DSP: Part II

Lecture.4.2

Infinite Impulse Response (IIR) Filter

Methodology for IIR

1) What is the characteristics of an IIR Filter
   - Its impulse response, its difference equation, its H(z), and stability

2) Analysis: From difference equation to H(z), to pole/zero plot and then to frequency response H(ω)

3) IIR Filters are usually designed from analog filters, so lets examine H(s) the Laplace domain.

4) Design an analog filter
   a) Simple RC analog filter
   b) Butterworth analog filter
5) Design an IIR filter by converting from H(s) to H(z)
   a) Bilinear transform method
   b) Impulse invariance method.
6) Designing of IIR Filters using Matlab.

Review what is H(s) and H(z)

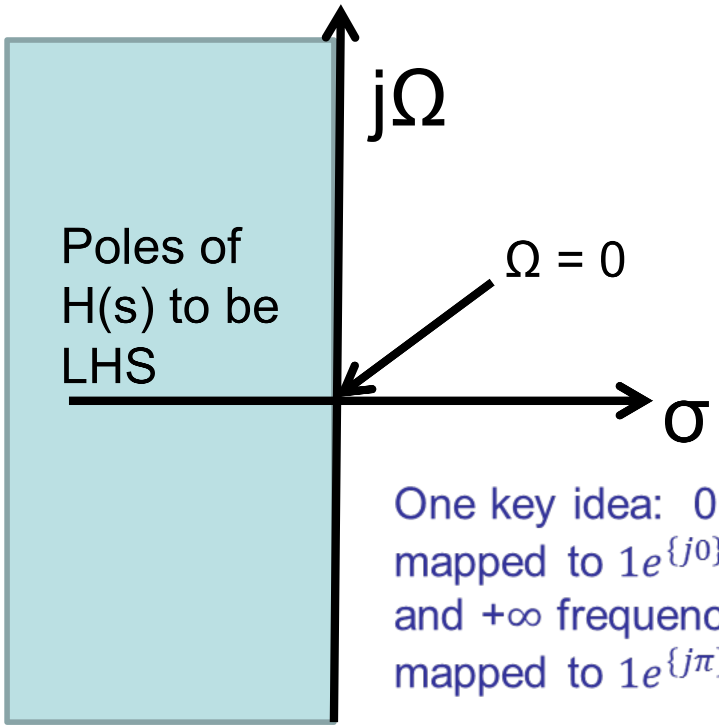
- If h(t) is the continuous time causal impulse response of a filter, then the transfer function of the filter in the Laplace domain is H(s), i.e. LT(h(t)) = H(s). For stability, all poles of H(s) must be on the left hand side of the imaginary axis. The imaginary axis represents the analog frequencies and goes from $0..\infty$

- If h(n) is digital filter's impulse response, then the z-transform of h(n) is H(z). For stability, all poles of H(z) must be inside the unit circle. The unit circle represents the digital frequency from $0..2\pi$.

- Note: one would naturally think that h(n) is a sampled version of h(t)! Such a conversion method is called impulse invariance (meaning same impulse). Although this is a possible way to get h(n), and hence an approximation of the desired transfer function H(z), its not a good way, or the only way. Lets proceed first with what is a widely used approach to get H(z)!

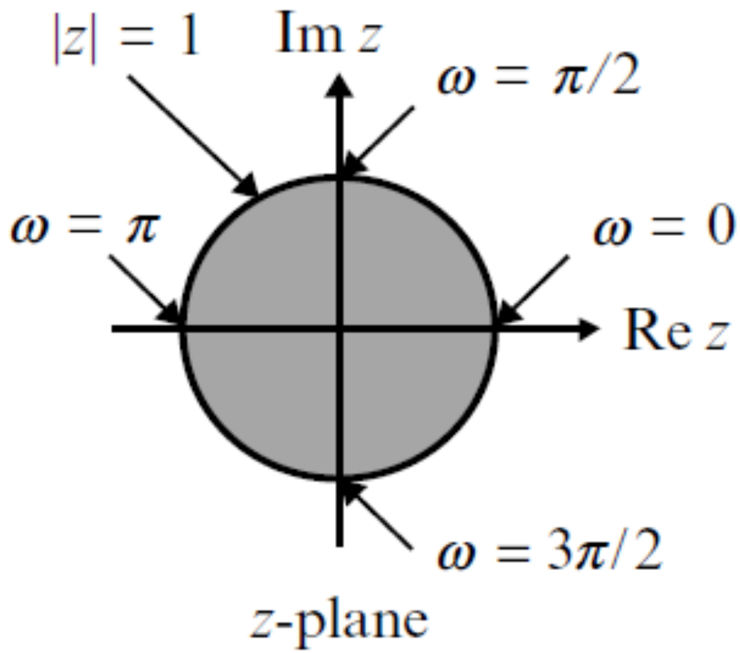
H(s) and H(z)
We want to map LHS of H(s) to inside unit circle for H(z)

Poles of H(z) to be inside unit circle!



One key idea: 0Hz in H(s) is mapped to $1e^{\{j0\}}$ for H(z), and +∞ frequency is mapped to $1e^{\{j\pi\}}$

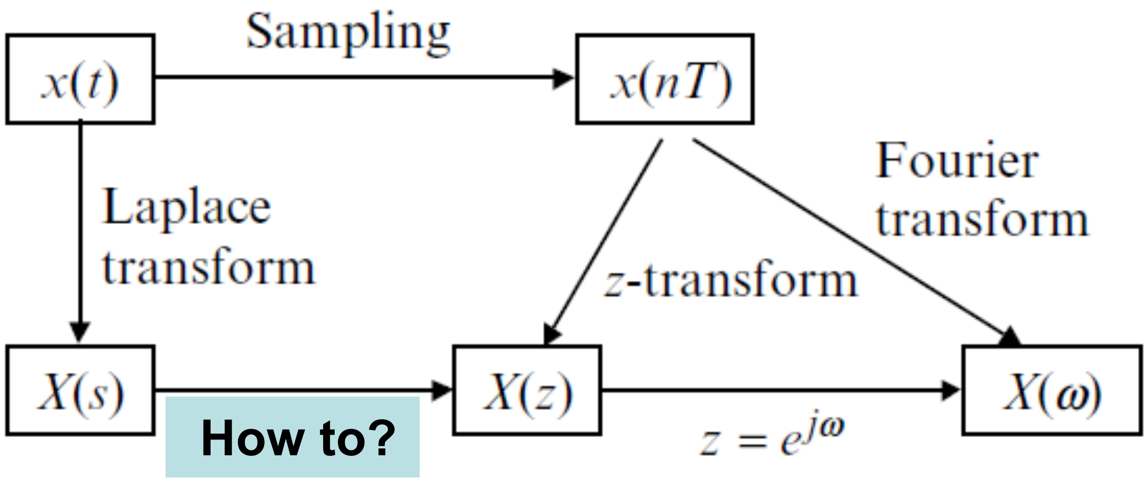
$s = \sigma + j\Omega$



H(s) is a rational function, an algebraic fraction such that both the numerator and the denominator are polynomials of s

H(z) is a rational function, such that both the numerator and the denominator are polynomials of z (complex variable)

Relating H(s) to H(z)



Relationships between Laplace Fourier and z-transforms

Our focus for digital IIR filter design is to design H(s) and then find a way to convert it into H(z). For the first method: bilinear transform.

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

'T' is a parameter that only has historical significance, the effect of ' T' value will cancel when the specification is mapped from digital filter specification to continuous domain and back again.

BILINEAR TRANSFORM

Given H(s), we can transform to H(z) by substituting ‘s’ of H(s) by the following:

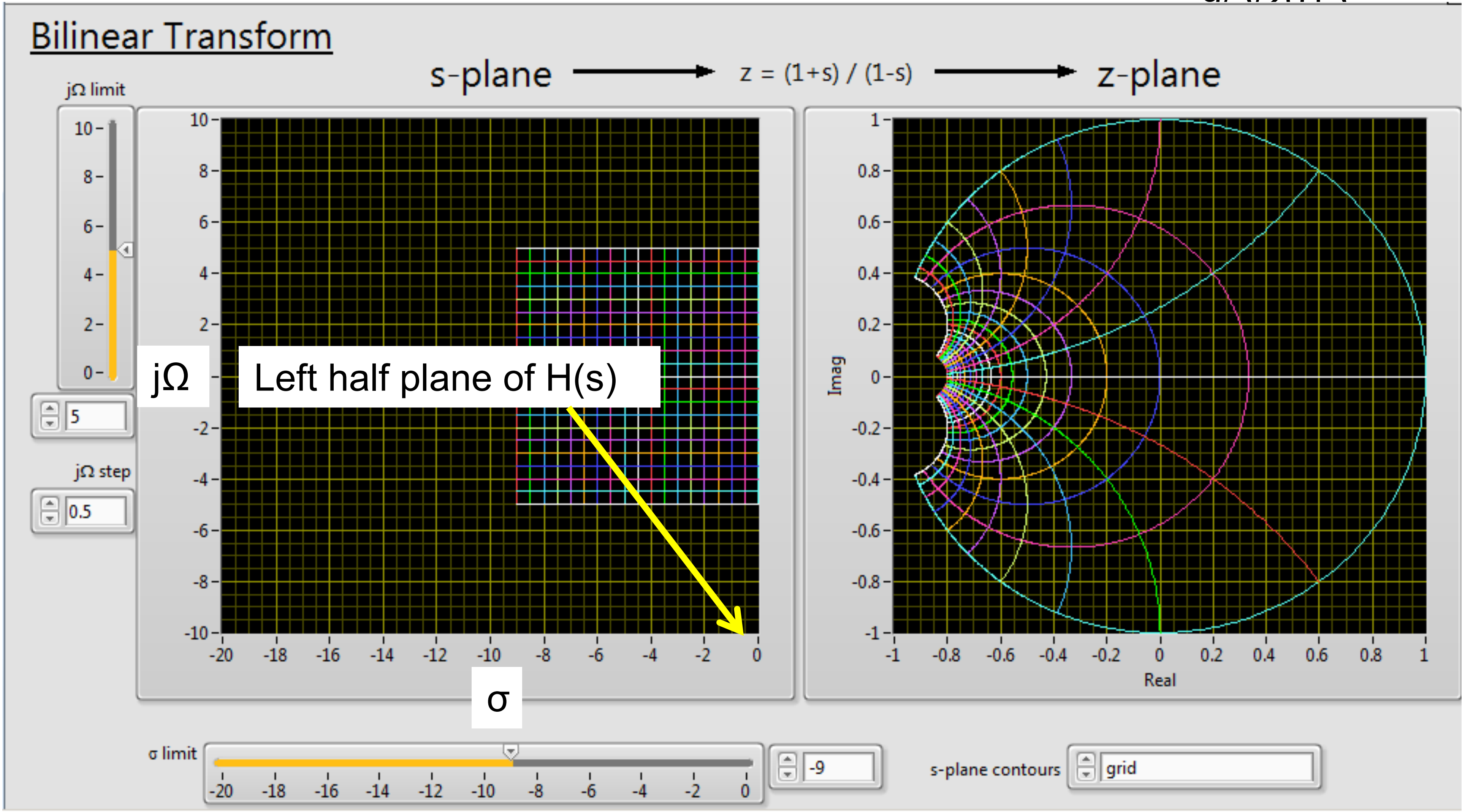
$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right) = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

‘T’ is a parameter that only has historical significance, the effect of ‘ T’ value will cancel when the specification is mapped from digital filter specification to continuous domain and back again.

- This transformation basically maps lines in s-domain to circles in z-domain.
    - It will map the entire left half plane of H(s) inside the unit circle of H(z).
    - It will map the entire right half plane of H(s) outside the unit circle of H(z)
    - It will map the jΩ axis of H(s) to the unit circle of H(z).
- For analog filter to be stable, poles ONLY are on left half planes of H(s), these poles will be mapped inside the unit circle of H(z) by the bilinear transform and hence the transformed H(z) will be stable!
- Zeros if present can be on either side of H(s) as its location does not affect stability of analog filters and hence it can be mapped inside or outside of unit circle in H(z) and its location do not affect stability of digital filters.

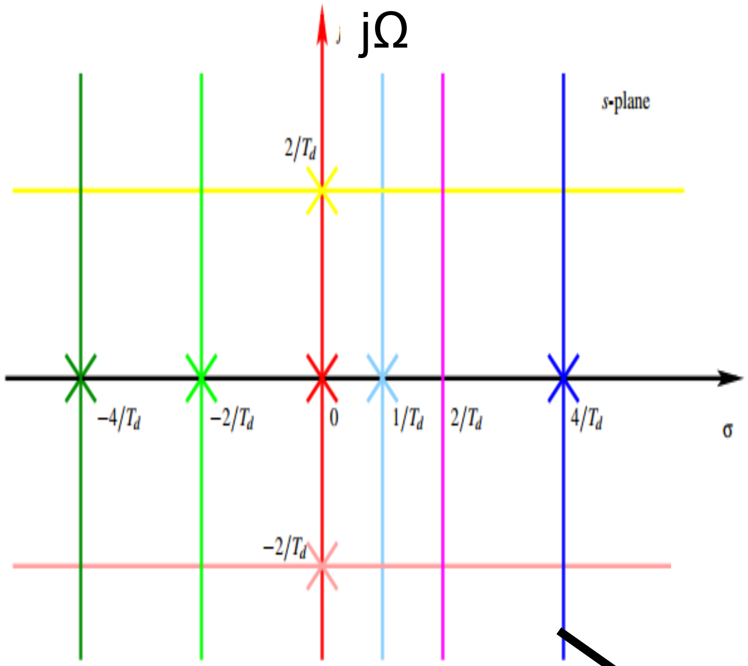
Conformal mapping , bilinear Transform, mobius transformation

9/3/2013

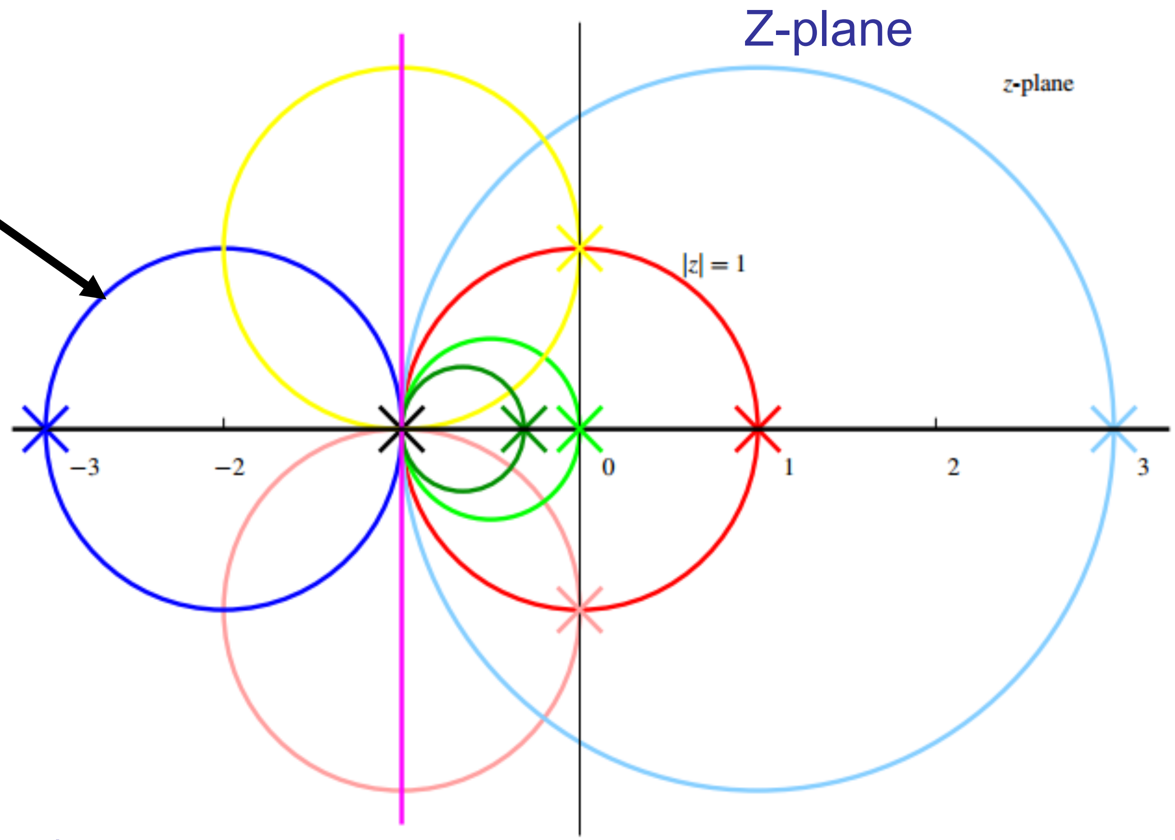


Visualizing the bilinear transform: https://decibel.ni.com/content/docs/DOC-23336

Mapping H(s) to H(z) by bilinear transformation



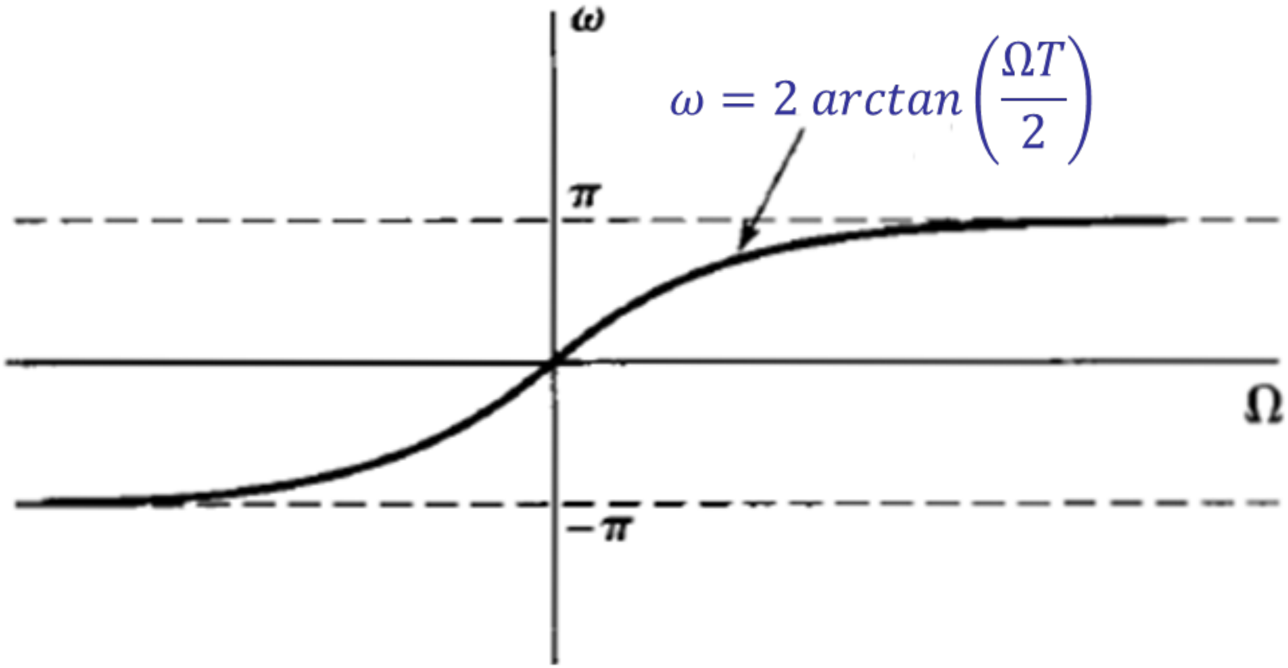
S-plane



Example in: Prof Bilmes (EE518) U.Washington
http://melodi.ee.washington.edu/courses/ee518/notes/lec16.pdf

Frequency warping of bilinear transformation

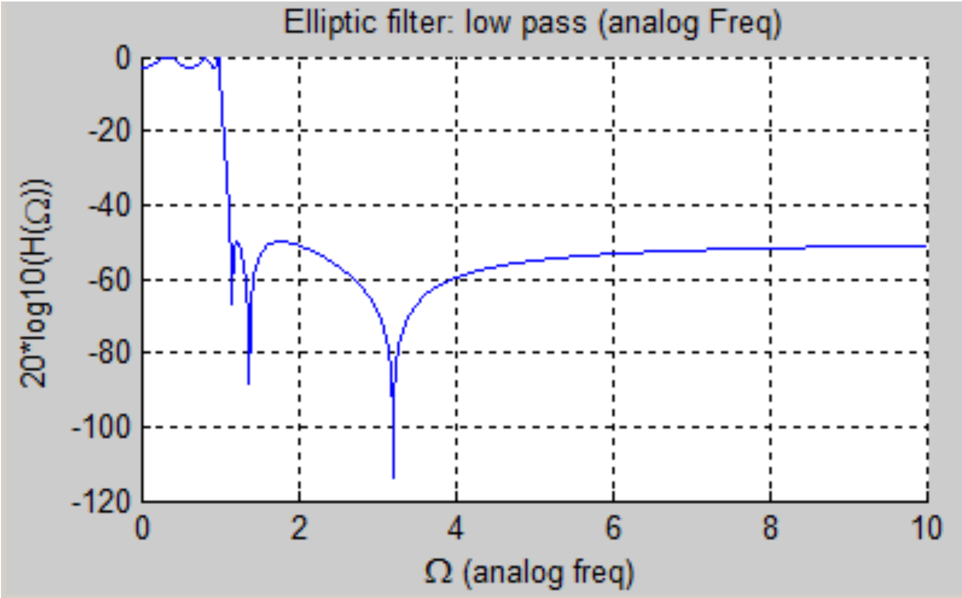
- The bilinear transformation from ‘s’ to the ‘z’ plane maps the jΩ (frequency axis) of the s-plane from 0..∞ radians (y-axis) to the upper half of the unit circle 0..π. Corollary, the analog frequency 0..-∞ radians is mapped to the lower half of the unit circle (z-plane).

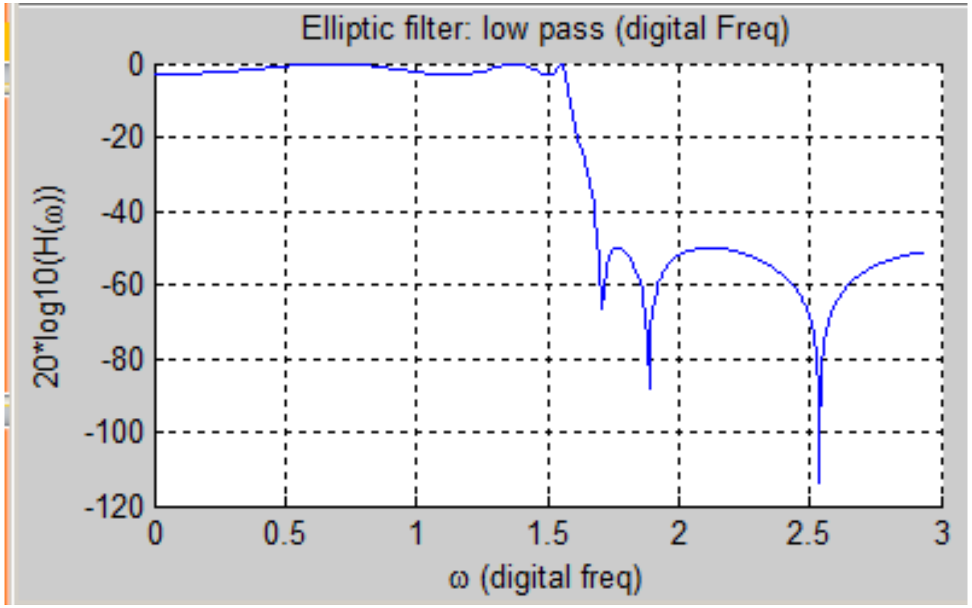


$$\Omega = \frac{2}{T}\tan\left(\frac{\omega}{2}\right)$$

Again, this parameter ‘T’ only has historical significance, the effect of T value will cancel when the specification is mapped from digital filter specification to continuous domain and back again. Hence, you can choose a simple ‘T=2’ !

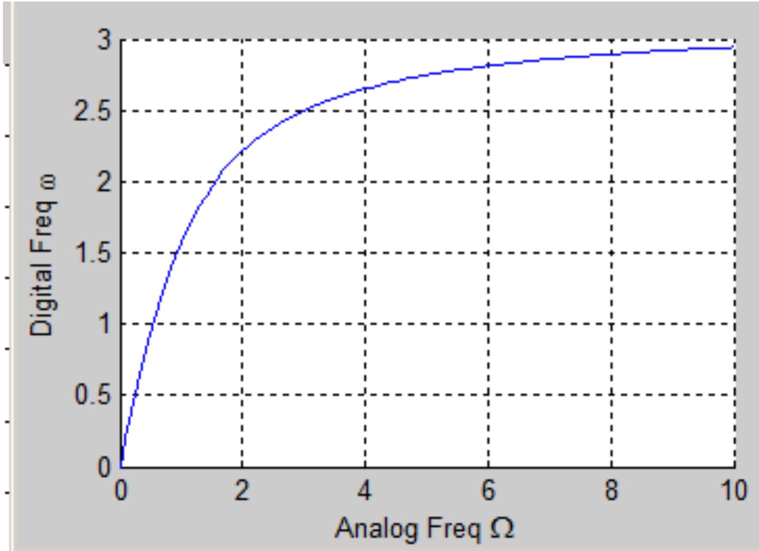
What happen during frequency warping?





The analog frequency is warped

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



Matlab: freq_warp_ex1.m

Note that the cut-off at analog freq has changed! After freq warping!

Bilinear Transformation Design Procedure Summary

0) Given a filter specification in the analog domain
1)Make filter specification in the digital domain.
2)Map the specification in the analog domain again to take care of frequency warping of bilinear transform.

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

3)Design the H(s) analog filter, e.g, using Butterworth filter design
4)Transform H(s)->H(z) by bilinear transform method to get digital filter

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

BILINEAR TRANFORMATION: Example

- Lets design a digital filter given the spec: the analog signal is sampled at 1000Hz, and a low pass filter with cutoff frequency of 400Hz is desired. Design an IIR filter using bilinear transformation from a 1st order Butterworth analog filter H(s).

- Note that you will need to use pre-warping and bilinear transform equation:

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

Pre-warp

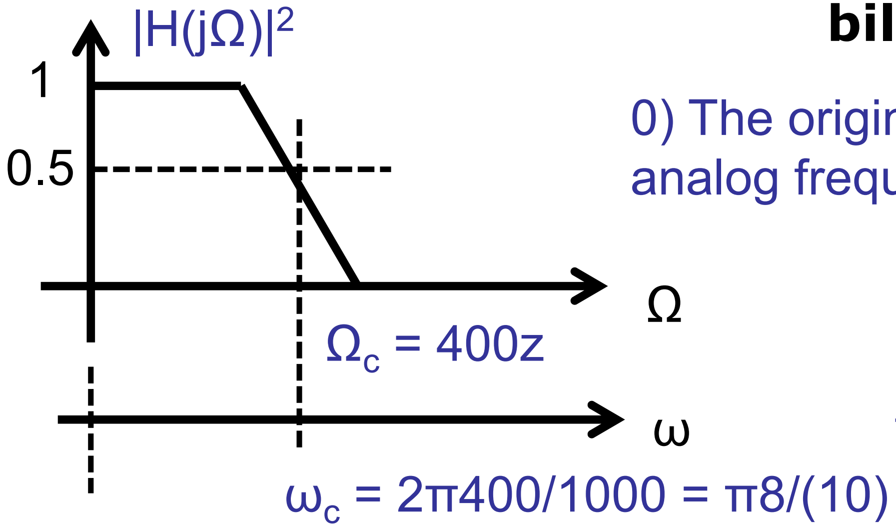
$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right) = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

Bilinear Transform

the variable ‘T’ in above will cancel.

In the following example, we retain the value of T in the equations and showed that it cancels in step 4

Example: bilinearTransform_ex2.m



0) The original specification in analog frequency

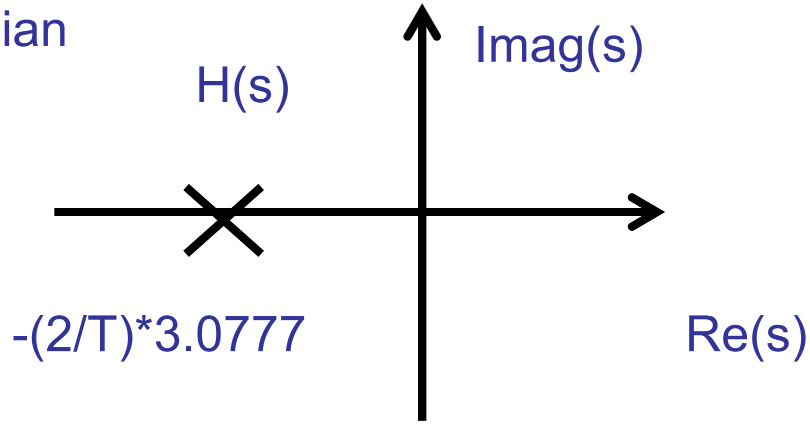
1) The specification in digital frequency

2) Find the pre-warp frequency to get the analog cutoff frequency to design the required Butterworth filter . Assuming T = 2; then

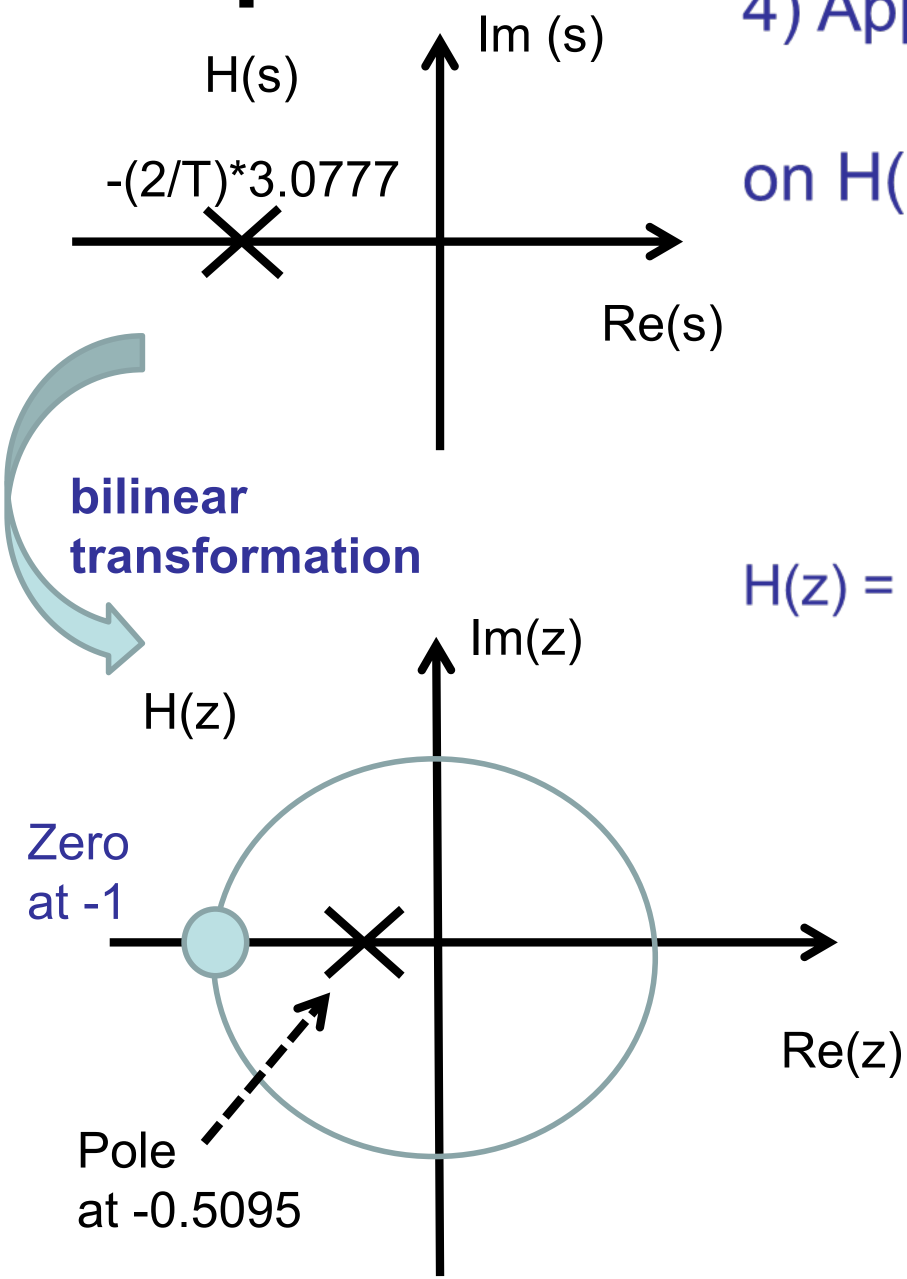
$\Omega = (2/T)*\tan(\pi 8/(10)) = (2/T)*3.0777$

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

(3) We want a 1st order Butterworth filter with cutoff frequency at (2/T)*3.0777 radian



Example



4) Applying bilinear transformation

on H(s) $= \dfrac{\Omega_c}{s+\Omega_c} = \dfrac{\left(\frac{2}{T}\right)*3.0777}{s+\left(\frac{2}{T}\right)*3.0777}$

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

H(z) $= \dfrac{\left(\frac{2}{T}\right)*3.0777}{\left(\frac{2}{T}\right)\frac{(z-1)}{z+1}+\left(\frac{2}{T}\right)*3.0777} = \dfrac{3.0777}{\left(\frac{z-1}{z+1}\right)+3.0777}$

$$= \frac{0.7548\,(z+1)}{z+0.5095}$$

Observation: a single pole in H(s) after bilinear transform introduces a zero as well in pi, and at -0.5095

Matlab Example

Matlab: bilinearTransform_ex2.m

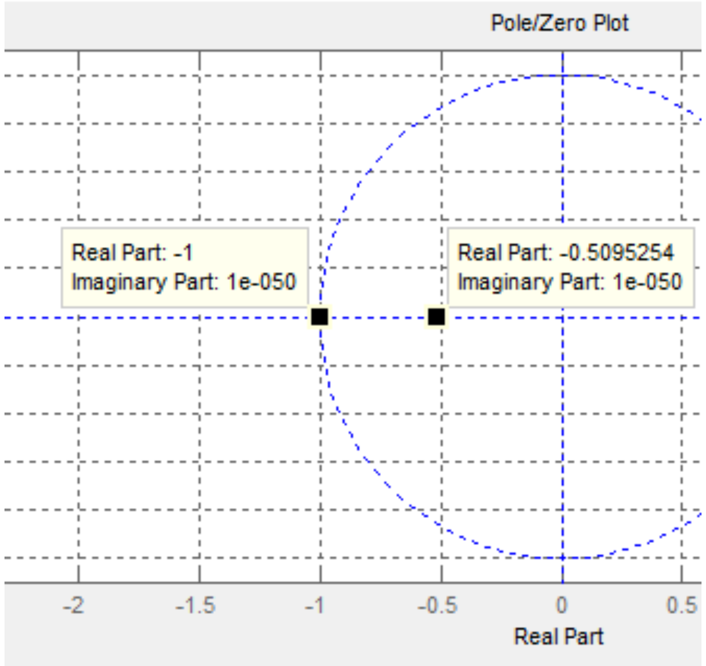
9/3/2013

```
cuttOffAF = 400;    % analog cutoff freq
Fs        = 1000;   % sampling freq
smallw    = 2*pi*cuttOffAF/Fs  % S1: first convert to digital frequency
ButterN   = 1;  % order of butterworth filter
```

```
Warped_AnalogW =

    3.0777
```

```
>> Bs2, As2

Bs2 =

         0    3.0777


As2 =

    1.0000    3.0777
```

```
Bz =

    0.7548    0.7548


Az =

    1.0000    0.5095
```




```
% to design from butterworth filter, using the pre-warped analog freq,
% get the butterworth filter Bs, As
[Bs2,As2] = butter(ButterN,Warped_AnalogW,'s');  % S3: design analog filter using butter, N=2, analog freq

% apply bilinear transformation on the found analog filter
[Bz,Az] = bilinear(Bs2,As2,1/2);  % S5: bilinear transform, using Td = 2 again!, Fs = 1/2
[Hz,Wz] = freqz(Bz,Az);
```


Matlab's solution: gain normalized to 1, POLE/Zero plot same as ours!

Methodology for IIR

1) What is the characteristics of an IIR Filter
   - Its impulse response, its difference equation, its H(z), and stability

2) Analysis: From difference equation to H(z), to pole/zero plot and then to frequency response H(ω)

3) IIR Filters are usually designed from analog filters, so lets examine H(s) the Laplace domain.

4) Design an analog filter
   a) Simple RC analog filter
   b) Butterworth analog filter
5) Design an IIR filter by converting from H(s) to H(z)
   a) Bilinear transform method
   b) Impulse invariance method.
6) Designing of IIR Filters using Matlab.

Impulse invariance transform to convert H(s) -> H(z)

- As the name implies, this method transform the analog filter H(s) ->H(z) while maintaining the shape of the impulse response.

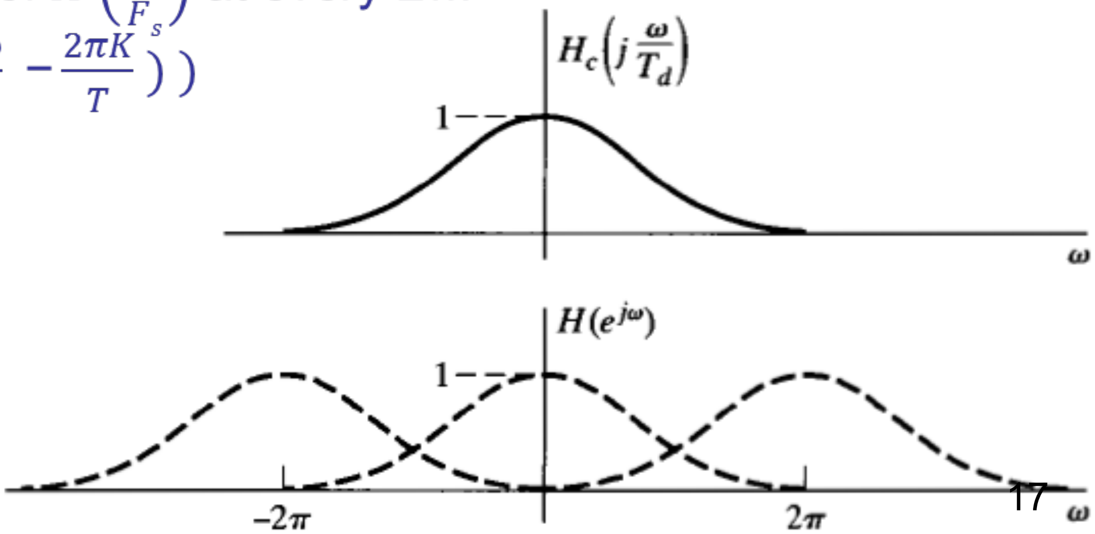
  See Oppenheim 3rd edition, Section 4.4.2 (pg 202)

- Given $h_c(t)$, its FT is $H(j\Omega)$,
- If $h_c(t)$ is band-limited, $|\Omega| < \pi/T$, then its sampled version h[n]'s Fourier Transform

$$\mathrm{H}\left(e^{j\omega}\right) = H\left(\frac{j\Omega}{F_s}\right) = H(j\omega Fs) \text{ for } |\omega| < \pi$$

- In addition, $\mathrm{H}\left(e^{j\omega}\right)$ has images of $H\left(\frac{j\Omega}{F_s}\right)$ at every 2π.

$$\mathrm{H}(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_c(j(\frac{\omega}{T} - \frac{2\pi K}{T}))$$

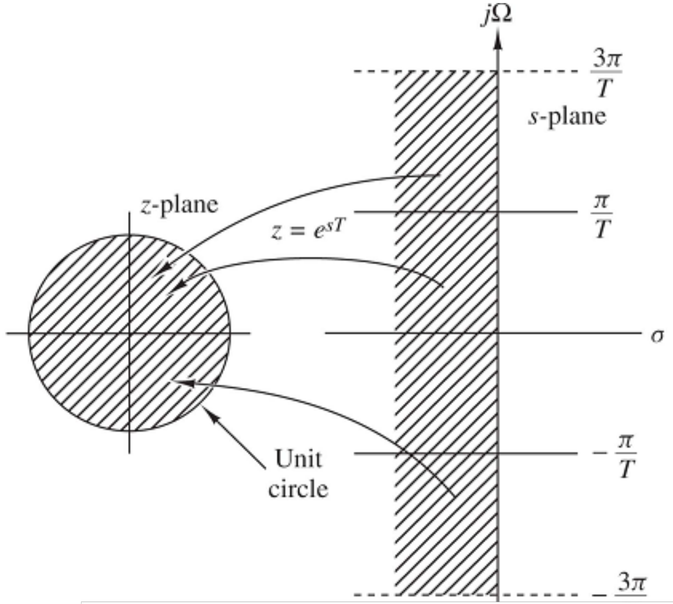


IIR filter design by Impulse Invariance

Question: What s-plane <-> z-plane mapping does this perform ?

Utilize $z = re^{j\omega}$ and $s = \sigma + j\Omega$ representations in the mapping

$$z = e^{sT}$$
$$re^{j\omega} = e^{\sigma T} e^{j\Omega T}$$
$$r = e^{\sigma T} \text{ and } \omega = \Omega T$$



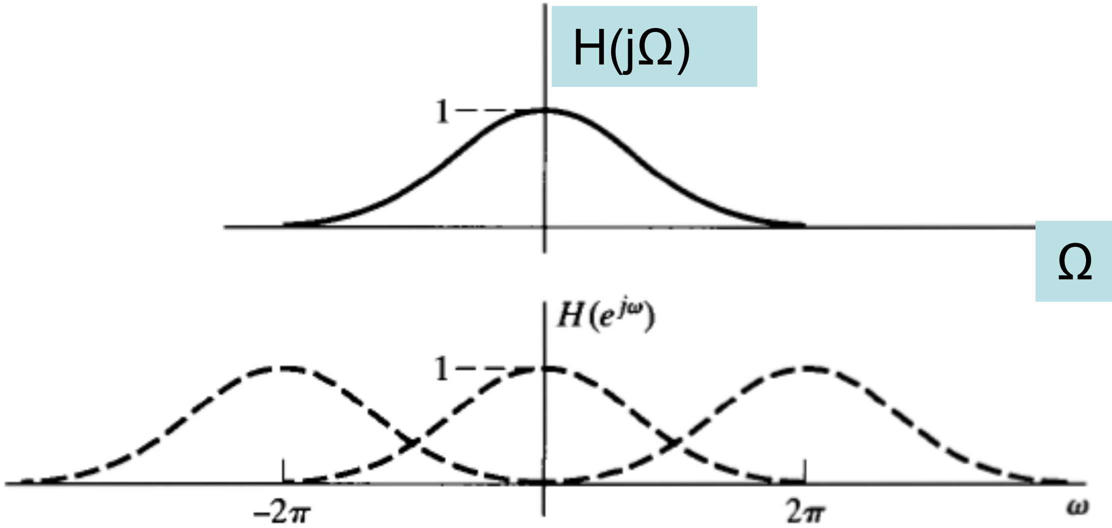
$z = e^{sT}$ induced s-plane to z-plane mapping

Observations:
- Mapping introduces aliasing (i.e., s and s + jk2π/T) -> same z
- Zeros and poles don't follow the same mapping

For σ negative, $r = e^{\sigma T}$ is always less than 0!,
And aliasing due to ΩT (sampling)

IIR filter design by Impulse Invariance

If the continuous time filter's frequency response H(jΩ) is not band-limited, then sampling the impulse response cause aliasing.



Hence impulse invariance design cannot be used for high pass and band-stop filter design as these filters are not bandlimited.

Note: frequency mapping is linear, no pre-warping is necessary.

Impulse Invariance Transformation Design Procedure Summary

0) Given a filter specification in the analog frequency domain
1) Convert the specification into digital frequency domain
1) Design the H_a (s) analog filter, e.g, using Butterworth filter design on the digital frequency specification.
2) Perform partial fraction expansion of $H_a(s)$

$$H_a(s) = \sum_{k=1}^{N} \frac{R_k}{s - p_k}$$

3) Convert H_a (s)->H(z) by impulse invariance transform method. Transform analog poles $\{p_k\}$ into digital poles $\{e^{p_k T}\}$ to obtain

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

Note: compare to bilinear transform method, NO frequency warping is necessary as frequency is linearly mapped!

Example: Compare analog versus digital filter response (impulse invariance) using Butterworth filter

```
>> [z,p,k] = butter(2,pi/5,'s')
```

```
z =

  Empty matrix: 0-by-1

p =

  -0.4443 + 0.4443i
  -0.4443 - 0.4443i

k =

    0.3948
```

By designing the analog filter cut-off using normalized frequency, The transformation is simplified as T_d = 1

- **Design a second order ideal low pass digital filter with cutoff at** $\omega_d = \pi / 5$ radians / sample

  (Assume $T = 1$**)**

  – **Analog cutoff frequency**

$$f_c = \frac{\omega_d}{2\pi T} = 0.1 \text{ Hz}$$

$$\omega_c = 2\pi f_c = \pi / 5 \text{ rad / sec}$$

- **Use Butterworth analog prototype**

$$H_a(s) = \frac{0.3948}{[s-(-0.4443+j0.4443)][s-(-0.4443-j0.4443)]}$$

$$H_a(s) = \frac{-j0.4443}{[s-(-0.4443+j0.4443)]} + \frac{j0.4443}{[s-(-0.4443-j0.4443)]}$$

Using impulse invariance mapping:

$$H_d(z) = \frac{-j0.4443}{1-0.6413e^{j0.4443}z^{-1}} + \frac{j0.4443}{1-0.6413e^{-j0.4443}z^{-1}}$$

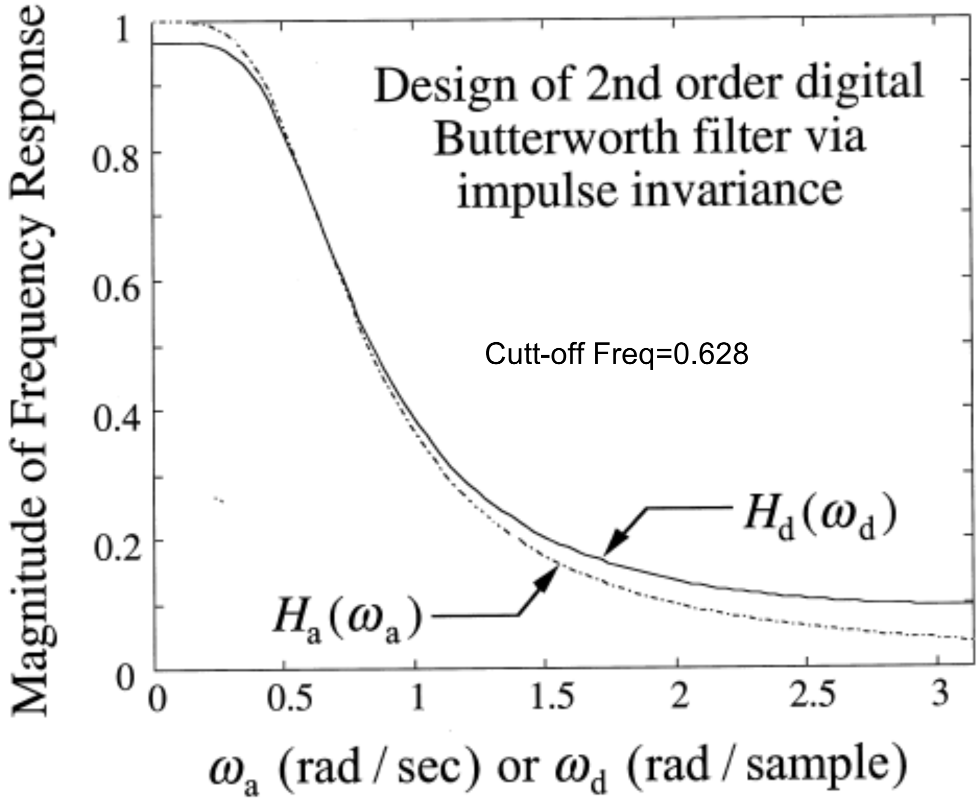
$$p_k = e^{s_k T_d}$$

```
>> abs(exp([-0.4443+j*0.4443]))

ans =

    0.6413
```

```
>> angle(exp([-0.4443+j*0.4443]))

ans =

    0.4443
```

Example: Compare analog versus digital filter response (impulse invariance) using Butterworth filter

```
>> [z,p,k] = butter(2,pi/5,'s')
```



Impulse invariance method: summary

- Preserve impulse response and frequency response if there is no aliasing
- Linear mapping of analog frequency to digital frequency (no pre-warping required)
- As mapping is due to sampling, aliasing may occur, and is only suitable for low-pass filter and band-pass filter design.
- Poles in analog domain maps to poles in digital domain
  $(poles\ in\ analog\ domain\ p_k) = (digital\ domain)e^{P_k T}$

Methodology for IIR

1) What is the characteristics of an IIR Filter
   - Its impulse response, its difference equation, its H(z), and stability

2) Analysis: From difference equation to H(z), to pole/zero plot and then to frequency response H(ω)

3) IIR Filters are usually designed from analog filters, so lets examine H(s) the Laplace domain.

4) Design an analog filter
   a) Simple RC analog filter
   b) Butterworth analog filter
5) Design an IIR filter by converting from H(s) to H(z)
   a) Bilinear transform method
   b) Impulse invariance method.
6) Designing of IIR Filters using Matlab.

Designing Filters using Frequency Transformation

- Use the digital filter specifications to determine a suitably normalized lowpass filter H(s)
- Wrap the critical frequencies of the digital filter using $\Omega_i = \tan(\frac{\omega_i}{2})$. For lowpass/highpass, there is one critical frequency say Ω_c. For bandpass/bandstop there are two, Ω_u and Ω_l ($\Omega_u > \Omega_l$)
- Replace s in the transfer function H(s) using one of the transformations presented in the table below .
- Apply the bilinear transform to the resulting analogue filter

Table: the transformations of a unit cutoff analogue frequency Ω_c (i.e. ω_c=1) to other filters

Lowpass to...	Use substitution	New cutoff(s)
Lowpass	$s \leftarrow s/\Omega_c$	Ω_c
Highpass	$s \leftarrow \Omega_c/s$	Ω_c
Bandpass	$s \leftarrow \left(s^2 + \Omega_l\Omega_u\right) / \left(s\left(\Omega_u - \Omega_l\right)\right)$	Ω_l, Ω_u
Bandstop	$s \leftarrow s\left(\Omega_u - \Omega_l\right) / \left(s^2 + \Omega_l\Omega_u\right)$	Ω_l, Ω_u

Example. Frequency tranformation

- Example 1: One is interested in designing a highpass digital filter with cutoff ω_c (between 0 and π) using a first order analogue filter given by

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

- In this case, it follows that the transformation to apply is s <- Ω_c/s where $\Omega_c = \tan(\omega_c/2)$. It yields

$$\mathrm{H}'(\mathrm{s}) = \frac{1}{\Omega_c\ /s+1} = \frac{s}{\Omega+s}$$

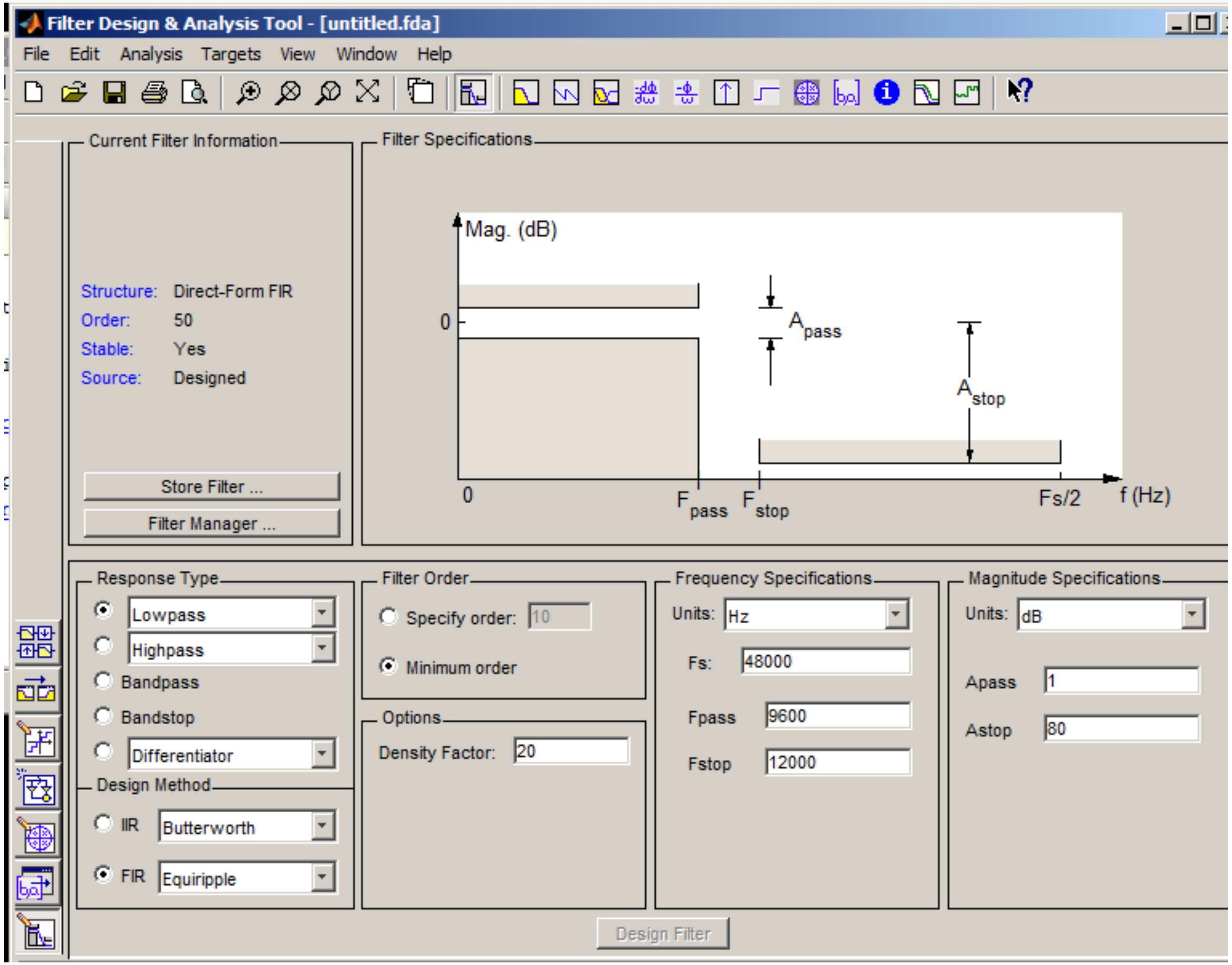
- Now using the bilinear transform one obtains

$$\mathrm{H(z)} = H'(s)|_{s=\frac{1-z^{-1}}{1+z^{-1}}} = \frac{1}{\Omega_c\frac{(1+z^{-1})}{(1-z^{-1})}+1}$$

$$= \frac{(1-z^{-1})}{\Omega_c(1+z^{-1})+(1-z^{-1})} = \frac{(1-z^{-1})}{(\Omega_c+1)+(\Omega_c-1)z^{-1}}$$

Designing FIR/IIR Filters using matlab

- fdatool - filter design and analysis tool (Signal processing toolbox)
- Allows you to design FIR and IIR filter.
- Specify order, specification of tolerance, design method, etc,
- Allow you to study quantization effects



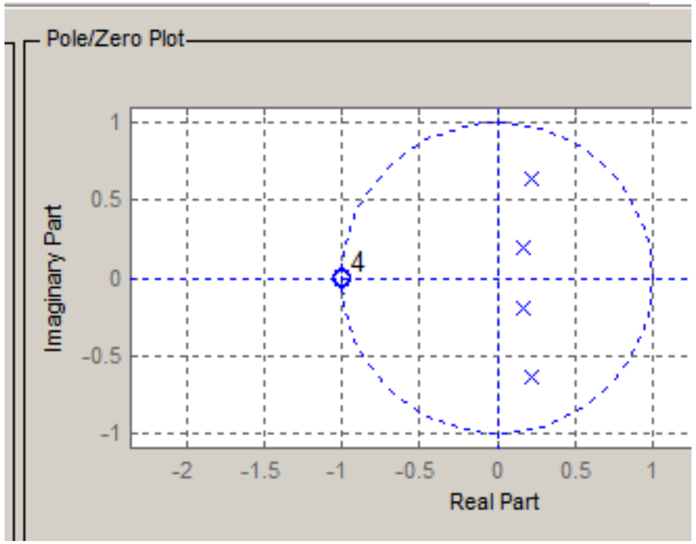
Designing FIR/IIR Filters using matlab

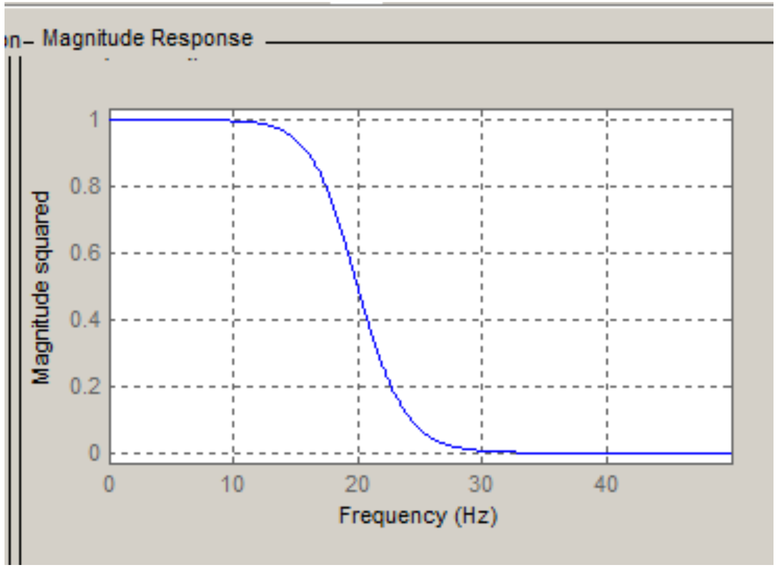
- We wish to see:
  - [B,A] coefficients
  - Frequency response
  - Pole Zero

Structure:	Direct-Form I
Order:	4
Stable:	Yes
Source:	Converted (converted)

4th order IIR low pass filter

```
Numerator:
0.046582906636443676
0.1863316265457747
0.27949743981866204
0.1863316265457747
0.046582906636443676
Denominator:
 1
-0.7820951980233793
 0.6799785269162999
-0.1826756977530325
 0.03011887504316928
```





SUMMARY

- What have we learnt?
- 1) Designing an IIR Digital filter, we begin with a low pass analog design and transform it to digital filter by either bilinear transformation or impulse invariance transformation.
- 2) Bilinear Transform is more commonly used!
- 3) If we want a high-pass, band-stop, band-pass, we can apply frequency transformation technique on the low pass design for H(s) and then
  H(s) ->H(z) to get the required digital filter.
- Using Fdatool to design filters in Matlab

If you are the DSP engineer

- You will design filters using fdatool and export the coefficients into your design directly.
- You will simulate fix point arithmetic,
- And you will use second order structure in cascade, or lattice structure for numerical robustness.
- You will not simply just get direct form [B,A] from fdatools and implement them.

So why did we learn all this theory!

The understanding of all this theory will allow you to make an inform choice of the tradeoffs.

If you are going into research of DSP, or wish to work for Matlab!, then you will be the one who write these routines for the rest of the world to use anyway.