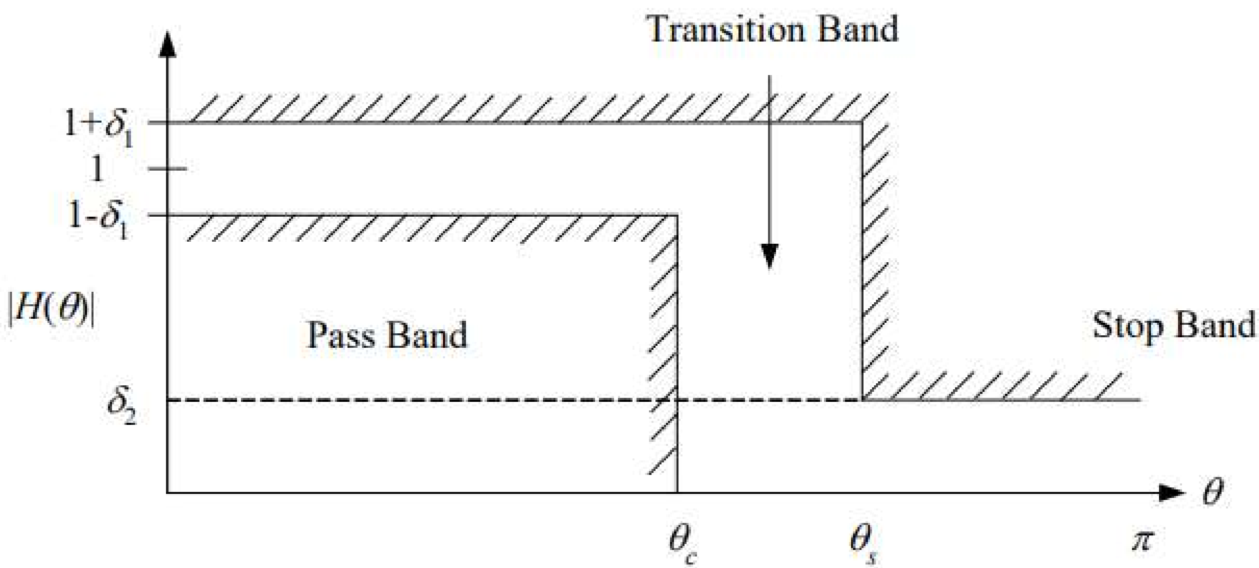
- Design usually starts with a specification of the desired frequency response.
- Magnitude response and phase response characteristics.
- Often the case that the phase response is not important – extra "degree of freedom"
- On the other hand, the application may place some constraints on the phase response, for example, if the application requires linear phase, then an FIR filter should be used.

Side note – effect of phase distortion



Overview of filter design

- The desired filter specification is often described as shown below



Overview of filter design

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

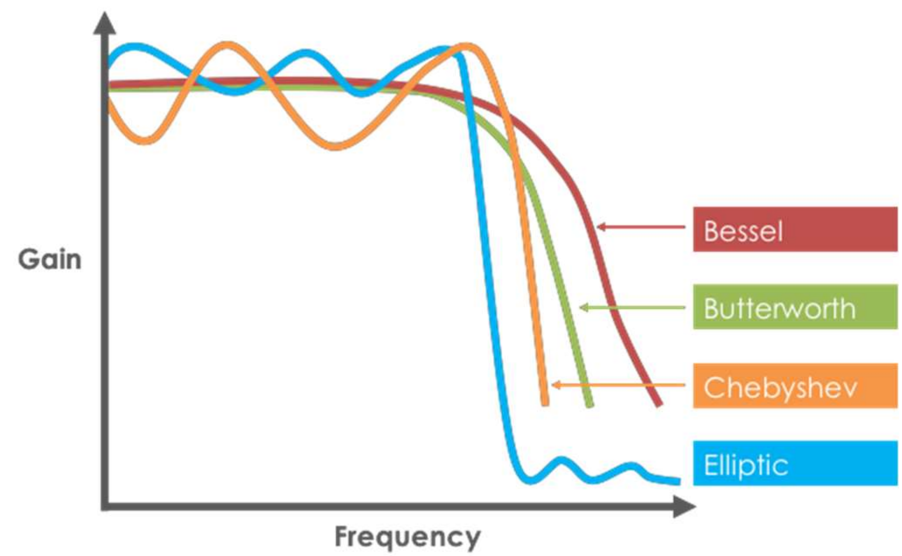
$$1 - \delta_1 \le |H(\theta)| \le 1 + \delta_1 \quad 0 \le \theta \le \theta_c \quad Passband$$

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

- where δ_1 is the allowable pass band ripple, δ_2 is the stop band attenuation, θ_c is the cut-off 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 above.
- Design of digital filters involves trade-offs …

Analogue Filter summary

- Filter selection It depends on the application
- Many different elements that must be traded during the filter design process such as:
  - Frequency response
  - Selectivity (i.e. control over frequency range)
  - Number of stages (complexity)
  - Phase delay



Overview of filter design

- The previous Figure refers to a low pass filter, but same principles apply for the other types of filter
- 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
- The general procedure for digital filter design is as follows:
  1. Obtain the performance specifications 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 …

Overview of filter design

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.

Overview of filter design

- 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 of the impulse response	No direct relationship between the length of the impulse response and filter complexity
No analogue counterpart	Analogue filters can be readily transformed into digital filters

Example 6.1

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

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

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

$$\begin{aligned}
\text{where}\quad & h(0) && = 0.546032\text{x}10^{-2} && = h(11) \\
& h(1) && = -0.450687\text{x}10^{-1} && = h(10) \\
& h(2) && = 0.691694\text{x}10^{-1} && = h(9) \\
& h(3) && = -0.553844\text{x}10^{-1} && = h(8) \\
& h(4) && = -0.634284\text{x}10^{-1} && = h(7) \\
& h(5) && = 0.578924 && = h(6)
\end{aligned}$$

Example 6.1

- For each filter, draw a block diagram and write the difference equation. Comment on the relative complexity of the two filters. Block diagrams and determining the difference equations is straightforward.
- Complexity: FIR filter requires 12 multiplies and 11 additions (though some efficiencies are possible), while the IIR filter requires 5 multiplications and 4 additions.
- Memory requirements: 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).

Filter design techniques

- 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

Filter design techniques

- We have already dealt with the Pole-Zero Placement method for designing “simple” IIR digital filters
- However, this method is not suitable for “high-spec” designs
- 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.

Impulse Invariant Transformation

- 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.

Impulse Invariant Transform

- 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 $0 \le f \le \frac{f_{samp}}{2}$).
- 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 $\frac{2\pi}{T}$ 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.

Impulse Invariant Transform

- 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.
- Impulse Invariant Transformation samples the impulse response, therefore the effect on the system behaviour will be as described in Sections 2 and 3.
- Not suitable for filters whose magnitude responses have significant amplitude above half the desired sampling frequency, e.g. a high pass filter

Impulse Invariant Transform

- 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)
- Need to decide how much aliasing is tolerable, and selecting the sampling period accordingly.
- Design procedure involves starting with the transfer function of an analogue filter, $H(s)$, and transforming this to the z-domain.

Impulse Invariant Transform

- Take the simple case of a single pole in the s-domain:

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

- Inverse Laplace transformation gives

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

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

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

Impulse Invariant Transform

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

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

- Impulse Invariant transformation performs the following mapping:

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

- 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
- 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:

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

- 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:

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

Example 6.3

- For an analogue filter of the form $\frac{1}{s+a}$, a rule of thumb often used to choose a suitable sampling rate is to choose a value that is 5 to 10 times the cut-off frequency of the analogue filter (given by a in radians/s).
- In this example, there are two such terms, hence we must examine the one with the higher cut-off frequency, which is $\omega_c = 2\, radians/s$, or $\frac{1}{\pi}\ Hz$. Hence, a suitable sampling rate would be $\frac{5}{\pi}\ Hz$, which means that T, the sampling period, is $\frac{\pi}{5}$ seconds

Example 6.3

- 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

$$y(n) = 0.9955x(n-1) + 0.8181y(n-1) - 0.1518y(n-2)$$

Note: Matlab contains a function called *impinvar*

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 cut-off frequency, and one equal to ten times the cut-off 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); as noted above, this operation preserves the impulse response, but can result in a slightly different frequency response.

Bilinear Transformation

- 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)

Bilinear Transformation

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

$$z = e^{sT}$$

- Represent this in the form of a power series, as follows:

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

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

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

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

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

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

Bilinear Transformation

- The entire imaginary axis is mapped onto the unit circle itself
- 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 $z = e^{j\theta}$

Bilinear Transformation

- Recall

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

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

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

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

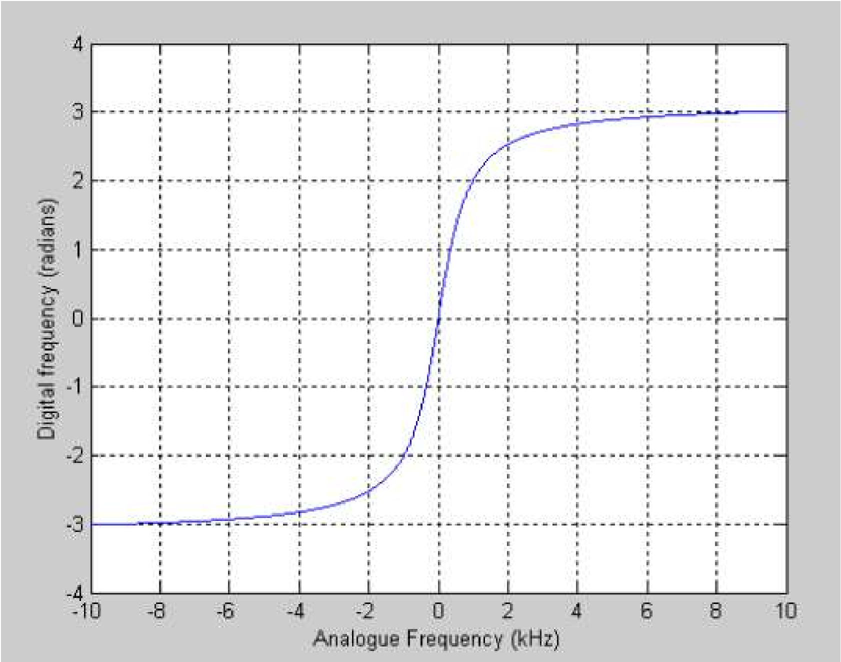
- Inverting the above equation, we get

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

Bilinear Transformation

- In other words, the usual relationship between analogue and digital frequency $-\theta = \omega_a T$ has been replaced by an inverse tan relationship.
- • 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.
- For higher values of ω_a, the compressive behaviour of the inverse tan function dominates

Bilinear Transformation



Bilinear Transformation

- 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.

Example 6.5 - 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:

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

$$\omega_d = \frac{\theta_d}{T} = 5987rads^{-1}\ or\ f_d = 952Hz$$

Example 6.5 - analysis

- In other words, the digital low pass filter has a cut off frequency of 952 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:

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

- Then we calculate the “pre-warped” analogue frequency:

$$\omega_a = \frac{2}{T}\tan\left(\frac{\theta_c}{2}\right) = 16000\tan\left(\frac{\pi}{8}\right) = 6627.4 rads^{-1}$$

Example 6.5 - analysis

- 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.

Example 6.5 - analysis

- 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 :

$$y(n) = 0.2929x(n) + 0.2929x(n-1) + 0.4142y(n-1)$$

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.