BINGHAMTON UNIVERSITY

State University of New York

EEO 401
Digital Signal Processing

Prof. Mark Fowler

Note Set #32

- IIR Design – Introduction
- Reading: Sect. 10.3 up through 10.3.2 of Proakis & Manolakis

Motivation

When DSP first was being developed the theory of analog filters was already extremely well developed.

Since analog filters have infinite duration impulse responses and rational transfer functions it was natural for DSP researchers to seek ways to convert existing analog (IIR) designs into DT IIR designs.

Since there are three ways to describe a system (CT & DT) there are three paths we can take to convert an analog filter into DT:

- Differential Equation convert to Difference Equation
- Impulse Response $h(t)$ convert to Impulse Response $h[n]$
- Transfer Function $H(s)$ convert to Transfer Function $H(z)$

These two don't pan out so well… (first two paths)

We'll focus here! (transfer function path)

Goal: given a CT filter, convert it into a DT filter that:

- Retains the essential passband & stopband characteristics
- Remains stable when the CT filter was stable
- Is physically realizable (i.e., resulting $H(z)$ is a rational function)

Focus is on achieving magnitude response characteristics

– linear phase is not achievable with DT IIR filters!

Notation

Notation is crucial here… we will be slinging around CT & DT transfer function and frequency response… and because we are using the CT descriptions to arrive at the DT descriptions they will get linked together!!!

- Here I'll give in to P&M's use of Ω for CT & ω for DT
- But… I'll also insist on using Porat's superscript notation to keep these functions straight (since they all use "*H*")
- I'll also keep P&M's use of subscript "*a*" to indicate "analog"

$$H_a^L(s) = \frac{\sum_{k=0}^{M} \beta_k s^k}{\sum_{k=0}^{M} \alpha_k s^k} \quad \Rightarrow \quad H_a^F(\Omega) = \frac{\sum_{k=0}^{M} \beta_k (j\Omega)^k}{\sum_{k=0}^{N} \alpha_k (j\Omega)^k}$$

rad/sec

$$H^z(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k s^k} \quad \Rightarrow \quad H^f(\omega) = \frac{\sum_{k=0}^{M} b_k e^{-j\omega k}}{\sum_{k=0}^{N} a_k e^{-j\omega k}}$$

rad/sample

IIR Design – Diff Eq Conversion

This was probably the first method used to develop DT IIR filters because difference equations were used to numerically solve differential equations.

The basic idea is based on the fact a derivative can be approximated by a difference:

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} \approx \frac{y[n]-y[n-1]}{T}$$

$$\left.\frac{d^2y(t)}{dt^2}\right|_{t=nT} = \left[\frac{d}{dt}\left[\frac{dy(t)}{dt}\right]\right]_{t=nT}$$

$$\approx \frac{\left(y[n]-y[n-1]\right)/T - \left(y[n-1]-y[n-2]\right)/T}{T}$$

$$= \frac{y[n]-2y[n-1]+y[n-2]}{T^2}$$

Etc… These can then be substituted into the Differ<u>ential</u> Equation and re-arranged into standard Differ<u>ence</u> Equation form. Or….

Since each s^k in $H^L(s)$ indicates a k^{th} order derivative we can use these approximate derivative results to find how to replace s^k with an appropriate function of z:

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} \approx \frac{y[n]-y[n-1]}{T} \quad \Rightarrow \quad s = \frac{1-z^{-1}}{T}$$

$$\left.\frac{d^2y(t)}{dt^2}\right|_{t=nT} \approx \frac{y[n]-2y[n-1]+y[n-2]}{T^2} \quad \Rightarrow \quad s^2 = \frac{1-2z^{-1}+z^{-2}}{T^2} = \left(\frac{1-z^{-1}}{T}\right)^2$$

Generalizing gives: $$s^k = \left(\frac{1-z^{-1}}{T}\right)^k$$

The DT design from the CT TF is $$H^z(z) = \left.H_a^L(s)\right|_{s=\left(1-z^{-1}\right)/T}$$

Inverting this mapping gives $$z = \frac{1}{1 - sT}$$

This shows how the *s*-plane gets mapped to the *z*-plane:

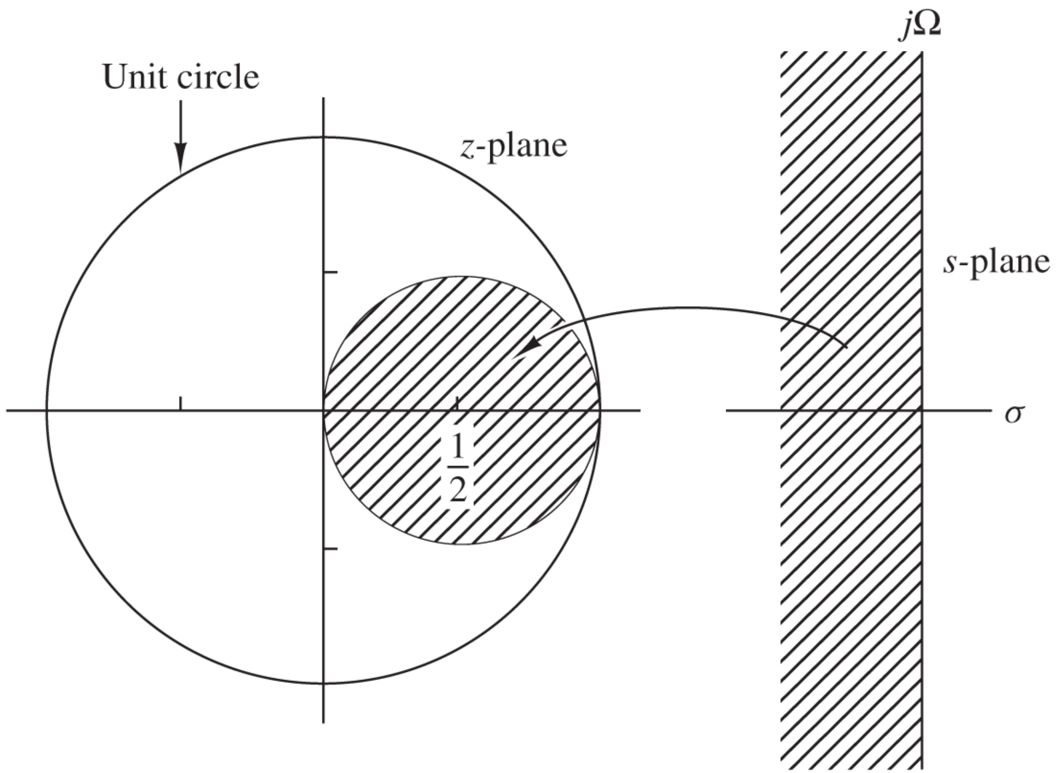


Figure 10.3.2 The mapping $s = (1 - z^{-1})/T$ takes LHP in the s-plane into points inside the circle of radius $\frac{1}{2}$ and center $z = \frac{1}{2}$ in the z-plane.

- Stability is preserved…
  - LHP maps inside UC
- But… can only map CT poles and zeros into a limited region of *z*-plane!!
  - Limits to LP filters!

IIR Design – Impulse Invariance

Idea: given a desired CT impulse response $h_a(t)$ design a DT IIR filter whose $h[n]$ matches the samples of $h_a(t)$

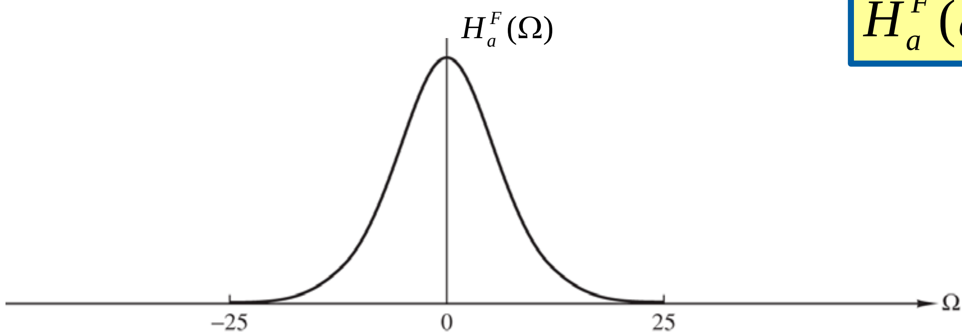
Using our understanding from sampling we know that the DTFT of this $h[n]$ = $h_a(nT)$ will contain spectral replicas of $H_a(\omega)$:

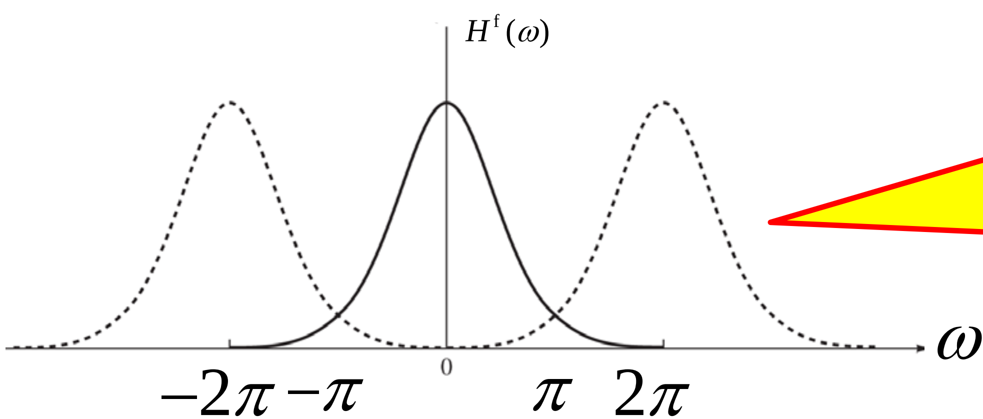
$$H^{\mathrm{f}}(\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a^F\left((\omega - 2\pi k)F_s\right)$$

$H_a^F(\Omega)$ rad/sec

$\downarrow$

$H_a^F(\omega F_s)$ (rad/smpl)(smpl/sec) = rad/sec





- Aliasing causes an error in desired Freq Resp!!!
- Limited to implementing LP filters (and maybe BP)

So… this method is limited but is useful when the focus is on making a filter that has a desired impulse response.

<u>Issue</u>: How do we get a DT **<u>rational</u>** transfer function that achieves the impulse invariance criteria??

Can generalize… but won't!

Can answer this for the case of the CT filter having distinct poles:

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

After doing PFE

Then…

$$h_a(t) = \sum_{k=1}^{N} c_k e^{p_k t} u(t)$$

And…

$$h[n] = h_a(nT) = \sum_{k=1}^{N} c_k e^{p_k nT} u[n] = \sum_{k=1}^{N} c_k \underbrace{\left(e^{p_k T}\right)^n u[n]}$$

On ZT Table!

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

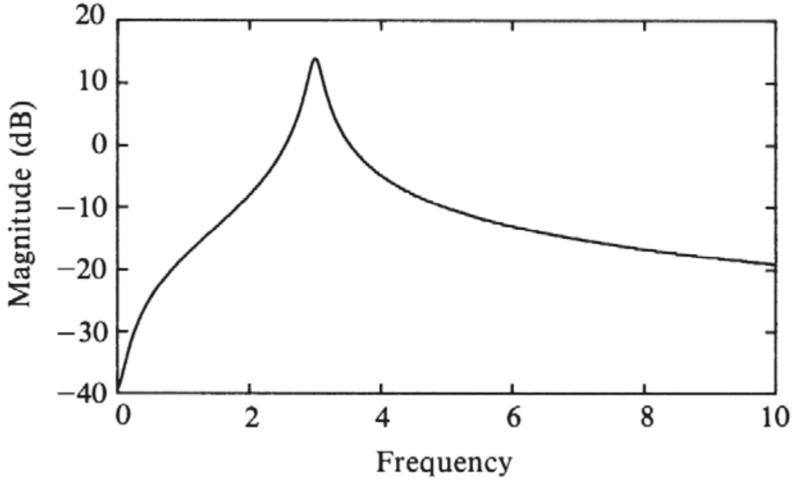
Can convert into Rational function

Example 10.3.3 $H_a^L(s) = \dfrac{1/2}{s+(0.1-j3)} + \dfrac{1/2}{s+(0.1+j3)}$

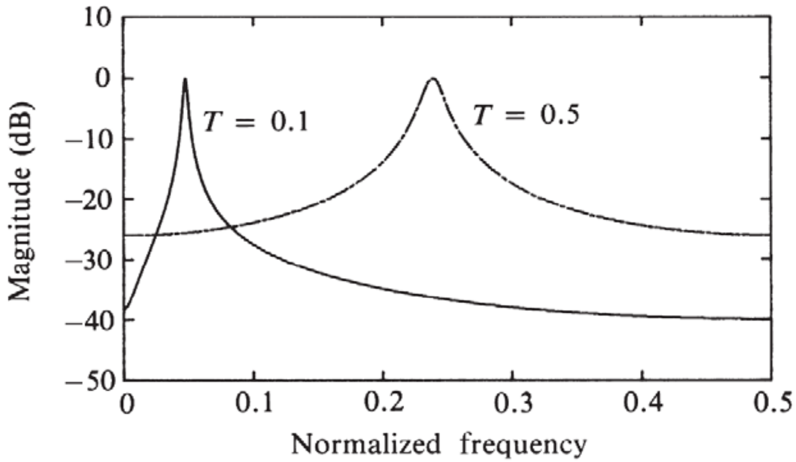
Applying the result to this CT transfer function gives

$$H^z(z) = \frac{1/2}{1-e^{-0.1T}e^{j3T}z^{-1}} + \frac{1/2}{1-e^{-0.1T}e^{-j3T}z^{-1}} = \frac{1-[e^{-0.1T}\cos(3T)]z^{-1}}{1-[2e^{-0.1T}\cos(3T)]z^{-1}+e^{-0.2T}z^{-1}}$$

$\left|H_a^F(\Omega)\right|$



$\left|H^f(\omega)\right|$



MATLAB has a command for this method

From the MATLAB website:

impinvar

R2015a

Impulse invariance method for analog-to-digital filter conversion

collapse all in page

Syntax

```
[bz,az] = impinvar(b,a,fs)
[bz,az] = impinvar(b,a,fs,tol)
```

Description

`[bz,az] = impinvar(b,a,fs)` creates a digital filter with numerator and denominator coefficients `bz` and `az`, respectively, whose impulse response is equal to the impulse response of the analog filter with coefficients `b` and `a`, scaled by `1/fs`. If you leave out the argument `fs`, or specify `fs` as the empty vector `[]`, it takes the default value of 1 Hz.

`[bz,az] = impinvar(b,a,fs,tol)` uses the tolerance specified by `tol` to determine whether poles are repeated. A larger tolerance increases the likelihood that `impinvar` interprets closely located poles as multiplicities (repeated ones). The default is 0.001, or 0.1% of a pole's magnitude. Note that the accuracy of the pole values is still limited to the accuracy obtainable by the `roots` function.