Title/Number:

DSP ECE 623

By

Andreas Spanias

Chapter 5 – IIR Filter Design

spanias@asu.edu

Phone: 480 965 1837, Fax: 480 965 8325
http://www.eas.asu.edu/~spanias

Design of IIR Digital Filters

Ch. 5 - Lecture 18

IIR DIGITAL FILTERS

Advantages:
Efficient in terms of order
- Poles create narrow-band peaks efficiently
- Arbitrarily long impulse responses with few feedback
- coefficients

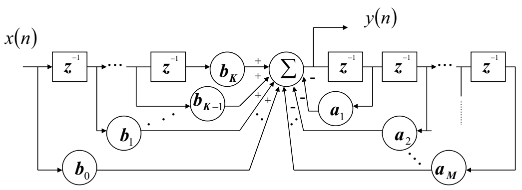
Disadvantages:
- Feedback and stability concerns
- Sensitive to Finite Word Length Effects
- Generally non-Linear Phase

Applications:
- Speech Processing, Telecommunications
- Data Processing, Noise Suppression, Radar

IIR FILTERS

The difference equation is:

$$y(n) = \sum_{i=0}^{L} b_i x(n-i) - \sum_{i=1}^{M} a_i y(n-i)$$



IIR FILTERS (Cont.)

The transfer function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_L z^{-L}}{1 + a_1 z^{-1} + \ldots + a_M z^{-M}}$$

The frequency-response function :

$$H(e^{j\Omega}) = \frac{b_0 + b_1 e^{-j\Omega} + \ldots + b_L e^{-jL\Omega}}{1 + a_1 e^{-j\Omega} + \ldots + a_M e^{-jM\Omega}}$$

IIR Filter Design by Analog Filter Approximation

The idea is to use many of the successful analog filter designs to design digital filters

This can be done by either:

- by sampling the analog impulse response (*impulse invariance*) and then determining a digital transfer function

or

- by transforming directly the analog transfer function to a digital filter transfer function using the *bilinear transformation*

IIR Filter Design by Analog Filter Approximation

The *impulse invariance* method suffers from aliasing and is not used often

The *bilinear transformation* does not suffer from aliasing and is by far more popular than the impulse invariance method. The frequency relationship from the s-plane to the z-plane is non-linear, and one needs to compensate by pre-processing the critical frequencies such that after the transformation the desired response is realized. Stability is maintained in this transformation since the left-half s-plane maps onto the interior of the unit circle.

IIR Filter Design by Impulse Invariance R C LPF Example

$$h_a(t) = \frac{1}{RC} e^{-\frac{t}{RC}} u(t) \quad \text{and} \quad H_a(j\omega) = \frac{1}{1 + j\omega RC}.$$

A discrete-time approximation of the impulse response can be written as

$$h(n) = T\ h_a(t)|_{t=nT}\ . \qquad h(n) = \frac{T}{RC}\ e^{-\frac{nT}{RC}}\ u(n).$$

$$H(z) = \frac{(T/RC)}{1 - e^{-(T/RC)} z^{-1}}.$$

IIR Filter Design by Impulse Invariance General

$$H_a(s) = \frac{1}{s - p_1} + \frac{1}{s - p_2} + ... + \frac{1}{s - p_M}$$

With impulse invariance it maps to

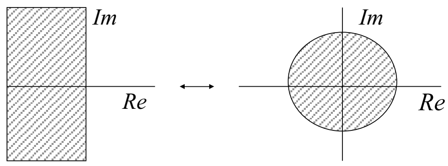
$$H(z) = \frac{T}{1 - e^{p_1 T} z^{-1}} + \frac{T}{1 - e^{p_2 T} z^{-1}} + ... + \frac{T}{1 - e^{p_M T} z^{-1}}.$$

Clearly, if the s-domain poles are on the left-half of the *s* plane, i.e. Re[*pi*]<0, the *z*-domain poles will be inside the unit circle, that is |e*pi*T |<1. The disadvantage of the impulse invariance method is the unavoidable frequency-domain aliasing.

IIR Design with the Bilinear Transformation

Bilinear Transform $\Rightarrow \; z = \dfrac{1+s}{1-s}$

$S - plane$ $\qquad\qquad$ $z - \textbf{plane}$



$$\boldsymbol{H}(z) = \boldsymbol{H}\left(\boldsymbol{s} = \frac{z-1}{z+1}\right)$$

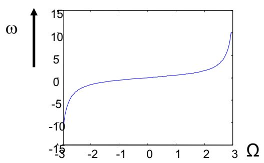
The Bilinear Transformation (Cont.)

The bilinear transformation compresses the frequency axis

$$\omega[-\infty,\infty] \leftrightarrow \Omega[-\pi,\pi]$$

The non-linear frequency transformation (frequency warping function) is given by

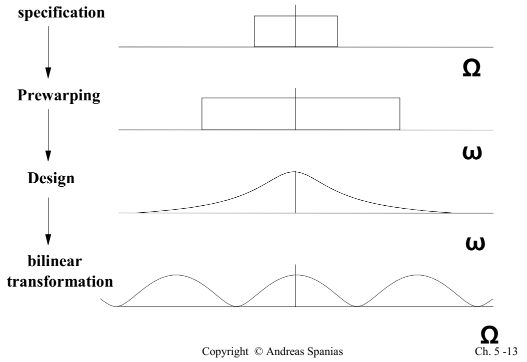
$$\omega = \tan\left(\frac{\Omega}{2}\right)$$



Procedure for Analog Filter Approximation

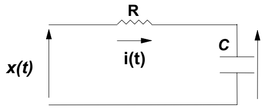
1. **Consider Critical Frequencies**
2. **Pre-warp critical frequencies**
3. **Analog Filter Design**
4. **Bilinear Transformation**
5. **Realization**

Applying the Bilinear Transformation



EXAMPLE: TRANSFORMING AN RC CIRCUIT TO A DF

Suppose we want a first-order
(R-C LPF) appoximation



$$H(\omega) = \frac{1}{1 + j\omega RC}$$

Say we have the following DF specs:

Step 1: $\Omega_c = \pi/2$

Apply pre-warping

Step 2: $\omega_c = \tan(\frac{\Omega_c}{2}) = \tan(\frac{\pi}{4}) = 1$

Step 3: Design the analog filter.
In this case the analog filter function is a first order LPF similar → to an RC circuit

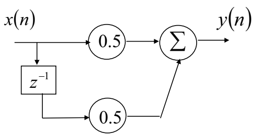
$$H(s) = \frac{1}{1+s}$$

TRANSFORMING AN RC CIRCUIT TO A DF (2)

Step 4: Apply the Bilinear Transform

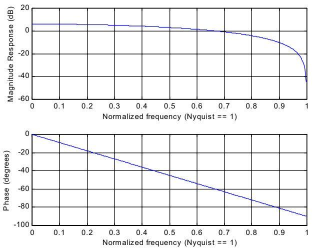
$$H(z) = H(s = \frac{z-1}{z+1}) = \frac{1}{1 + \frac{z-1}{z+1}} = 0.5 + 0.5z^{-1}$$

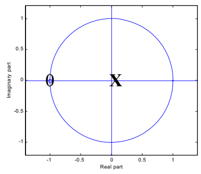
Step 5: Realization



TRANSFORMING AN RC CIRCUIT TO A DF (3)

Frequency Response $H(e^{j\Omega}) = 0.5 + 0.5e^{-j\Omega}$





Notice that there is no aliasing effect with the bilinear transformation. Although in this simple R-C example the resultant digital filter is FIR, more complex analog filters will yield IIR digital filters.

Old title/number:
MATLAB for DSP EEE 598

Title/Number:

DSP Algorithms and Software EEE 509

by
Andreas Spanias, Ph.D.

Chapter 5 – IIR Filter Design

spanias@asu.edu

Phone: 480 965 1837, Fax: 480 965 8325
http://www.eas.asu.edu/~spanias

Design of IIR Digital Filters

Ch. 5 - Lecture 19

Analog Filter Designs

- Butterworth - Maximally flat in passband
- Chebyshev I - Equiripple in passband
- Chebyshev II - Equiripple in stopband
- Elliptic - Equiripple in passband and stopband

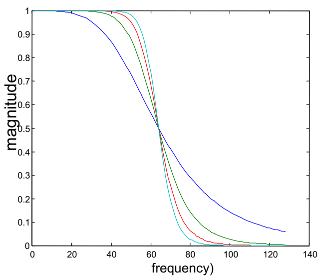
Butterworth Filter Design

- Maximally Flat in the Passband and Stopband

$$|H(\omega)|^2 = \frac{1}{1+\left(\frac{\omega}{\omega_C}\right)^{2M}}$$

Butterworth Max. Flat in Passband and Stopband

Butterworth frequency response - transition is steeper as order increases



Butterworth Transfer Function

$$H(s)H(-s) = \frac{1}{1+(\frac{s}{j\omega_C})^{2N}}$$

Poles on a circle of radius ω_c

$$(\frac{s}{j\omega_C})^{2N} = -1$$

$$s_k = (-1)^{1/2N}\, j\omega_C = \omega_C\, e^{\frac{j\pi(2k+N-1)}{2N}}$$

$$k = 0,1,\ldots,2N-1$$

note that roots can not fall on imaginary axis

Design Example- Butterworth

- A Butterworth filter is designed by finding the poles of H(s)H(-s)
- The poles fall on a circle with radius ω_c
- The poles falling on the left hand s-plane (stable poles) are chosen to form H(s)
- H(s) is transformed to H(z)

Butterworth Example

Determine the order and poles of a digital Butterworth filter .Sampling =8 kHz, passband edge=1 kHz, stopband edge=1.6 kHz, gain at passband edge=-1 dB, and gain at stopband edge=-40 dB. We use the bilinear transformation and we take the following steps:

- - **Step 1:** Determine the normalized passband edge and stopband edge frequencies;

$$\Omega_p = 2\pi(1000/8000) = 0.25\pi;\ \ \Omega_{st} = 2\pi(1600/8000) = 0.4\pi$$

- - **Step 2:** Pre-warp the critical frequencies .

$$\omega_p=\tan(\Omega_p/2)=\tan(0.125\pi);\ \ \omega_{st}=\tan(\Omega_{st}/2)=\tan(0.2\pi)$$

Butterworth Example (2)

- \- **Step 3:** Determine the order of the Butterworth filter

$$|H(\omega)|^2 = \frac{1}{1+(\frac{\omega}{\omega_C})^{2M}}$$

$$\frac{1}{g_p}-1=\left(\frac{\omega_p}{\omega_c}\right)^{2M} \quad and \quad \frac{1}{g_{st}}-1=\left(\frac{\omega_{st}}{\omega_c}\right)^{2M}$$

$$G=\frac{\frac{1}{g_p}-1}{\frac{1}{g_{st}}-1}=\left(\frac{\omega_p}{\omega_{st}}\right)^{2M} \quad hence\ M \geq \frac{\log G}{2\log\left(\frac{\omega_p}{\omega_{st}}\right)}$$

Butterworth Example (3)

$$\frac{1}{0.794}-1=\left(\frac{0.414}{\omega_c}\right)^{2M} \quad and \quad \frac{1}{0.0001}-1=\left(\frac{0.7265}{\omega_c}\right)^{2M}$$

$$G=\frac{\frac{1}{0.794}-1}{\frac{1}{0.0001}-1}=\left(\frac{0.414}{0.7265}\right)^{2M} \Rightarrow G=2.59\times10^{-5}$$

$$M \geq \frac{\log(2.59\times10^{-5})}{2\log\left(\frac{0.414}{0.7265}\right)}, hence\ M \geq 9.3898 = 10.$$

Butterworth Example (4)

- **Step 4**; Find the cutoff frequency. If one evaluates ωc at the passband edge, then the specification is met exactly in the passband and exceeded in the stopband. If ωc is evaluated at the stopband edge frequency, then the specification is met exactly in the stopband and exceeded in the passband.

For this example, we choose the stopband equation: . Now, substituting with M=10, we get **ωc = 0.458 rad.**

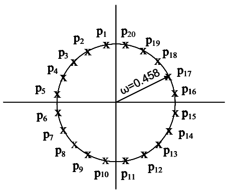
$$\frac{1}{0.0001}-1=\left(\frac{0.7265}{\omega_c}\right)^{20}.$$

Butterworth Example (5)

- **Step 5** The analog poles lie on a circle of radius ωc . There are a total of *2M = 20* poles on this s-domain circle which implies that the angular separation between any two poles is $2\pi/20$

$$p_k = (-1)^{1/2N}\, j\omega_C = \omega_C\, e^{\frac{j\pi(2k+N-1)}{2N}}$$

S-domain



Butterworth Example (6)

- **Apply the Bilinear transform and obtain the z-domain poles.**

k	*s*-domain poles	*z*-domain poles
1	-0.0716 + j 0.4524	0.5840 + j 0.6687
2	-0.2079 + j 0.4081	0.4861 + j 0.5021
3	-0.3239 + j 0.3239	0.4254 + j 0.3487
4	-0.4081 + j 0.2079	0.3901 + j 0.2053
5	-0.4524 + j 0.0716	0.3737 + j 0.0678
6	0.4524 - j 0.0716	2.5906 - j 0.4698
7	0.4081 - j 0.2079	2.0077 - j 1.0565
8	0.3239 - j 0.3239	1.4060 - j 1.1524
9	0.2079 - j 0.4081	0.9954 - j 1.0280
10	0.0716 - j 0.4524	0.7410 - j 0.8483
11	-0.0716 - j 0.4524	0.5840 - j 0.6687
12	-0.2079 - j 0.4081	0.4861 - j 0.5021
13	-0.3239 - j 0.3239	0.4254 - j 0.3487
14	-0.4081 - j 0.2079	0.3901 - j 0.2053
15	-0.4524 - j 0.0716	0.3737 - j 0.0678
16	0.4524 + j 0.0716	2.5906 + j 0.4698
17	0.2079 + j 0.4081	0.9954 + j 1.0280
18	0.3239 + j 0.3239	1.4060 + j 1.1524
19	0.4081 + j 0.2079	2.0077 + j 1.0565
20	0.0716 + j 0.4524	0.7410 + j 0.8483

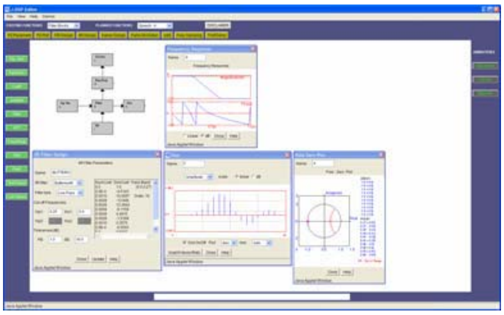
Again note the conjugate symmetry of the poles.

Butterworth Example (7)

- **Step 6** Choose the stable poles (left hand s plane) and form H(z)

$$H(z) = \left(\frac{0.0002z^{-1} + 0.0010z^{-2} + 0.0028z^{-3} + 0.0049z^{-4} + 0.0058z^{-5} + 0.0049z^{-6} + 0.0028z^{-7} + 0.0010z^{-8} + 0.0002z^{-9}}{1 - 4.5143z^{-1} + 10.0097z^{-2} - 13.948z^{-3} + 13.3604z^{-4} - 9.1159z^{-5} + 4.4615z^{-6} - 1.5399z^{-7} + 0.3575z^{-8} - 0.0503z^{-9} + 0.0032z^{-10}} \right.$$

Butterworth Example (8)



Examples of IIR Filter Design Using MATLAB

FUNCTIONS IN THE SP TOOLBOX

IIR digital filter design.
butter - Butterworth filter design.
cheby1 - Chebyshev type I filter design.
cheby2 - Chebyshev type II filter design.
ellip - Elliptic filter design.
maxflat - Generalized Butterworth lowpass filter design.
yulewalk - Yule-Walker filter design.

IIR filter order selection.
buttord - Butterworth filter order selection.
cheb1ord - Chebyshev type I filter order selection.
cheb2ord - Chebyshev type II filter order selection.
ellipord - Elliptic filter order selection.

Butterworth Design in MATLAB

```
• % Design an IIR Butterworth filter
• clear
• N=256; %for the computation of N discrete frequencies
• Wp=0.4; %passband edge
• Ws=0.6; %stopband edge
• Rp=1; % max dB deviation in passband
• Rs=40; %min dB rejection in stopband
• [M,Wn]=buttord(Wp,Ws,Rp,Rs);
• [b,a]=butter(M,Wn);

• theta=[(2*pi/N).*[0:(N/2)-2]]; % precompute the set of discrete frequencies up
  to fs/2
• H=freqz(b,a,theta); % compute the frequency response
• plot(angle(H))
• pause
• H=(20*log10(abs(H))); % plot the magnitude of the frequency response
• plot(H)
• title('frequency response')
• xlabel('discrete frequency index (N is the sampling freq.)')
• ylabel('magnitude (dB)')
```

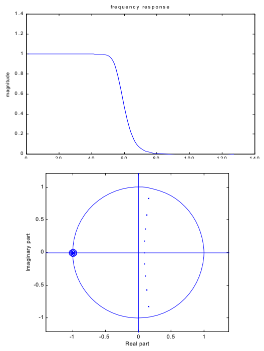

Butterworth Design in MATLAB (2)

- Wp=0.4; %passband edge
- Ws=0.6; %stopband edge
- Rp=1; % max dB deviation in passband
- Rs=40; %min dB rejection in stopband

- b = 0.0021 0.0186 0.0745 0.1739 0.2609 0.2609 0.1739
- 0.0745 0.0186 0.0021

- a = 1.0000 -1.0893 1.6925 -1.0804 0.7329 -0.2722 0.0916
- -0.0174 0.0024 -0.0001

Butterworth Design in MATLAB (3)

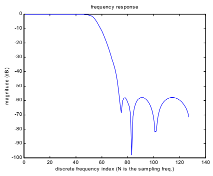


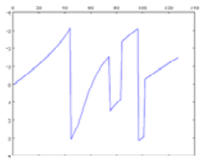
MATLAB Chebyshev II Design Example

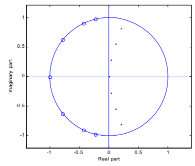
Chebyshev I - Equiripple in passband
Chebyshev II - Equiripple in stopband

```
% Design an IIR Chebyshev II filter - Ex 2.20

clear
N=256; %for the computation of N discrete frequencies

Wp=0.4; passband edge
Ws=0.5; stopband edge
Rp=1; ripple in passband (dB)
Rs=60; rejection (dB)

[M,Wn] = cheb2ord(Wp,Ws, Rp, Rs); % determine order

[b,a] = cheby2(M,58,Wn); %determine coefficients
size(a)
size(b)

% use routines to plot frequency response
```

IIR Chebyshev II Example







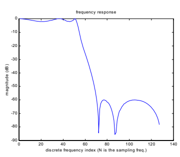
b = 0.0274 0.1065 0.2290 0.3252 0.3252 0.2290 0.1065 .0274

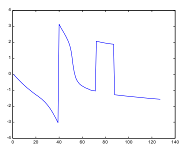
a =1.0000 -0.7484 1.2644 -0.4555 0.3427 -0.0454 0.0186 -0.0002

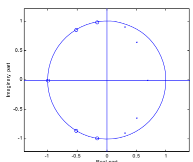
MATLAB Elliptic Design Example

```
% Design an IIR Elliptic filter
clear
N=256; %for the computation of N discrete frequencies

Wp=0.4; %passband edge
Ws=0.6; %stopband edge
Rp=2; % max dB deviation in passband
Rs=60; %min dB rejection in stopband
[M,Wn] = ellipord(Wp,Ws,Rp,Rs);
[b,a] = ellip(M,Rp,Rs,Wn); %design filter
size(a)
size(b)

theta=[(2*pi/N).*[0:(N/2)-2]]; % precompute the set of discrete frequencies up to fs/2
H=freqz(b,a,theta); % compute the frequency response
plot(angle(H))
pause
H=(20*log10(abs(H))); % plot the magnitude of the frequency response
plot(H)
title('frequency response')
xlabel('discrete frequency index (N is the sampling freq.)')
ylabel('magnitude (dB)')
pause
zplane(b,a) ;% z plane plot
```


IIR Elliptic







b = 0.0181 0.0431 0.0675 0.0675 0.0431 0.0181

a = 1.0000 -2.3214 3.3196 -2.8409 1.5154 -0.4151

Frequency Transformations

- Can be used to transform prototype LPF to: HPF, BPF, BSF
- Frequency transformations involve

$$H(z) = H_{LP}(\hat{z})\big|_{\hat{z}^{-1} = T(z^{-1})}$$

$$T(z^{-1}) = \pm \prod_{k=1}^{K} \frac{z^{-1} - \delta_k}{1 - \delta_k z^{-1}}$$

$$\delta_k \; - \; defined \quad in \quad tables$$

Group Delay Equalization

- Large variations in group delay can be equalized by cascading with an all-pass section

$$H_{eq}(z) = H(z) H_{AP}(z)$$

$$\tau_{eq} = \tau + \tau_{AP}$$

$$H_{AP}(z) = \frac{z^{-1} - a}{1 - az^{-1}}$$

Notes on IIR Filter Realizations

Direct Realization: This is a direct realization of the difference equation. This realization is susceptible to coefficient quantization .

Cascade Realization: This realization involves cascading first or second order sections.

$$H(z) = H_1(z) H_2(z) \ldots H_q(z)$$

Notes on IIR Filter Realizations (Cont.)

Parallel Realization: This basically uses a parallel arrangement of first and second order sections and relies on the parital fraction expansion of *H(z)*.

$$H(z) = H_1(z) + H_2(z) + \ldots + H_q(z) + B_0$$

Lattice Realization: This structure has two desirable properties that is: stability test by inspection, and reduced parameter sensitivity. Ladder realizations can be obtained by utilizing continuous partial fraction expansions starting with the system function. More on lattice realizations can be seen later with Linear Prediction.

FINITE WORD LENGTH EFFECTS

Finite word length results in significant losses in digital filter implementation. Errors are introduced by:
1. A/D conversion
2. FIR coefficient quantization
3. IIR coefficient quantization
4. Finite precision arithmetic

Quantization errors in IIR filters are more serious and sometimes catastrophic since they propagate through feedback. A stable IIR filter with poles inside but close to the unit circle, may become unstable when its coefficients are quantized.

FIR vs IIR Digital Filters

FIR	IIR
Always stable	Not always stable
Transversal	Recursive
All-zero model	All-pole or Pole-zero model
Moving Average(MA) model	Autoregressive(AR) or Autoregressive Moving Average (ARMA) model
Inefficient for spectral peaks	Efficient for spectral peaks (all-pole, pole-zero)
Efficient for spectral notches	All pole inefficient for spectral notches

FIR vs IIR Digital Filters (Cont.)

FIR	IIR
Requires high order design	Pole-zero efficient for both notches and peaks
	Generally requires lower order design
Less sensitive to finite word length implementation	More sensitive to finite word length implementation
Linear phase design	Generally non-linear phase