Lecture 17

Design of IIR Filters : Bilinear Transformation

Design by mapping the analogue frequency axis to the digital one

Dr. Dimitris Agrafiotis, Dept. Of Electrical & Electronic Engineering

Design of IIR Filters: Bilinear Transformation

From Continuous Time to Discrete Time Filters

First there were continuous time filters only …..

When digital signal processing came along, design of digital filters relied on mapping/transformation of well-known continuous time filter designs to discrete time ones

Impulse Invariance and **Bilinear Transformation** are two methods providing such a transformation/mapping

Impulse Invariance

Impulse Invariance means that the impulse response of the digital filter will be similar to that of the continuous time filter – achieved through sampling of the impulse response of the continuous time filter

Bilinear Transformation

Bilinear Transformation applies a non-linear mapping of the analogue frequency axis to the digital frequency one

Design of IIR Filters: Bilinear Transformation

From Continuous Time to Discrete Time Filters

Actual IIR filter design steps

Specification of continuous-time (analogue) prototype low-pass filter

↓

Frequency transformation of analogue prototype low-pass filter

↓

Transformation of continuous-time filter to discrete-time filter

Impulse Invariance

Impulse Invariance means that the impulse response of the digital filter will be similar to that of the continuous time filter – achieved through sampling of the impulse response of the continuous time filter

Bilinear Transformation

Bilinear Transformation applies a non-linear mapping of the analogue frequency axis to the digital frequency one

Design of IIR Filters: Bilinear Transformation

From Continuous Time to Discrete Time Filters

Specification of continuous-time (analogue) low-pass filters in Matlab

Butterworth analogue filter

[N, Omegac] = buttord(Omegap, Omegas, Ap, As, ‘s’)

[b,a] = butter(N, Omegac, ‘s’)

Chebychev analogue filter (type I and II)

[N, Omegap] = cheb1ord(Omegap, Omegas, Ap, As, ‘s’)

[b,a] = cheby1(N, Ap,Omegap, ‘s’)

[N, Omegas] = cheb2ord(Omegap, Omegas, Ap, As, ‘s’)

[b,a] = cheby2(N, Ap,Omegas, ‘s’)

Elliptic analogue filter

[N, Omegap] = ellipord(Omegap, Omegas, Ap, As, ‘s’)

[b,a] = ellip(N, Ap, As ,Omegap, ‘s’)

N	-	filter order
Omegac	-	3dB cutoff frequency
Omegap	-	passband right edge
Omegas	-	stopand left edge
Ap	-	passband ripple
As	-	stopband ripple
‘s’	-	Analogue filter
b	-	Numerator
a	-	Denominator

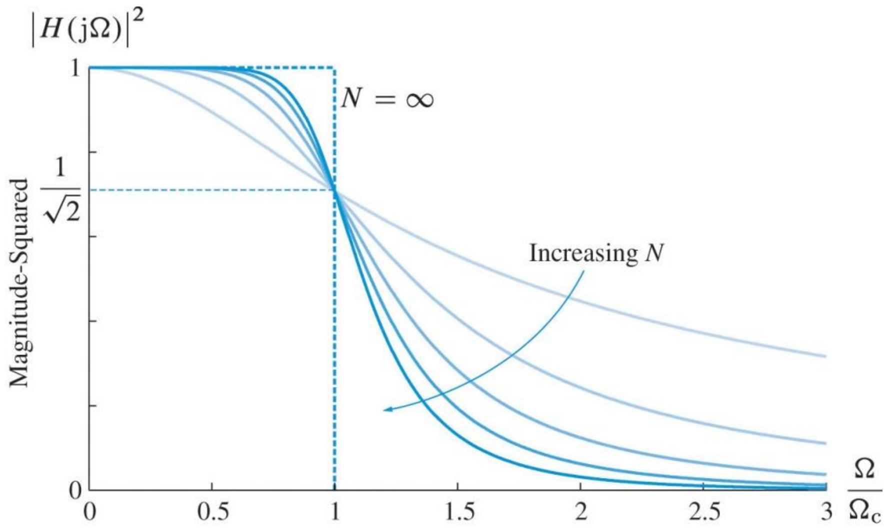
Omega measured in radians/sec

From Continuous Time to Discrete Time Filters

Performance of analogue prototype filters

Butterworth analogue filter

- Maximal flatness at Ω=0 and Ω=∞
- Maximum distortion at Ω= Ω_c
- Not very sharp cut-off

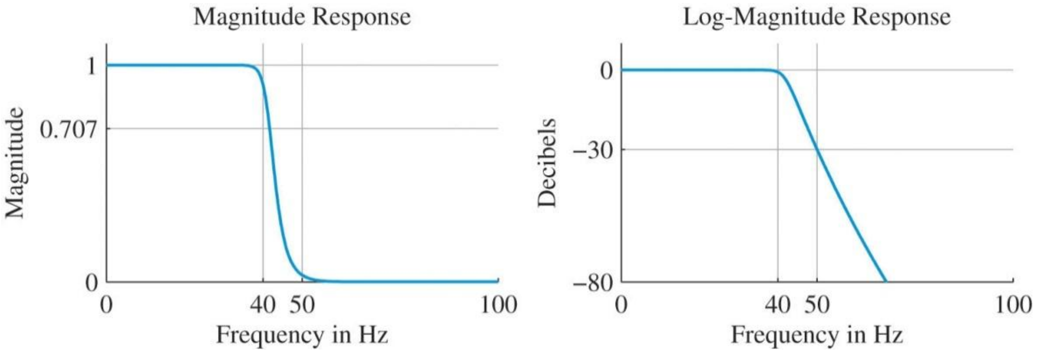


Example

F_p = 40Hz, A_p =1 dB,

F_s = 50Hz, A_s = 30dB

N = 19

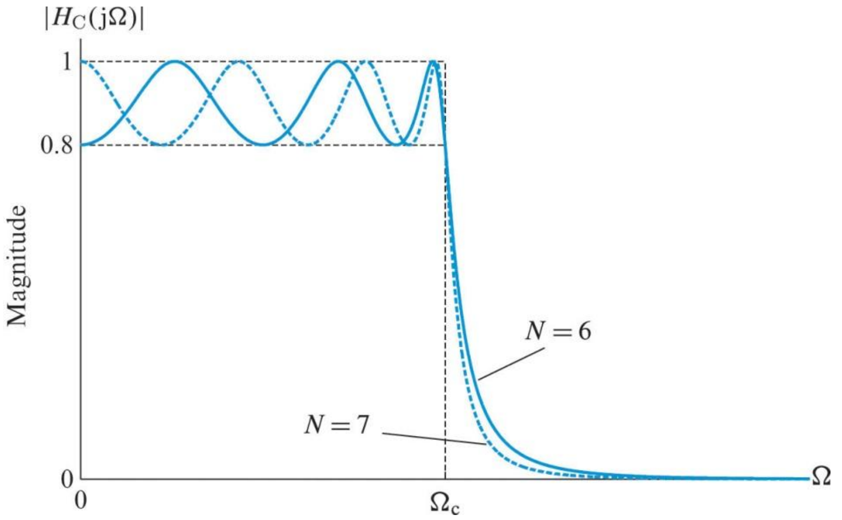


From Continuous Time to Discrete Time Filters

Performance of analogue prototype filters

Chebychev analogue filter

- Equiripple passband (type I)
- Equiripple stopband (type II)
- Sharper cut-off compared to Butterworth
- Lower order filter needed to satisfy the requirements

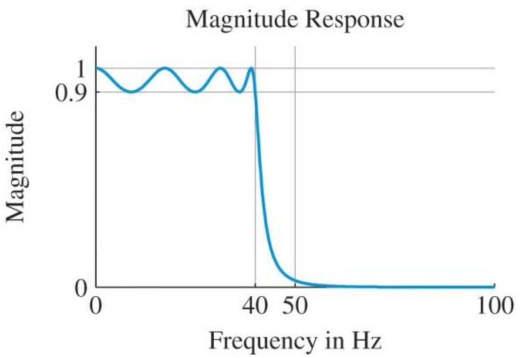


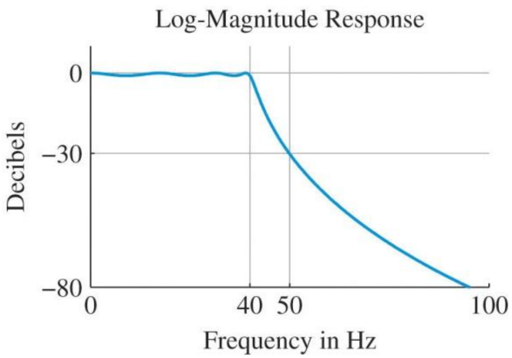
Example

F_p = 40Hz, A_p =1 dB,

F_s = 50Hz, A_s = 30dB

N = 7





Design of IIR Filters: Bilinear Transformation

From Continuous Time to Discrete Time Filters

Design of Digital IIR filters in Matlab

Butterworth response based digital IIR filter design

[N, Omegac] = buttord(Omegap, Omegas, Ap, As)

[b,a] = butter(N, Omegac)

No ‘s’ argument

Omega normalised (from 0 to 1)

1. Algorithm first finds the lowpass analogue prototype filter.

2. It then transforms the lowpass filter into a bandpass, highpass, or bandstop filter with desired cut-off frequencie(s)

3. For digital filter design, *butter* uses the *<bilinear>* Matlab command to convert analogue filter into a digital filter through bilinear transformation with frequency pre-warping.

Similar steps for digital IIR filter design based on other analogue prototype filters

Design of IIR Filters: Bilinear Transformation

Why Bilinear Transformation

Impulse Invariance Limitations

IIR Filter design through sampling of the continuous time impulse response

The frequency response of the continuous time filter has to be bandlimited otherwise aliasing will occur from sampling the impulse response

Impulse invariance cannot be used to map high-pass or band-stop continuous time designs to high-pass or band-stop discrete-time designs since high pass and band-stop continuous-time filters are not bandlimited

$\updownarrow$

The mapping of the $j\omega$ axis in the s-plane (analogue frequency) to the unit circle $e^{j\Omega}$ in the z-plane (digital frequency Ω) is many to one (the entire axis maps to multiple revolutions of the unit circle)

$$-\frac{\pi}{T} \le \omega \le \frac{\pi}{T} \rightarrow -\pi \le \Omega \le \pi \ , \ \frac{-\pi + 2\pi}{T} \le \omega \le \frac{\pi + 2\pi}{T} \rightarrow -\pi \le \Omega \le \pi ....$$

$$-\frac{(2k-1)\pi}{T} \le \omega \le \frac{(2k+1)\pi}{T} \rightarrow -\pi \le \Omega \le \pi$$

Design of IIR Filters: Bilinear Transformation

Why Bilinear Transformation

Impulse Invariance Limitations

Impulse Invariance Transformation

Many to one mapping (aliasing)

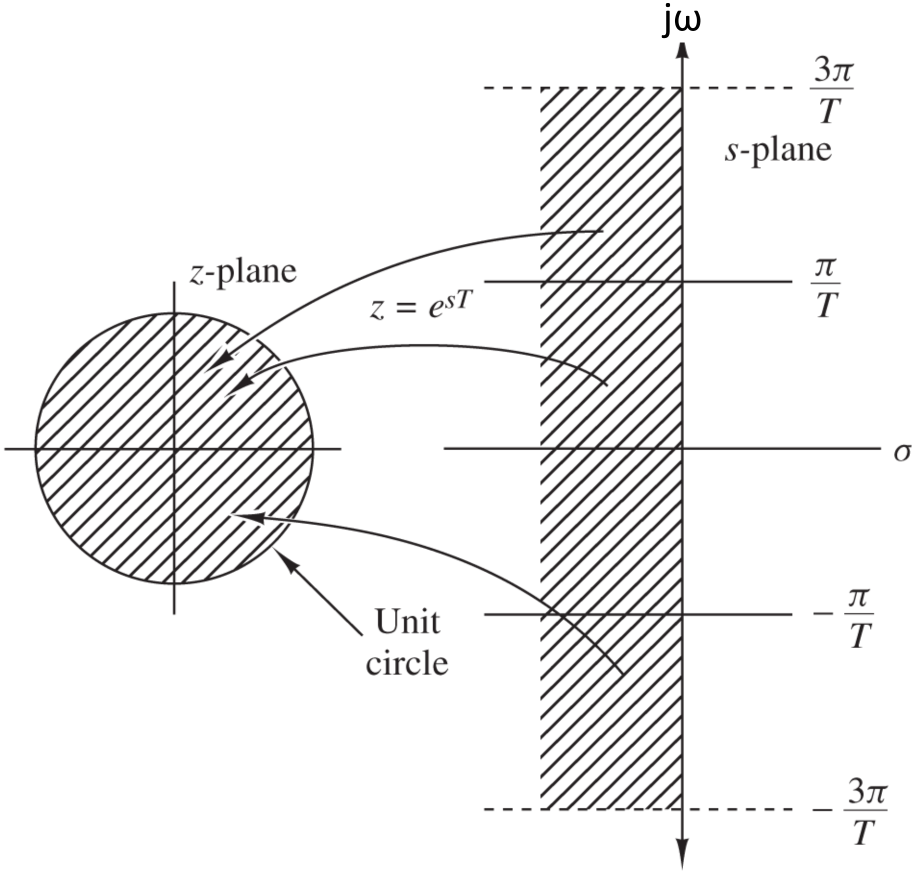
$$H_c(s) = \sum_{k=1}^{N} \frac{A_k}{s - s_k}$$

$$\downarrow$$

$$s = s_k \rightarrow z = e^{s_k T}$$

$$\downarrow$$

$$H(z) = \sum_{k=1}^{N} \frac{A_k}{1 - e^{s_k T} z^{-1}}$$



Design of IIR Filters: Bilinear Transformation

Solution: Bilinear Transformation

Map the entire *jω* axis to one revolution of the unit circle in the z-plane

Bilinear Transformation

Non-linear mapping needed

$$-\infty \le \omega \le \infty \rightarrow -\pi \le \omega \le \pi$$

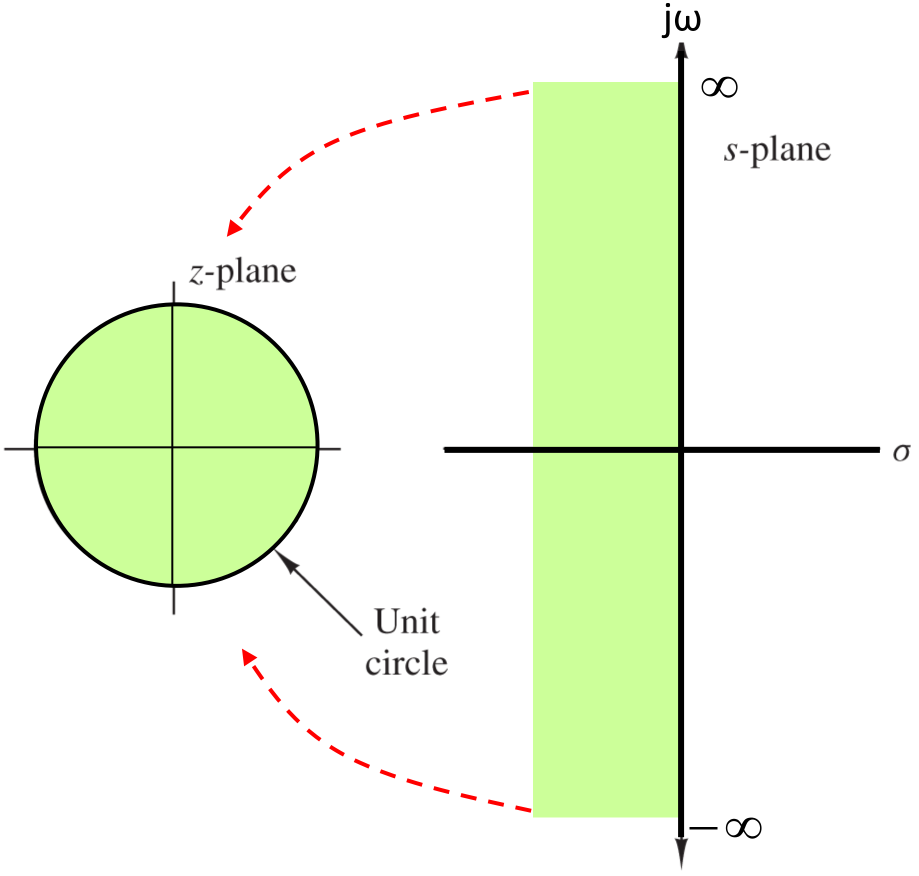
$$H_c(s) = \sum_{k=1}^{N} \frac{A_k}{s - s_k}$$

$$\downarrow$$

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

$$\downarrow$$

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



Bilinear Transformation

Properties

I. Stable continuous time filters map into stable discrete time filters (the left half *s*-plane maps inside the unit circle)

II. The $j\omega$ axis maps on the unit circle

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

$$s=\sigma+j\omega$$

$$\Rightarrow z=\frac{1+\sigma T/2+j\omega T/2}{1-\sigma T/2-j\omega T/2}$$

I. $\sigma<0\Rightarrow|z|<1$

II. $s=j\omega\Rightarrow\sigma=0\Rightarrow|z|=1$

Design of IIR Filters: Bilinear Transformation

Bilinear Transformation

Relationship between analogue frequency ω and digital frequency Ω

$$\omega = \frac{2}{T}\tan(\Omega/2) \Leftrightarrow \Omega = 2\arctan(\omega/2)$$

$$s = j\omega = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) = \frac{2}{T}\left(\frac{1-e^{-j\Omega}}{1+e^{-j\Omega}}\right) = \frac{2}{T}\left(\frac{e^{-j\Omega/2}(e^{j\Omega/2}-e^{-j\Omega/2})}{e^{-j\Omega/2}(e^{j\Omega/2}+e^{-j\Omega/2})}\right)$$

$$= \frac{2}{T}\left(\frac{e^{-j\Omega/2}(2j\sin(\Omega/2))}{e^{-j\Omega/2}(2\cos(\Omega/2))}\right) = \frac{2}{T}j\tan(\Omega/2) \Leftrightarrow s = j\omega = j\frac{2}{T}\tan(\Omega/2) \Rightarrow$$

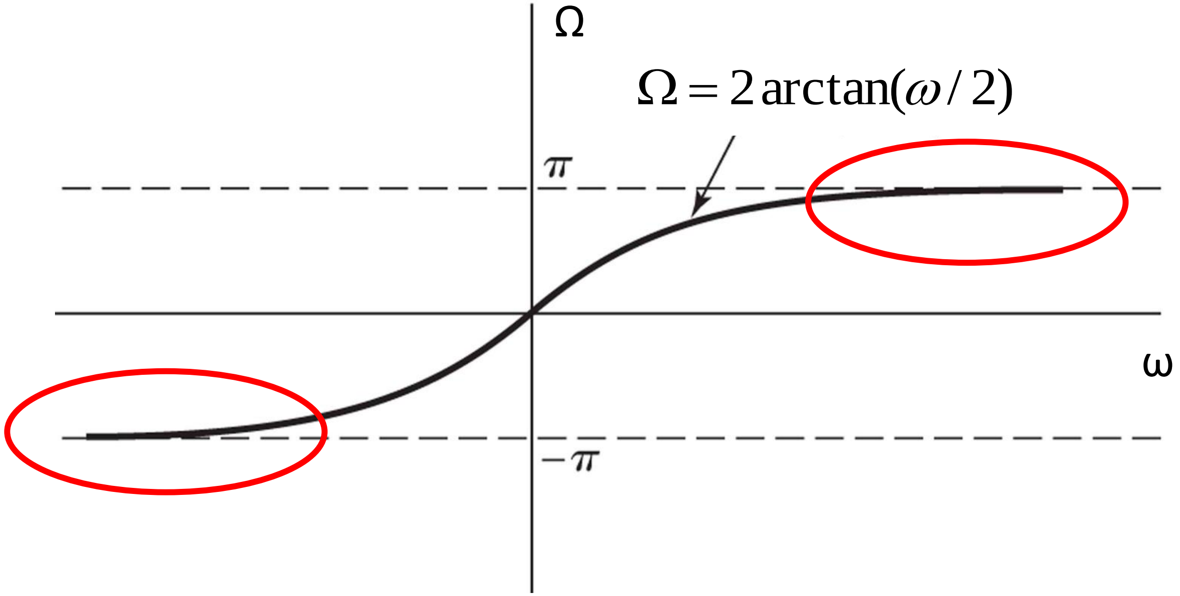
$$\omega = \frac{2}{T}\tan(\Omega/2) \quad \text{or} \quad \Omega = 2\arctan(\omega T/2)$$

Design of IIR Filters: Bilinear Transformation

Bilinear Transformation

Mapping of analogue frequency ω to digital frequency Ω

$$\omega = \frac{2}{T}\tan(\Omega/2) \Leftrightarrow \boxed{\Omega = 2\arctan(\omega/2)}$$



Frequency Warping : Non-linear compression of the frequency axis

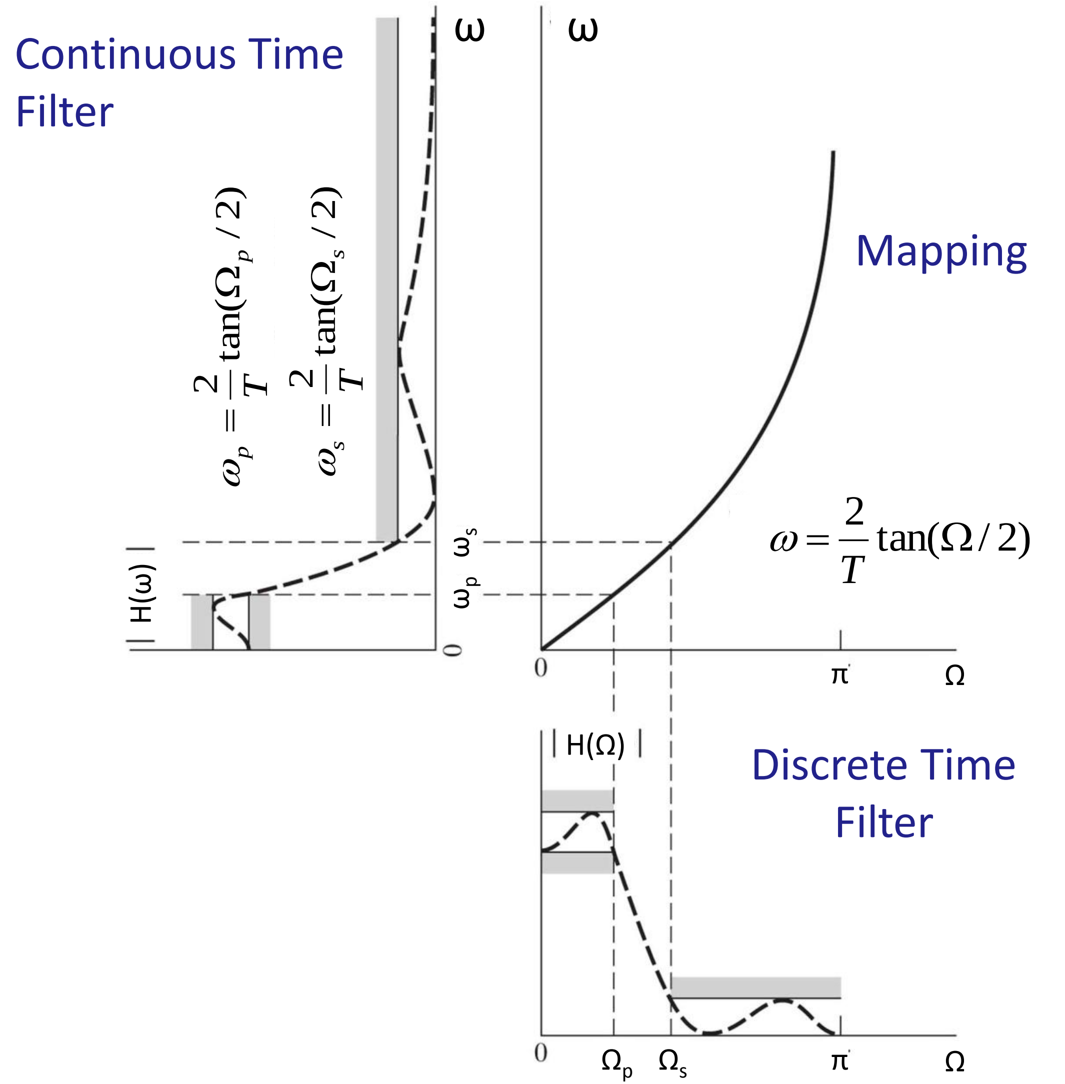
Design of IIR Filters: Bilinear Transformation

Frequency Warping - Solution

Pre-warping of analogue frequency

$$\omega = \frac{2}{T}\tan(\Omega/2)$$

Compensate for the warping effect that takes place when mapping analogue frequency to digital frequency by <u>pre-warping</u> the frequency design specifications of the discrete time filter using the above equation



Design of IIR Filters: Bilinear Transformation

Frequency Warping - Solution

Pre-warping of analogue frequency

1. Given Ω_s (the desired digital stop-band frequency)

2. Pre-warp using

$$\omega_s = \frac{2}{T}\tan(\Omega_s / 2)$$

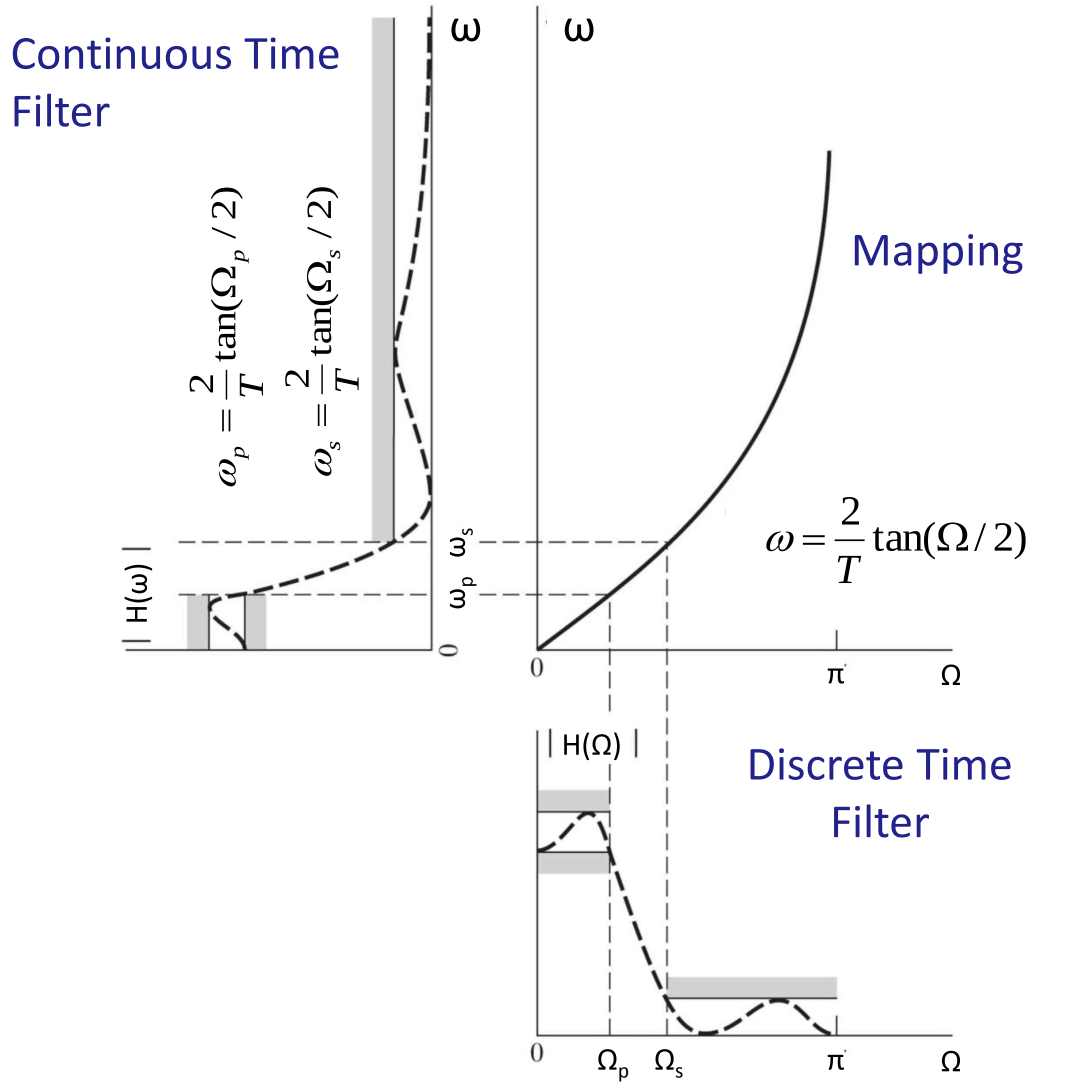
to get the analogue filter's required stop band frequency

3. so that when the bilinear transformation is applied

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

4. we get a digital filter with the desired stop-band

$$\Omega = 2\arctan(\omega_s / 2)$$



Design of IIR Filters: Bilinear Transformation

Design Procedure

1. Given Ω_{cd} calculate ω_{ca} incorporating frequency axis pre-warping as appropriate.

---

2. Select analogue filter prototype $H_a(s)$ to satisfy design spec (Butterworth, Chebychev, etc.)

---

3. Apply frequency transformation

---

4. Evaluate H(s) and apply the bilinear transform to obtain H(z)

Ω_{cd} *Digital filter cut-off frequency* , ω_{ca} *Analogue filter cut-off frequency,* $H_a(s)$ Analogue filter system function

Design of IIR Filters: Bilinear Transformation

Design Procedure - Transformations

Frequency axis pre-warping	$\omega_{ca} = \frac{2}{T}\tan(\Omega_{cd}/2) = \frac{2}{T}\tan(\omega_{cd}T/2)$
Frequency transformation	Low pass -> low pass : $s \rightarrow s/\omega_{ca}$
	Low pass -> high pass : $s \rightarrow \omega_{ca}/s$
	Low pass -> band pass : $s \rightarrow \frac{s^2 + \omega_{ca1}\omega_{ca2}}{\omega_{ca2} - \omega_{ca1}}$
Bilinear transformation	$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} = \frac{2}{T}\frac{z-1}{z+1}$

Design of IIR Filters: Bilinear Transformation

Example 1

Design a low-pass digital filter with $\mathbf{f_{cd}}$**=100Hz** and **T=1.6ms** using bilinear transform, based on a 2nd order Butterworth response (with a normalised cut-off frequency):

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

1. Pre-warp frequency axis: $\omega_{ca} = \frac{2}{T}\tan\left(\frac{\omega_{cd}T}{2}\right) = 687.2 rads^{-1}$

---

2. Apply LP to LP frequency transformation: $s \rightarrow s/\omega_{ca}$ $\Leftrightarrow H(s) = \frac{\omega_{ca}^2}{s^2 + \omega_{ca}\sqrt{2}s + \omega_{ca}^2}$

$$H(s) = \frac{1}{\left(\frac{s}{687.2}\right)^2 + \frac{\sqrt{2}s}{687.2} + 1} = \frac{472243.84}{s^2 + 971.85s + 472243.84}$$

---

3. Apply the bilinear transform: $s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$

$$H(z) = \frac{472243.84}{156.25*10^4\left[\frac{1-z^{-1}}{1+z^{-1}}\right]^2 + 1214812.5\left[\frac{1-z^{-1}}{1+z^{-1}}\right] + 472243.84} \Leftrightarrow H(z) = \frac{1+2z^{-1}+z^{-2}}{6.88 - 4.62z^{-1} + 1.74z^{-2}}$$

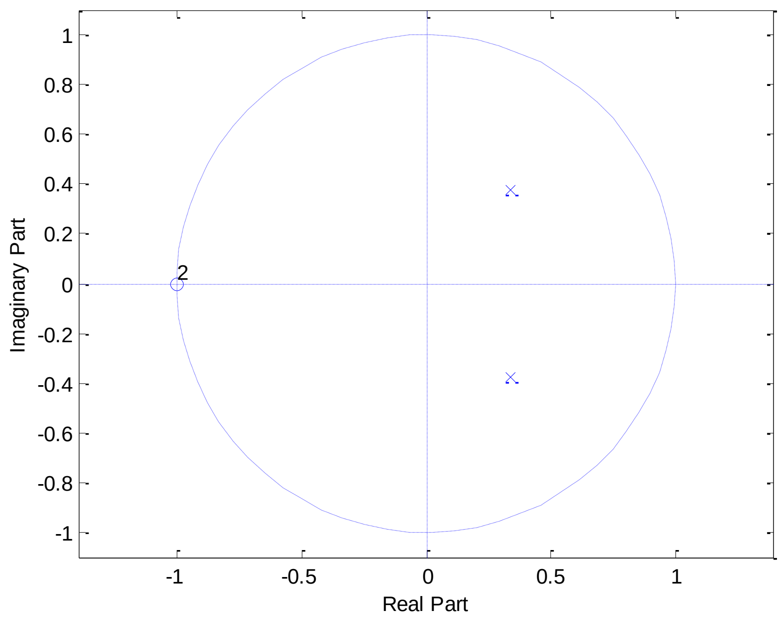
Design of IIR Filters: Bilinear Transformation

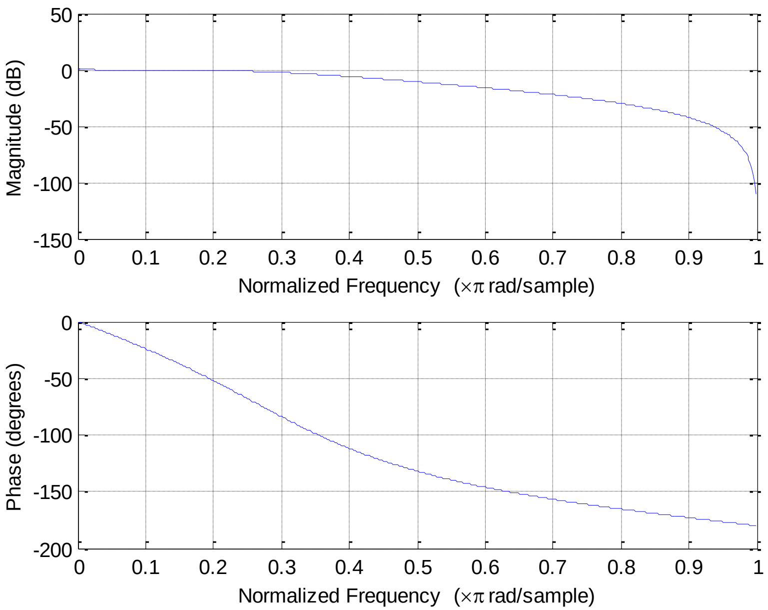
Example 1

Design a low-pass digital filter with $\mathbf{f_{cd}=100Hz}$ and **T=1.6ms** using bilinear transform, based on a 2nd order Butterworth response (with a normalised cut-off frequency):

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

$$H(z)=\frac{1+2z^{-1}+z^{-2}}{6.88-4.62z^{-1}+1.74z^{-2}}$$





Design of IIR Filters: Bilinear Transformation

Example 1

Design a low-pass digital filter with **f_{cd}=100Hz** and **T=1.6ms** using bilinear transform, based on a 2nd order Butterworth response (with a normalised cut-off frequency):

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

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{6.88 - 4.62z^{-1} + 1.74z^{-2}}$$

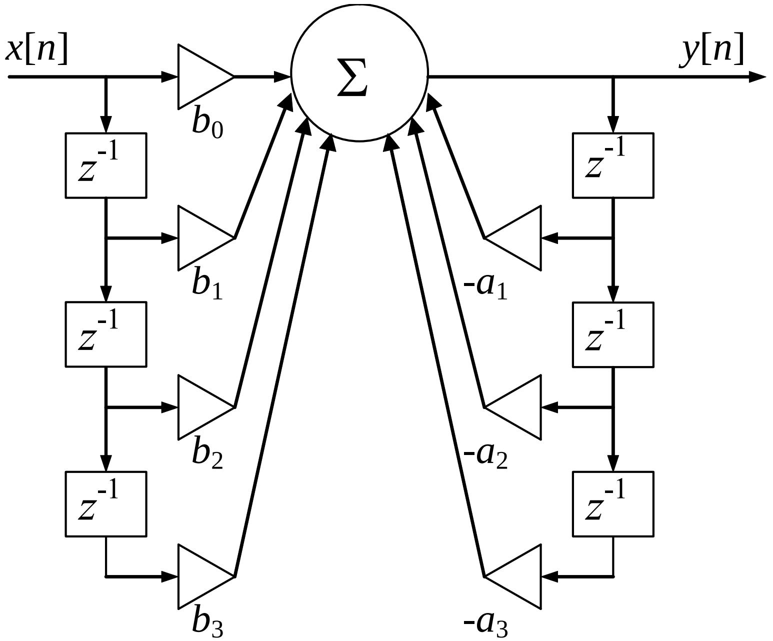
Implementation

Need to express system function in terms of difference equations

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \quad with\ a_0 = 1$$

$$y[n] = \sum_{k=0}^{N} b_k x[n-k] - \sum_{k=1}^{N} a_k y[n-k]$$

General IIR digital filter implementation form



Example 1

Design a low-pass digital filter with $\mathbf{f_{cd}=100Hz}$ and **T=1.6ms** using bilinear transform, based on a 2nd order Butterworth response (with a normalised cut-off frequency):

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

$$H(z)=\frac{1+2z^{-1}+z^{-2}}{6.88-4.62z^{-1}+1.74z^{-2}}$$

$$H(z)=\frac{Y(z)}{X(z)}=\frac{1+2z^{-1}+z^{-2}}{6.88-4.62z^{-1}+1.74z^{-2}}=\frac{\dfrac{1+2z^{-1}+z^{-2}}{6.88}}{\dfrac{6.88-4.62z^{-1}+1.74z^{-2}}{6.88}}$$

$$\frac{Y(z)}{X(z)}=\frac{0.145+0.291z^{-1}+0.145z^{-2}}{1-0.671z^{-1}+0.253z^{-2}}\Leftrightarrow(1-0.671z^{-1}+0.253z^{-2})Y(z)=(0.145+0.291z^{-1}+0.145z^{-2})X(z)$$

$$Y(z)-0.671z^{-1}Y(z)+0.253z^{-2}Y(z)=0.145X(z)+0.291z^{-1}X(z)+0.145z^{-2}X(z)$$

$$y[n]=0.145x[n]+0.291x[n-1]+0.145x[n-2]+0.671y[n-1]-0.253y[n-2]$$

Design of IIR Filters: Bilinear Transformation

Example 2

Design a low-pass digital filter with **f_{cd}=200Hz** and **f_s=1200Hz** using bilinear transform, based on a 2nd order Butterworth response (with a normalised cut-off frequency):

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

1. Apply LP to LP frequency transformation: $s \to s/\omega_{ca}$

$$H(s)=\frac{\omega_{ca}^2}{s^2+\omega_{ca}\sqrt{2}s+\omega_{ca}^2}$$

2. Apply the bilinear transform: $s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$ $\Leftrightarrow H(z)=H\left(s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right)$

$$H(z)=\frac{\omega_{ca}^2}{\frac{4}{T^2}\frac{(1-z^{-1})^2}{(1+z^{-1})^2}+\frac{2\omega_{ca}\sqrt{2}}{T}\frac{(1-z^{-1})}{(1+z^{-1})}+\omega_{ca}^2}=\frac{\omega_{ca}^2T^2(1+z^{-1})^2}{4(1-z^{-1})^2+2\omega_{ca}T\sqrt{2}(1-z^{-1})(1+z^{-1})+\omega_{ca}^2T^2(1+z^{-1})^2}$$

$$=\frac{\omega_{ca}^2T^2(1+2z^{-1}+z^{-2})}{4(1-2z^{-1}+z^{-2})+2\omega_{ca}T\sqrt{2}(1-z^{-2})+\omega_{ca}^2T^2(1+2z^{-1}+z^{-2})}$$

$$=\frac{\omega_{ca}^2T^2(1+2z^{-1}+z^{-2})}{4+2\omega_{ca}T\sqrt{2}+\omega_{ca}^2T^2+(-8+2\omega_{ca}^2T^2)z^{-1}+(4-2\omega_{ca}T\sqrt{2}+\omega_{ca}^2T^2)}$$

Example 2

Design a low-pass digital filter with **f_{cd}=200Hz** and **f_s=1200Hz** using bilinear transform, based on a 2nd order Butterworth response (with a normalised cut-off frequency):

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

3. Frequency pre-warping: $\omega_{ca}=\frac{2}{T}\tan(\Omega_{cd}/2)$

$$\omega_{ca}T=2\tan(\pi f_{cd}/f_s)=2\tan(\pi 200/1200)=2\tan(\pi/6)=\frac{2\sqrt{3}}{3}$$

$$H(z)=\frac{\omega_{ca}^2T^2(1+2z^{-1}+z^{-2})}{4+2\omega_{ca}T\sqrt{2}+\omega_{ca}^2T^2+(-8+2\omega_{ca}^2T^2)z^{-1}+(4-2\omega_{ca}T\sqrt{2}+\omega_c^2T^2)}$$

$$=\frac{1.3333+2.6667z^{-1}+1.3333z^{-2}}{8.5993-5.3333z^{-1}+2.0673z^{-2}}\Leftrightarrow H(z)=\frac{0.1551+0.3101z^{-1}+0.1551z^{-2}}{1-0.6202z^{-1}+0.2404z^{-2}}$$

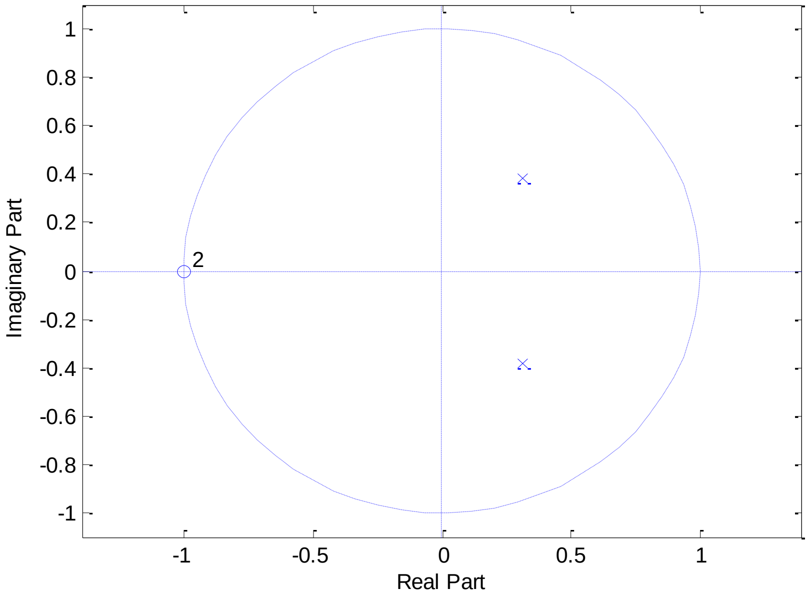
Design of IIR Filters: Bilinear Transformation

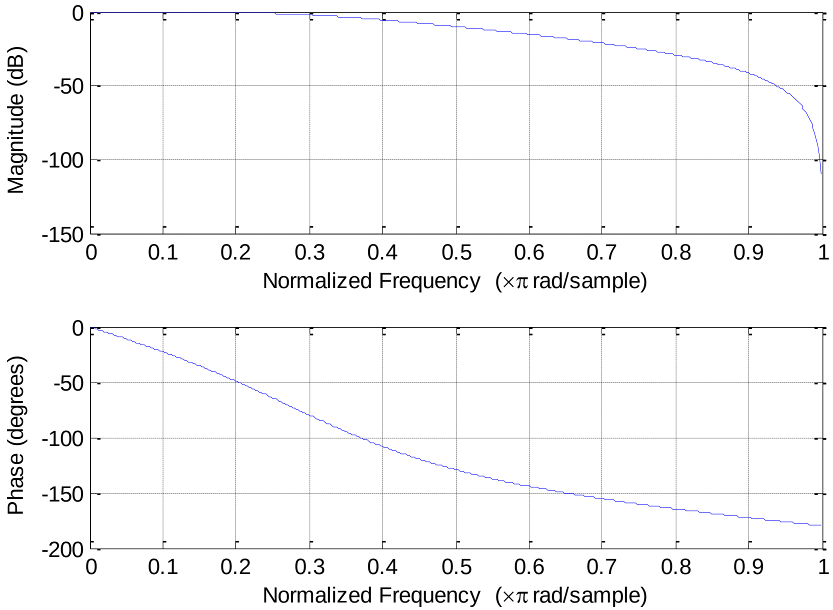
Example 2

Design a low-pass digital filter with $\mathbf{f_{cd}=200Hz}$ and $\mathbf{f_s=1200Hz}$ using bilinear transform, based on a 2nd order Butterworth response (with a normalised cut-off frequency):

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

$$H(z)=\frac{0.1551+0.3101z^{-1}+0.1551z^{-2}}{1-0.6202z^{-1}+0.2404z^{-2}}$$





Design of IIR Filters: Bilinear Transformation

Bilinear Transformation Disadvantages

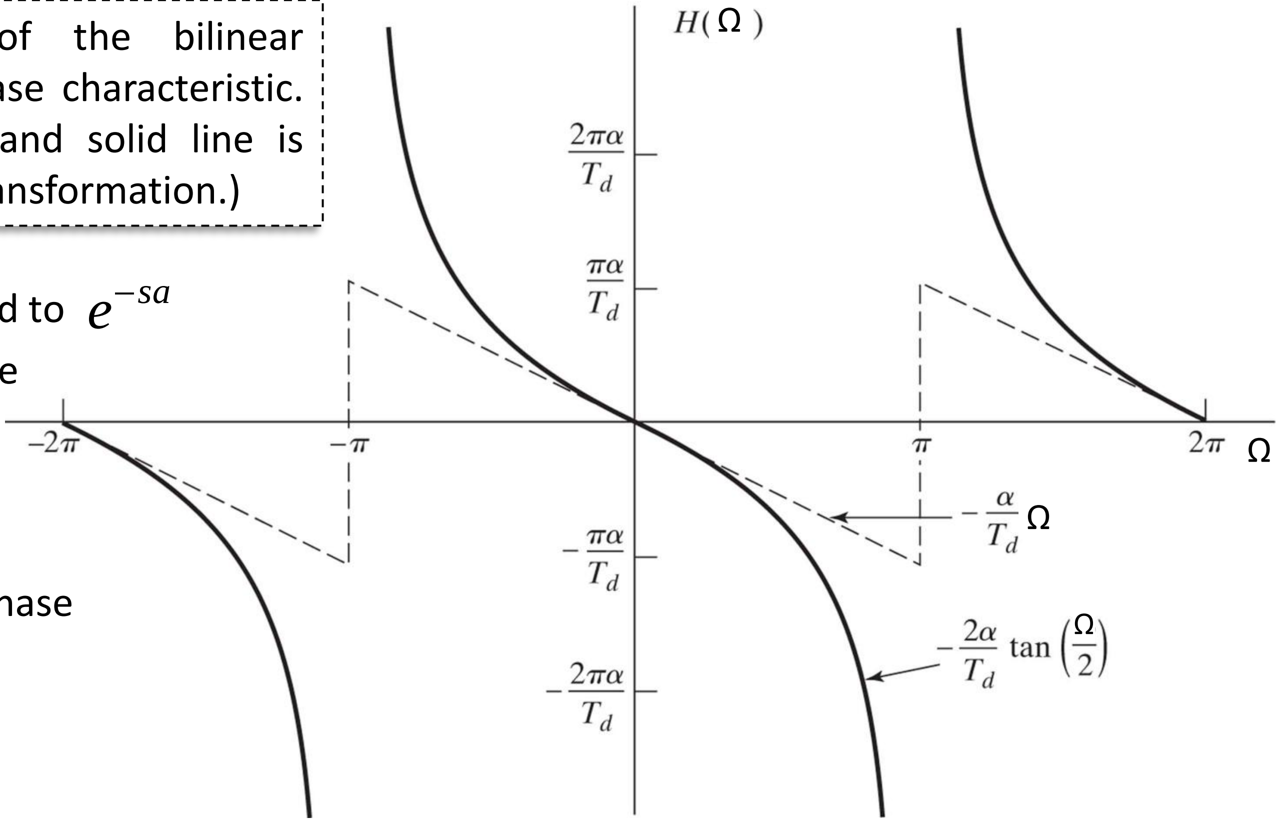
– Frequency axis warping

– **Distortion of phase characteristics**

– Distortion of time domain characteristics

Illustration of the effect of the bilinear transformation on a linear phase characteristic. (Dashed line is linear phase and solid line is phase resulting from bilinear transformation.)

- Bilinear Transformation applied to e^{-sa} and evaluated on the unit circle
- The resulting phase angle is $-(2a/T)\tan(\Omega/2)$
- Compared to periodic linear-phase function $-(\Omega a/T)$



Direct Optimisation

Design Procedure

1. Specify the design objectives
   - e.g. mini-max ripple in pass-band and or stop-band
2. Choose the filter structure and parameters
   - e.g. a 2nd order filter has parameters a_0, a_1, a_2, b_1 and b_2.
3. Select an appropriate optimisation tool to find the optimal or sub-optimal set of filter parameters.
   - e.g. least squares fitting - Matlab function yulewalk

Design of IIR Filters: Bilinear Transformation

Direct Optimisation

Matlab *yulewalk* function

- **yulewalk** designs recursive IIR digital filters using a least-squares fit to a specified frequency response.
- **[b,a] = yulewalk(n,f,m)** returns row vectors **b** and **a** containing the n+1 coefficients of the order **n** IIR filter whose frequency-magnitude characteristics approximately match those given in vectors **f** and **m**:
- **f** is a vector of frequency points, specified in the range between 0 and 1, where 1 corresponds to half the sample frequency (the Nyquist frequency). The first point of f must be 0 and the last point 1, with all intermediate points in increasing order.
- **m** is a vector containing the desired magnitude response at the points specified in f.
- f and m must be the same length.
- plot(f,m) displays the desired frequency response.