DSP

Chapter-3 : Filter Design

Marc Moonen
Dept. E.E./ESAT-STADIUS, KU Leuven
marc.moonen@esat.kuleuven.be
www.esat.kuleuven.be/stadius/

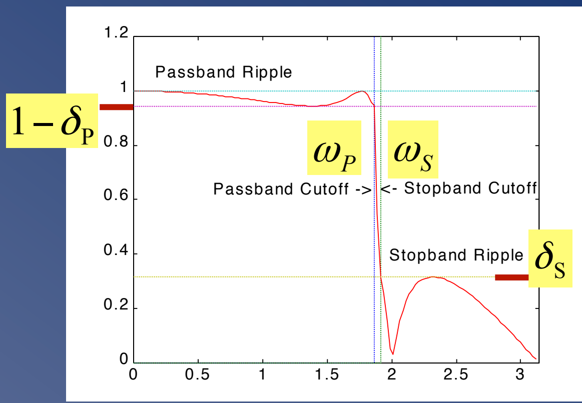
Filter Design Process

- **Step-1 : Define filter specs**
  Pass-band, stop-band, optimization criterion,…
- **Step-2 : Derive optimal transfer function**
  FIR or IIR filter design — **Chapter-3**
- **Step-3 : Filter realization** (block scheme/flow graph)
  Direct form realizations, lattice realizations,… — **Chapter-4**
- **Step-4 : Filter implementation** (software/hardware)
  Finite word-length issues, … — **Chapter-5**
  Question: implemented filter = designed filter ?
  'You can't always get what you want' -Jagger/Richards (?)

Filter Design Process

- **Step-1: Filter Specification**

  Example: Low-pass filter



Chapter-3 : Filter Design

- **FIR filters**
  - Linear-phase FIR filters
  - FIR design by optimization

    Weighted least-squares design, Minimax design
  - FIR design in practice

    `Windows' , Equiripple design, …

    Software (Matlab,…)
- **IIR filters**
  - Poles and zeros
  - IIR design by optimization

    Weighted least-squares design, Minimax design
  - IIR design in practice

    Software (Matlab,…)

FIR Filters

FIR filter = finite impulse response filter

$$H(z)=\frac{B(z)}{z^L}=b_0+b_1z^{-1}+\ldots+b_Lz^{-L}$$

- L poles at the origin z=0 (hence guaranteed stability)
- L zeros (zeros of B(z)), `all zero' filters
- Corresponds to difference equation

$$y[k]=b_0.u[k]+b_1.u[k-1]+\ldots+b_L.u[k-L]$$

- Hence also known as `moving average filters' (MA)
- Impulse response

$$h[0]=b_0,h[1]=b_1,\ldots,h[L]=b_L,h[L+1]=0,\ldots$$

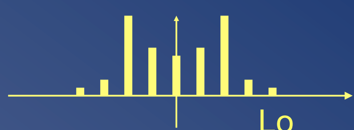
Linear Phase FIR Filters

- Non-causal zero-phase filters :

  Example: symmetric impulse response (length 2.Lo+1)

  h[-Lo],….h[-1], h[0] ,h[1],...,h[Lo]

  h[k]=h[-k], k=1..Lo



Frequency response is

$$e^{+jx}+e^{-jx}=2.\cos x$$

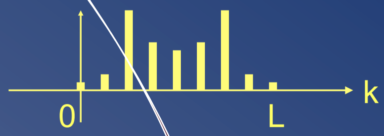
$$H(e^{j\omega})=\sum_{k=-Lo}^{+Lo}h[k].e^{-j\omega.k}=\ldots=\sum_{k=0}^{Lo}d_k.\cos(\omega.k) \qquad d_k=2.h[k] \text{ except } d_0=h[0]$$

2Lo+1 terms Lo+1 terms

i.e. real-valued (**=zero-phase**) transfer function

Linear Phase FIR Filters

- Causal linear-phase filters = non-causal zero-phase + delay
  Example: symmetric impulse response & L even
  - h[0],h[1],….,h[L]
  - L=2.Lo
  - h[k]=h[L-k], k=0..L



Frequency response is

$$H(e^{j\omega}) = \sum_{k=0}^{L} h[k].e^{-j\omega.k} = \ldots = e^{-j\omega.Lo}.\sum_{k=0}^{Lo} d_k.\cos(\omega.k) \qquad d_k = 2.h[k+Lo] \text{ except } d_0 = h[Lo]$$

= i.e. causal implementation of zero-phase filter, by introducing delay $z^{-Lo}\big|_{z=e^{j\omega}} = e^{-j\omega.Lo}$ ← phase is linear function of frequency

Linear Phase FIR Filters

Type-1	Type-2	Type-3	Type-4
L=2Lo=even	L=2Lo+1=odd	L=2Lo=even	L=2Lo+1=odd
symmetric	symmetric	anti-symmetric	anti-symmetric
h[k]=h[L-k]	h[k]=h[L-k]	h[k]=-h[L-k]	h[k]=-h[L-k]
$e^{-j\omega L/2}.\sum_{k=0}^{Lo} d_k.\cos(\omega.k)$	$e^{-j\omega L/2}\cos(\frac{\omega}{2}).\sum_{k=0}^{Lo} d_k.\cos(\omega.k)$	$je^{-j\omega L/2}\sin(\omega).\sum_{k=0}^{Lo-1} d_k.\cos(\omega.k)$	$j.e^{-j\omega L/2}\sin(\frac{\omega}{2}).\sum_{k=0}^{Lo} d_k.\cos(\omega.k)$
	zero at $\omega = \pi$	zero at $\omega = 0,\pi$	zero at $\omega = 0$
LP/HP/BP	LP/BP	BP	HP/BP

PS: `Modulating' Type-2 with 1,-1,1,-1,.. gives Type-4 (LP->HP)
PS: `Modulating' Type-4 with 1,-1,1,-1,.. gives Type-2 (HP->LP)
PS: `Modulating' Type-1 with 1,-1,1,-1,.. gives Type-1 (LP<->HP)
PS: `Modulating' Type-3 with 1,-1,1,-1,.. gives Type-3 (BP<->BP)

PS: IIR filters can NEVER have linear-phase property ! (proof see literature)

FIR Filter Design by Optimization

(I) Weighted Least Squares Design :

- Select one of the basic forms that yield linear phase e.g. Type-1

$$H(e^{j\omega}) = e^{-j\omega L/2} . \sum_{k=0}^{Lo} d_k . \cos(\omega . k) = e^{-j\omega L/2} . A(\omega)$$

- Specify desired frequency response (LP,HP,BP,…)

$$H_d(\omega) = e^{-j\omega L/2} . A_d(\omega)$$

- Optimization criterion is

$$\min_{d_0,\dots,d_{Lo}} \int_{-\pi}^{+\pi} W(\omega) \left| H(e^{j\omega}) - H_d(\omega) \right|^2 d\omega = \min_{d_0,\dots,d_{Lo}} \underbrace{\int_{-\pi}^{+\pi} W(\omega) \left| A(\omega) - A_d(\omega) \right|^2 d\omega}_{F(d_0,\dots,d_{Lo})}$$

where $W(\omega) \geq 0$ is a weighting function

FIR Filter Design by Optimization

- This is solved by replacing function F(..) by…

$$\underline{F}(d_0,\dots,d_{Lo}) = \sum_i W(\omega_i) . \left| A(\omega_i) - A_d(\omega_i) \right|^2$$

where the wi' s are a (sufficiently large) set of **sample frequencies**
This leads to an equivalent `discretized' quadratic optimization function

$$\underline{F}(d_0,\dots,d_{Lo}) = \sum_i W(\omega_i) \left\{ c^T(\omega_i) . \begin{bmatrix} d_0 \\ \vdots \\ d_{Lo} \end{bmatrix} - A_d(\omega_i) \right\}^2 = x^T . \underline{Q} . x - 2x^T . \underline{p} + \underline{\mu}$$

$$\underline{Q} = \sum_i W(\omega_i) . c(\omega_i) . c^T(\omega_i) \qquad \underline{p} = \sum_i W(\omega_i) . A_d(\omega_i) . c(\omega_i) \qquad \underline{\mu} = \dots$$

Optimal solution is $x_{OPT} = \underline{Q}^{-1} . \underline{p}$

FIR Filter Design by Optimization

- Can be supplemented with additional constraints, e.g. for pass-band and stop-band ripple control :

$$|A(\omega_{P,i})-1| \leq \delta_P, \quad \text{for pass-band freqs. } \omega_{P,1},\omega_{P,2},... \quad (\delta_P \text{ is pass-band ripple})$$
$$|A(\omega_{S,i})| \leq \delta_S, \quad \text{for stop-band freqs. } \omega_{S,1},\omega_{S,2},... \quad (\delta_S \text{ is stop-band ripple})$$

- The resulting optimization problem is :

minimize : $\underline{F}(d_0,...,d_{Lo}) = ...$ (=quadratic function)

$$x^T = \begin{bmatrix} d_0 & d_1 & ... & d_{Lo} \end{bmatrix}$$

subject to $A_P.x \leq b_P$ (=pass-band constraints)

$A_S.x \leq b_S$ (=stop-band constraints)

= `Quadratic Programming' problem

FIR Filter Design by Optimization

(II)`Minimax' Design :

- Select one of the basic forms that yield linear phase

e.g. Type-1 $H(e^{j\omega}) = e^{-j\omega L/2}.\sum_{k=0}^{Lo} d_k.\cos(\omega.k) = e^{-j\omega N/2}.A(\omega)$

- Specify desired frequency response (LP,HP,BP,…)

$$H_d(\omega) = e^{-j\omega L/2}.A_d(\omega)$$

- Optimization criterion is

$$\min_{d_0,...,d_{Lo}} \max_{-\pi\leq\omega\leq\pi} W(\omega).|H(e^{j\omega}) - H_d(\omega)| = \min_{d_0,...,d_{Lo}} \max_{-\pi\leq\omega\leq\pi} W(\omega).|A(\omega) - A_d(\omega)|$$

where $W(\omega) \geq 0$ is a weighting function

- Leads to `Semi-Definite Programming' (SDP) problem, for which efficient interior-point algorithms & software are available.

FIR Filter Design by Optimization

- **Conclusion:**
  (I) Weighted least squares design
  (II) Minimax design
  provide general `framework', procedures to translate filter design problems into standard optimization problems

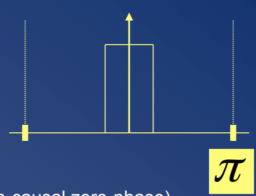
- **In practice (and in textbooks):**
  Emphasis on specific (ad-hoc) procedures :
  - Filter design based on `windows'
  - Equiripple design

FIR Filter Design using `Windows'

Example : Low-pass filter design

- Ideal low-pass filter is

$$H_d(\omega) = \begin{cases} 1 & |\omega| < \omega_C \\ 0 & \omega_C \prec |\omega| < \pi \end{cases}$$



- Hence ideal time-domain impulse response is (non-causal zero-phase)

$$h_d[k] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} H_d(e^{j\omega}).e^{j\omega k} d\omega = \ldots = \alpha.\frac{\sin(\omega_c k)}{\omega_c k} \qquad -\infty < k < \infty$$

- <u>Truncate</u> $h_d[k]$ to L+1 samples (L even):

$$h[k] = \begin{cases} h_d[k] & -L/2 < k < L/2 \\ 0 & \text{otherwise} \end{cases}$$

- Add delay to turn into causal filter

FIR Filter Design using `Windows'

Example : Low-pass filter design (continued)

- PS : It can be shown (use Parceval's theorem) that the filter obtained by such time-domain truncation is also obtained by using a weighted least-squares design procedure with the given H_d, and weighting function

$$W(\omega)=1$$

- Truncation corresponds to applying a `rectangular window' :

$$h[k]=h_d[k].w[k]$$

$$w[k]=\begin{cases} 1 & -L/2<k<L/2 \\ 0 & \text{otherwise} \end{cases}$$

- Can also apply other window fuctions, e.g., Han-, Hamming-, Blackman-, Kaiser window,…. (see textbooks). Window choice/design is trade-off between side-lobe levels (define peak pass-/stop-band ripple) and width main-lobe (defines transition bandwidth), see examples

FIR Equiripple Design

- Starting point is minimax criterion, e.g.

$$\min_{d_0,\ldots,d_{Lo}} \max_{0\le\omega\le\pi} W(\omega).\left|A(\omega)-A_d(\omega)\right| = \min_{d_0,\ldots,d_{Lo}} \max_{0\le\omega\le\pi} \left|E(\omega)\right|$$

- Based on theory of Chebyshev approximation and the `alternation theorem', which (roughly) states that the optimal d's are such that the `max' (maximum weighted approximation error) is obtained at L_o+2 extremal frequencies…

$$\max_{0\le\omega\le\pi}\left|E(\omega)\right|=\left|E(\omega_i)\right| \quad \text{for } i=1,..,Lo+2$$

…that hence will exhibit the same maximum ripple (`equiripple')

- Iterative procedure for computing extremal frequencies, etc. (Remez exchange algorithm, Parks-McClellan algorithm)
- Very flexible, etc., available in many software packages
- Details omitted here (see textbooks)

FIR Filter Design Software

- FIR Filter design abundantly available in commercial software
- Matlab:

b=fir1(L,Wn,type,window), windowed linear-phase FIR design, L is filter order, Wn defines band-edges, type is `high’,`stop’,…

b=fir2(L,f,m,window), windowed FIR design based on inverse Fourier transform with frequency points f and corresponding magnitude response m

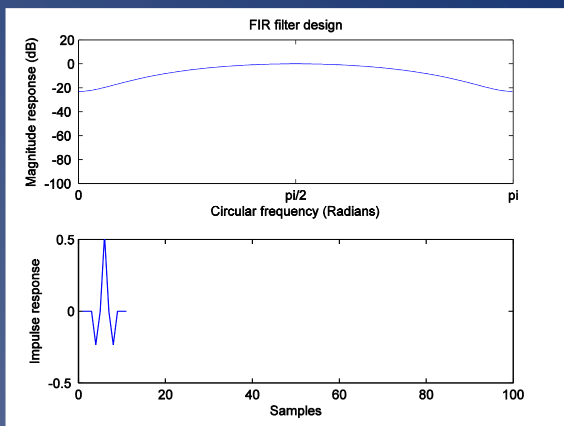
b=remez(L,f,m), equiripple linear-phase FIR design with Parks-McClellan (Remez exchange) algorithm

FIR Filter Design – Matlab Examples

1/8

- for filter_order=10:30:100
- % Impulse response
- b = fir1(filter_order,[W1 W2],'bandpass');
- % Frequency response
- % Plotting
- end

filter_order-10

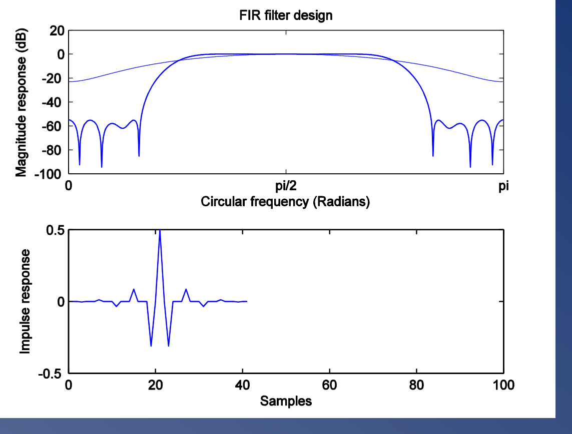


FIR Filter Design – Matlab Examples

2/8

- for filter_order=10:30:100
- % Impulse response
- b = fir1(filter_order,[W1 W2],'bandpass');
- % Frequency response
- % Plotting
- end

filter_order-40

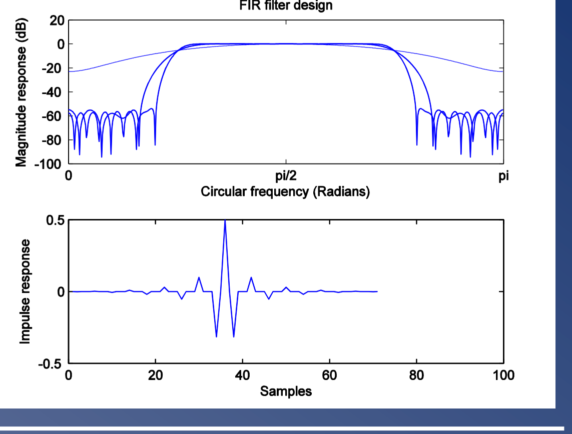


FIR Filter Design – Matlab Examples

3/8

- for filter_order=10:30:100
- % Impulse response
- b = fir1(filter_order,[W1 W2],'bandpass');
- % Frequency response
- % Plotting
- end

filter_order-70

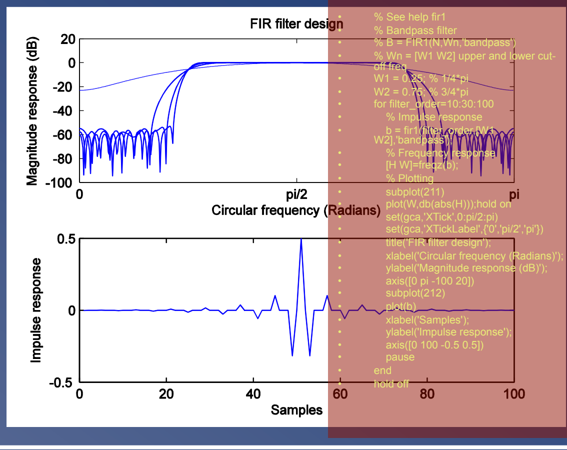


FIR Filter Design – Matlab Examples

4/8

- for filter_order=10:30:100
- % Impulse response
- b = fir1(filter_order,[W1 W2],'bandpass');
- % Frequency response
- % Plotting
- end

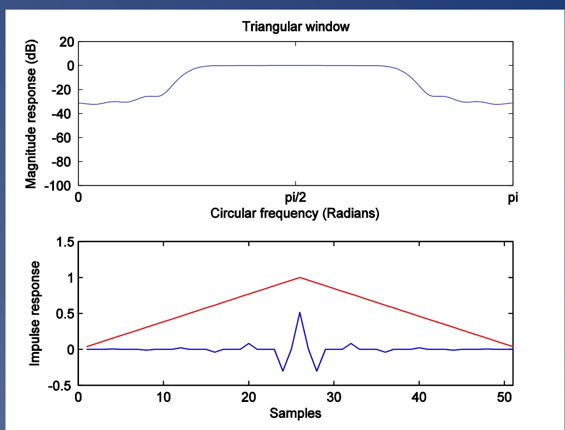
filter_order-100




```
% See help fir1
% Bandpass filter
% B = FIR1(N,Wn,'bandpass')
% Wn = [W1 W2] upper and lower cut-off freq
W1 = 0.25; % 1/4*pi
W2 = 0.75; % 3/4*pi
for filter_order=10:30:100
    % Impulse response
    b = fir1(filter_order,[W1 W2],'bandpass');
    % Frequency response
    [H W]=freqz(b);
    % Plotting
    subplot(211)
    plot(W,db(abs(H)));hold on
    set(gca,'XTick',0:pi/2:pi)
    set(gca,'XTickLabel',{'0','pi/2','pi'})
    title('FIR filter design');
    xlabel('Circular frequency (Radians)');
    ylabel('Magnitude response (dB)');
    axis([0 pi -100 20])
    subplot(212)
    plot(b)
    xlabel('Samples');
    ylabel('Impulse response');
    axis([0 100 -0.5 0.5])
    pause
end
hold off
```

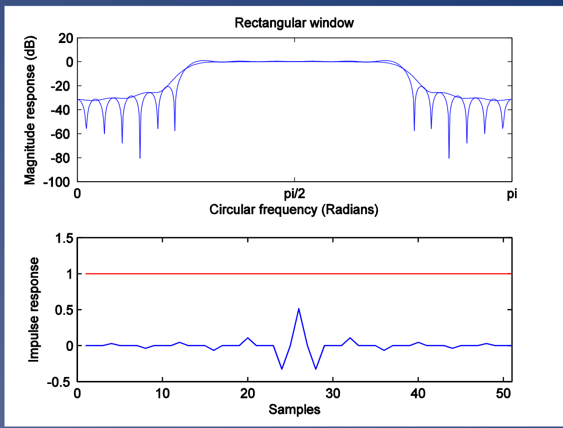

FIR Filter Design – Matlab Examples

5/8



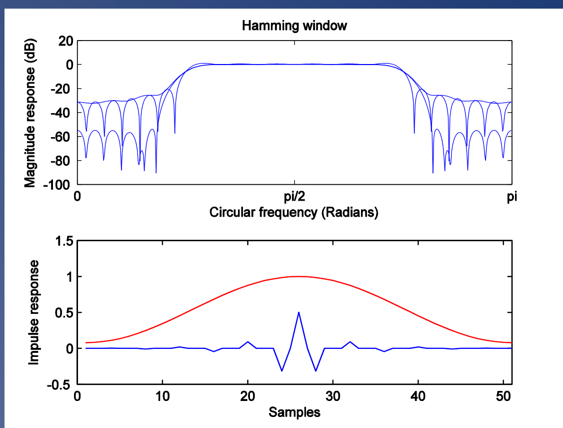
FIR Filter Design – Matlab Examples

6/8



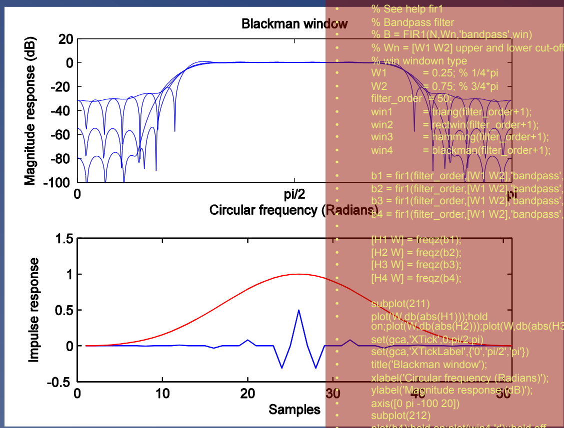
FIR Filter Design – Matlab Examples

7/8



FIR Filter Design – Matlab Examples

8/8




```
% See help fir1
% Bandpass filter
% B = FIR1(N,Wn,'bandpass',win)
% Wn = [W1 W2] upper and lower cut-off frequencies
% win windown type
W1          = 0.25; % 1/4*pi
W2          = 0.75; % 3/4*pi
filter_order  = 50;
win1        = triang(filter_order+1);
win2        = rectwin(filter_order+1);
win3        = hamming(filter_order+1);
win4        = blackman(filter_order+1);

b1 = fir1(filter_order,[W1 W2],'bandpass',win1);
b2 = fir1(filter_order,[W1 W2],'bandpass',win2);
b3 = fir1(filter_order,[W1 W2],'bandpass',win3);
b4 = fir1(filter_order,[W1 W2],'bandpass',win4);

[H1 W] = freqz(b1);
[H2 W] = freqz(b2);
[H3 W] = freqz(b3);
[H4 W] = freqz(b4);

subplot(211)
plot(W,db(abs(H1)));hold on;plot(W,db(abs(H2)));plot(W,db(abs(H3)));plot(W,db(abs(H4)));hold off
set(gca,'XTick',0:pi/2:pi)
set(gca,'XTickLabel',{'0','pi/2','pi'})
title('Blackman window');
xlabel('Circular frequency (Radians)');
ylabel('Magnitude response (dB)');
axis([0 pi -100 20])
subplot(212)
plot(b4);hold on;plot(win4,'r');;hold off
xlabel('Samples');
ylabel('Impulse response');
axis([0 filter_order+1 -0.5 1.5])
```


IIR filters

Rational transfer function :

- $$H(z)=\frac{B(z)}{A(z)}=\frac{b_0z^L+b_1z^{L-1}+\ldots+b_L}{z^L+a_1z^{L-1}+\ldots+a_L}=\frac{b_0+b_1z^{-1}+\ldots+b_Lz^{-L}}{1+a_1z^{-1}+\ldots+a_Lz^{-L}}$$

  L poles (zeros of A(z)) , L zeros (zeros of B(z))
- Infinitely long impulse response
- Stable iff poles lie inside the unit circle
- Corresponds to difference equation

$$y[k]+a_1.y[k-1]+\ldots+a_L.y[k-L]=b_0.u[k]+b_1.u[k-1]+\ldots+b_L.u[k-L]$$

$$y[k]=\underbrace{b_0.u[k]+b_1.u[k-1]+\ldots+b_L.u[k-L]}_{`MA'}\underbrace{-a_1.y[k-1]-\ldots-a_L.y[k-L]}_{`AR'}$$

= also known as `ARMA' (autoregressive-moving average)

IIR Filter Design

+

- Low-order filters can produce sharp frequency response
- Low computational cost (cfr. difference equation p.29)

-

- Design more difficult
- Stability should be checked/guaranteed
- Phase response not easily controlled
  (e.g. no linear-phase IIR filters)
- Coefficient sensitivity, quantization noise, etc. can be a problem (see Chapter-6)

IIR filters

Frequency response versus pole-zero location :

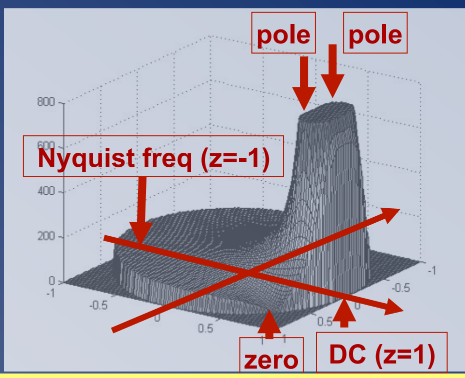
(~ Frequency response is z-transform evaluated on the unit circle)

Example

Low-pass filter with

poles at $0.80 \pm 0.20j$

zeros at $0.75 \pm 0.66j$



Pole near unit-circle introduces `peak' in frequency response

hence ***pass-band*** can be set by ***pole placement***

Zero near (or on) unit-circle introduces `dip' (or transmision zero) in freq. response

hence ***stop-band*** can be emphasized by ***zero placement***

IIR Filter Design by Optimization

(I) Weighted Least Squares Design :

- IIR filter transfer function is

$$H(z)=\frac{B(z)}{A(z)}=\frac{b_0+b_1z^{-1}+...+b_Lz^{-L}}{1+a_1z^{-1}+...+a_Lz^{-L}}$$

- Specify desired frequency response (LP,HP,BP,…)

$$H_d(\omega)$$

- Optimization criterion is

$$\min_{b_0,...,b_L,a_1,...,a_L}\underbrace{\int_{-\pi}^{+\pi}W(\omega)\left|H(e^{j\omega})-H_d(\omega)\right|^2 d\omega}_{F(b_0,...,b_L,a_1,...,a_L)}$$

where $W(\omega)\geq 0$ is a weighting function

- Stability constraint : $A(z)\neq 0, |z|\geq 1$

IIR Filter Design by Optimization

(II) `Minimax' Design :

- IIR filter transfer function is

$$H(z)=\frac{B(z)}{A(z)}=\frac{b_0+b_1z^{-1}+...+b_Lz^{-L}}{1+a_1z^{-1}+...+a_Lz^{-L}}$$

- Specify desired frequency response (LP,HP,BP,…)

$$H_d(\omega)$$

- Optimization criterion is

$$\min_{b_0,...,b_L,a_1,...,a_L}\max_{0\leq\omega\leq\pi} W(\omega).\left|H(e^{j\omega})-H_d(\omega)\right|$$

where $W(\omega)\geq 0$ is a weighting function

- Stability constraint :

$$A(z)\neq 0, |z|\geq 1$$

IIR Filter Design by Optimization

These optimization problems are significantly more difficult than those for the FIR design case… :

- **Problem-1**: Presence of denominator polynomial leads to non-linear/non-quadratic optimization
- **Problem-2**: Stability constraint
  (zeros of a high-order polynomial are related to the polynomial' s coefficients in a highly non-linear manner)
  - Solutions based on alternative stability constraints, that e.g. are affine functions of the filter coefficients, etc…
  - Topic of ongoing research, details omitted here

IIR Filter Design Software

- IIR filter design considerably more complicated than FIR design (stability, phase response, etc..)
- (Fortunately) IIR Filter design abundantly available in commercial software
- Matlab:

[b,a]=butter/cheby1/cheby2/ellip(L,…,Wn),

IIR LP/HP/BP/BS design based on **analog prototypes**, pre-warping, bilinear transform, …

immediately gives H(z) ☺

analog prototypes, transforms, … can also be called individually

filter order estimation tool

etc...