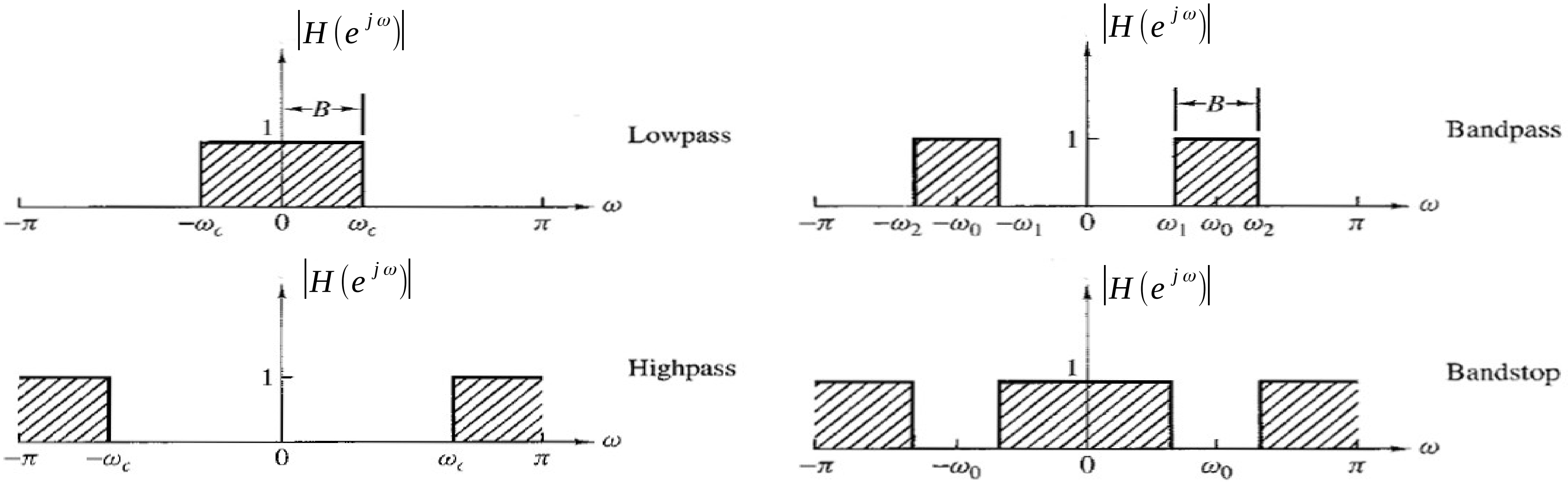
Digital filter design

- Introduction
- Specifications of filter properties
- Performance constraints
- FIR filters versus IIR filters
- Design of IIR filters
- Design by positioning zeros and poles
- Design via analog prototypes
- Design of FIR filters
- Ideal lowpass filter
- Design by windowing
- Minimax Weighted Error
- Design by iterative optimization
- (Additional materials)

Introduction

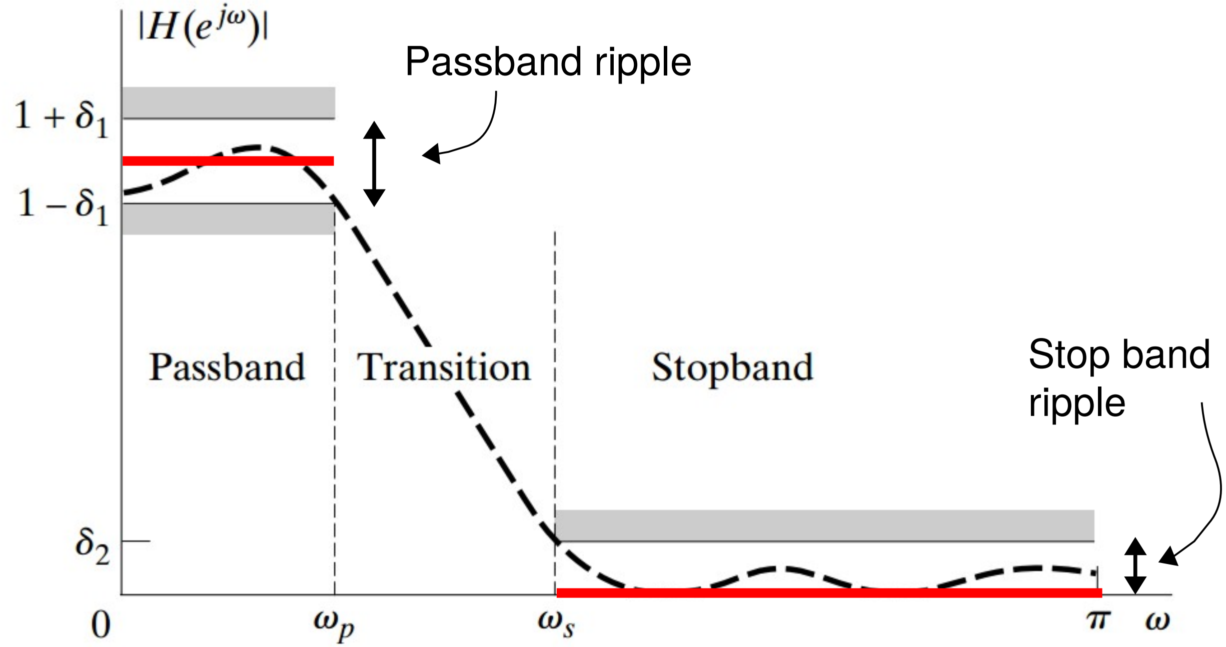
- From the DSP point of view, filter is a system which is changing the frequency content of signal with purpose
  - The most important is frequency aspect
  - Filter is changing amplitudes and phase angles of the components of input signal



Specifications of filter properties

- Specifications
  - Passband (~ 1)
  - Stop band (~ 0)
  - Transition band
  - Phase response (usually linear)

- Performance constraints
  - Smallest passband ripple (smallest 2.δ1)
  - Narrowest transition band
  - Greatest stop band attenuation (lowest stop band ripple, smallest δ2, or 2.δ2)



Real amplitude response ~ *Desired amplitude response*

$|H(e^{j\omega})|$ $|H_D(e^{j\omega})|$

δ1 - error in passband
δ2 - error in stop band

[Lyons]

Performance constraints

- Desired

Smallest
Passband ripple

Narrowest
Transition band

Greatest
Stop band attenuation

- Improving one usually worsens others
- Increasing filter order (cost) can improve all three measures
- Desired
  - the lowest computational complexity
  - the lowest filter order

FIR filters versus IIR filters

- Computational cost important

  → Use low complexity IIR

  (computational cost unimportant → use linear-phase FIR filters)
- Phase response important

  → Use linear-phase FIR filters

  (phase unimportant → Use simple IIR filters)

Design of IIR filters

- Design by positioning zeros and poles in the Z plane
- Design via analog prototypes
- Design using optimization methods (minimum Least Integral-squared Error (ISE) between desired, $|H_D(e^{j\omega})|$, and actual, $|H(e^{j\omega})|$, frequency response

Design by positioning zeros and poles

- Transfer function of a filter, H(z), is ratio of two polynomials
- Suitable for designing narrow band stop and bandpass filters (notch filters)

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} = \frac{b_0}{a_0} z^{N-M} \frac{(z-z_1)(z-z_2)\ \dots\ (z-z_M)}{(z-p_1)(z-p_2)\ \dots\ (z-p_N)}$$

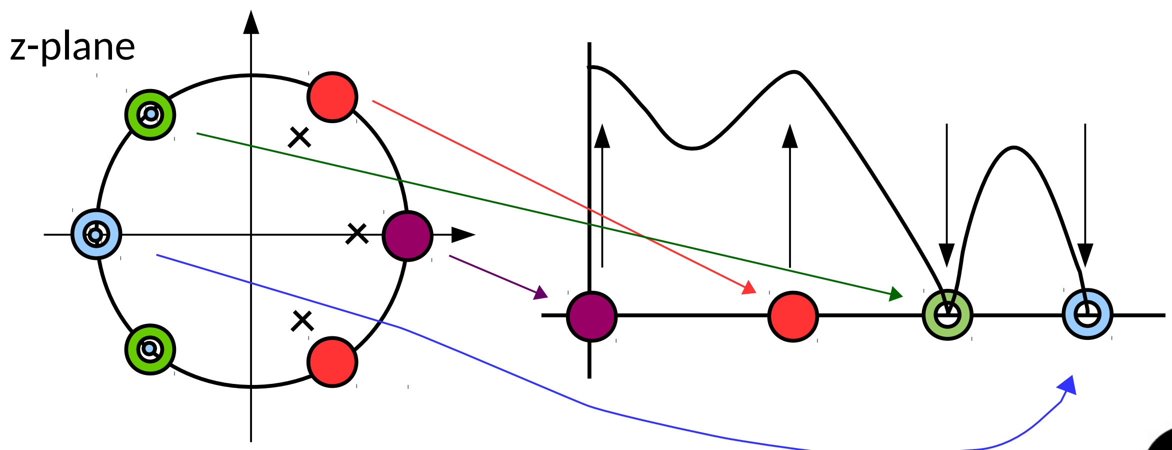
$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}}$$

- Frequency response of the filter is evaluated on the unit circle in the z-plane
- Zeros of *Y(z)* are zeros of the filter (lowering the value of frequency and amplitude response)
- Zeros of *X(z)* are poles of the filter (rising the value of frequency and amplitude response)
- Matlab:

  [b,a] = zp2tf(Z, P , 1)

  [h,w] = freqz(b, a, 512);

- Poles inside the unit circle (stability)
- Poles, zeros out of real axis → complex conjugate pair



Design via analog prototypes

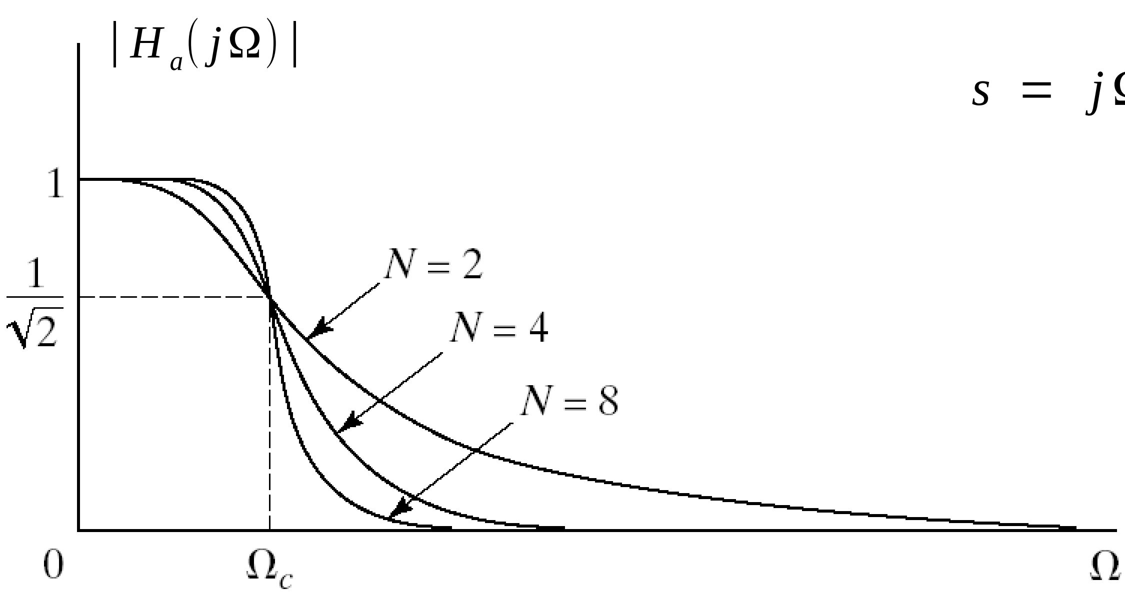
- Analog filter family
  - Butterworth
  - Chebyshev Type I
  - Chebyshev Type II
  - Elliptic
- Procedure
  - Transform digital filter specifications to equivalent analog filter specifications
  - Design analog fiter
  - Transform analog filter to digital filter

Design via analog prototypes

- Butterworth lowpass filters
  - Passband is designed to be maximally flat
  - The amplitude (squared) characteristic (analog domain)

$$|H_a(j\Omega)|^2 = \frac{1}{1 + (j\Omega / j\Omega_c)^{2N}} \qquad |H_a(s)|^2 = \frac{1}{1 + (\Omega / \Omega_c)^{2N}}$$

$$s = j\Omega$$

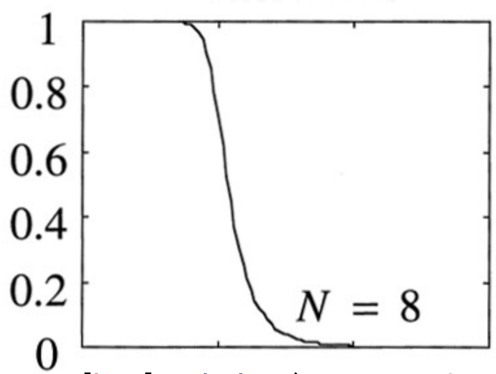


Design via analog prototypes

- Frequency responses of analog lowpass filters

[b, a] = butter(n, Wn, ftype);

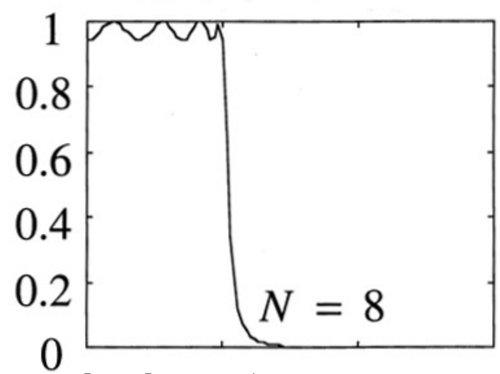
Butterworth



No ripples

[b, a] = cheby1(n, Rp, Wp, ftype);

Chebyshev Type I

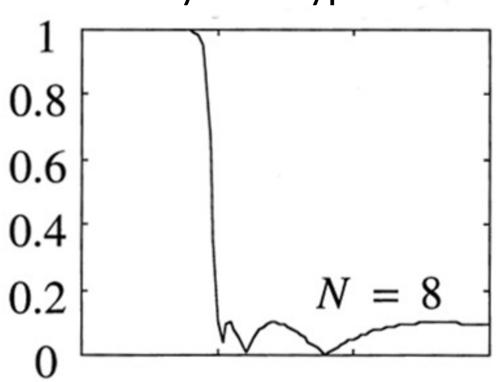


Ripples in the passband

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

[b, a] = cheby2(n, Rs, Ws, ftype);

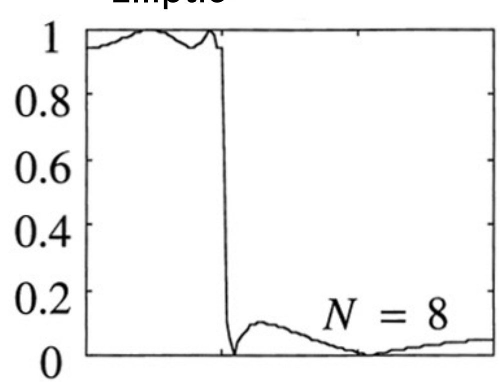
Chebyshev Type II



Ripples in the stop band

[b, a] = ellip(n, Rp, Rs, Wp, ftype);

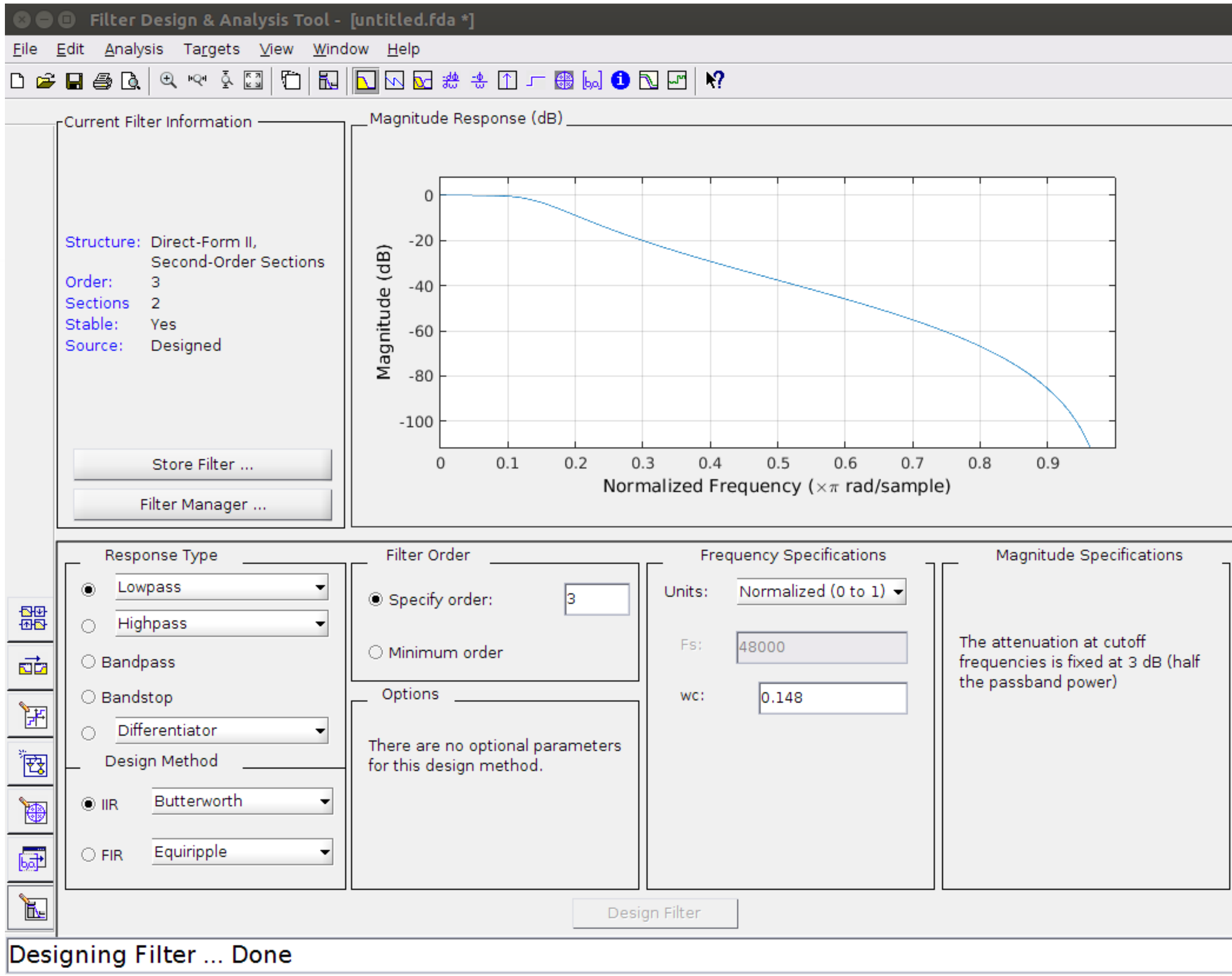
Elliptic



Ripples in the passband and stop band

Design via analog prototypes

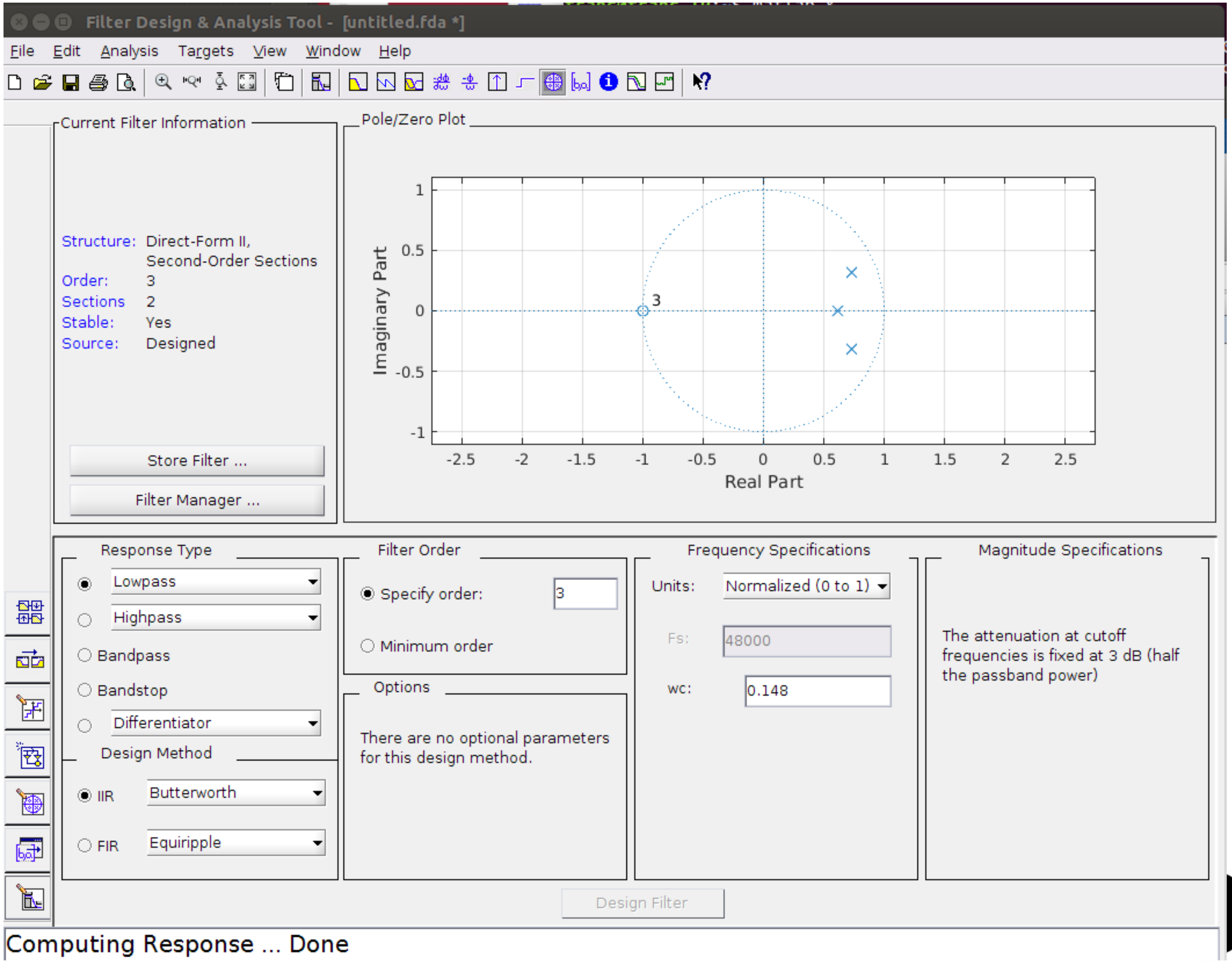
- MATLAB's Filter Design and Analysis Tool (fdatool)
- Example of 3^{th} order Butterworth lowpass filter
- Cuttof frequency *Wc* is 0.148



University of Ljubljana
Faculty of Computer and
Information Science

Design via analog prototypes

- MATLAB's Filter Design and Analysis Tool (fdatool)
- Example of 3^{th} order Butterworth lowpass filter
- Cuttof frequency *Wc* is 0.148
- Zero-pole plot



Design of FIR filters

- FIR filters

  - No poles (at z = 0 only), no precedent in analog filter design

- Approaches

  - Design by windowing (minimum Least Integral-Squared Error (ISE), or approximation, between desired, $|H_D(e^{j\omega})|$, and actual, $|H(e^{j\omega})|$, frequency response

$$\min_{b_0,\dots,b_M} \underbrace{\frac{1}{2\pi}\int_{-\pi}^{\pi} |H_D(e^{j\omega}) - H(e^{j\omega})|^2 \, d\omega}_{F(b_0,\dots,b_M)} = \epsilon^2 \qquad H(z) = \sum_{k=0}^{M} b_k z^{-k}$$

  - Design by iterative optimization (minimizing the maximum weighted approximation error → minimax criterion)

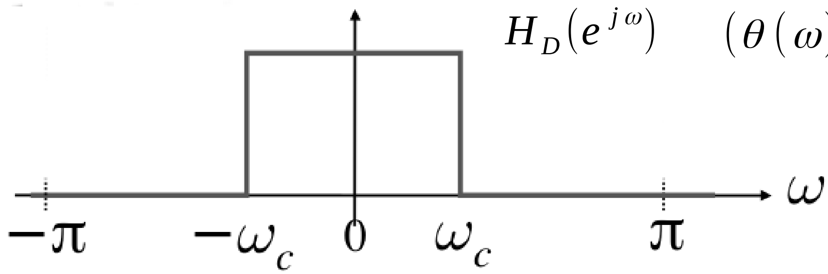
$$\min_{b_0,\dots,b_M} \underbrace{\frac{1}{2\pi}\int_{-\pi}^{\pi} |H_D(e^{j\omega}) - H(e^{j\omega})|^2 \, d\omega}_{F(b_0,\dots,b_M)} = \epsilon^2$$

Ideal lowpass filter

- Given ideal (desired) lowpass filter

("Brickwall LP filter")

$$H_D(e^{j\omega}) \qquad (\theta(\omega) = 0)$$



- Impulse response *h(n)*

$$h_D(n) = \mathrm{IDTFT}[\, H_D(e^{j\omega}) \,]$$

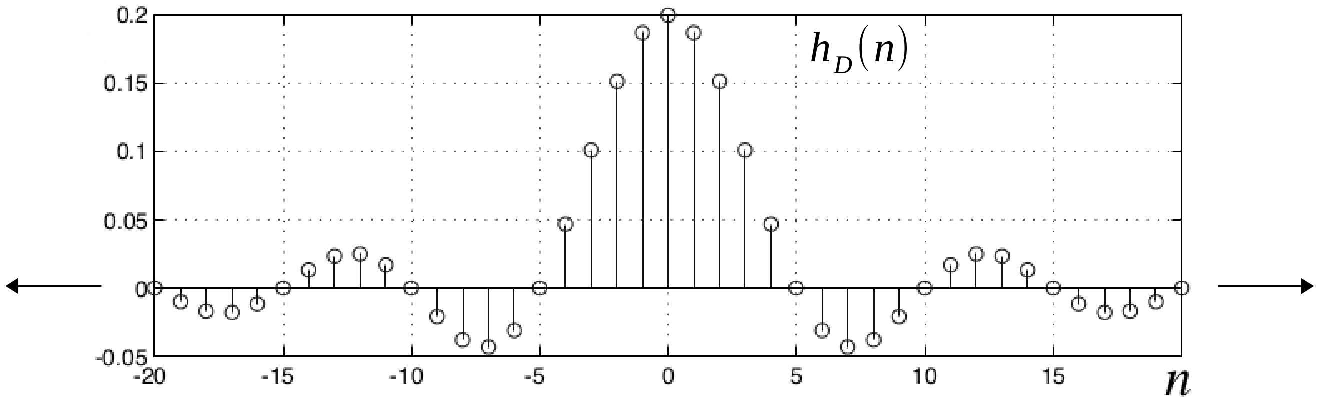
$$h_D(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_D(e^{j\omega})\, e^{j\omega n}\, d\omega = \frac{1}{2\pi}\int_{-\omega_C}^{\omega_C} e^{j\omega n}\, d\omega$$

- Ideal lowpass filter →

$$h_D(n) = \frac{\sin(\omega_C\, n)}{\pi\, n} \qquad (sinc)$$

Ideal lowpass filter

$$h_D(n) = \frac{\sin(\omega_C n)}{\pi n} \quad (\text{sinc})$$

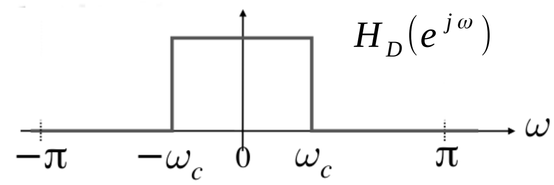


- Double infinite length of impulse response, $h_D(n)$, $n = -\infty, \ldots, \infty$
- Very long FIR (no rational polynomial)
- Nice frequency domain characteristics

Design by windowing

- Start with ideal frequency response, example: lowpass filter

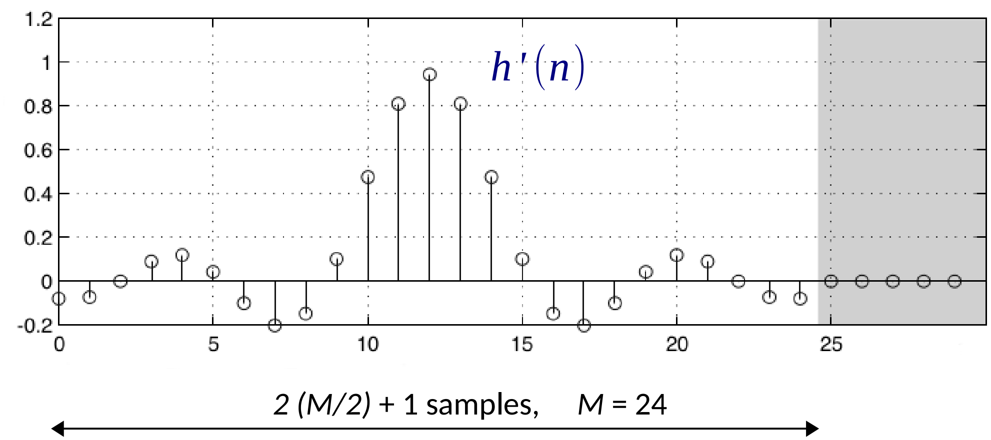
$$h_D(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_D(e^{j\omega})\, e^{j\omega n}\, d\omega$$



- Ideal frequency response is desired response
- Truncate $h_D(n)$, → $h(n)$, and make it causal (shift for M/2 samples), → $h'(n)$
- Ideal filter

$$h_D(n) = \frac{\sin(\omega_C\, n)}{\pi\, n}$$

- After truncating (→ minimum mean-squared approximation between desired, $|H_D(e^{j\omega})|$, and actual, $|H(e^{j\omega})|$, frequency resp.) (→ it becomes causal)



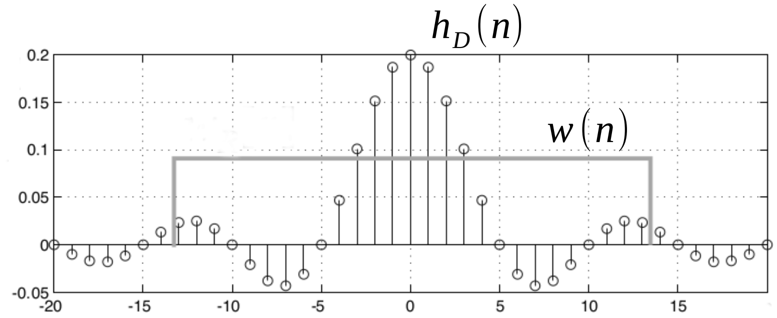
Design by windowing

- Truncation of *$h_D(n)$* to $2(M/2)+1$ points is multiplication by a rectangular window, *$w(n)$*

$$h(n) = h_D(n) \,.\, w(n)$$

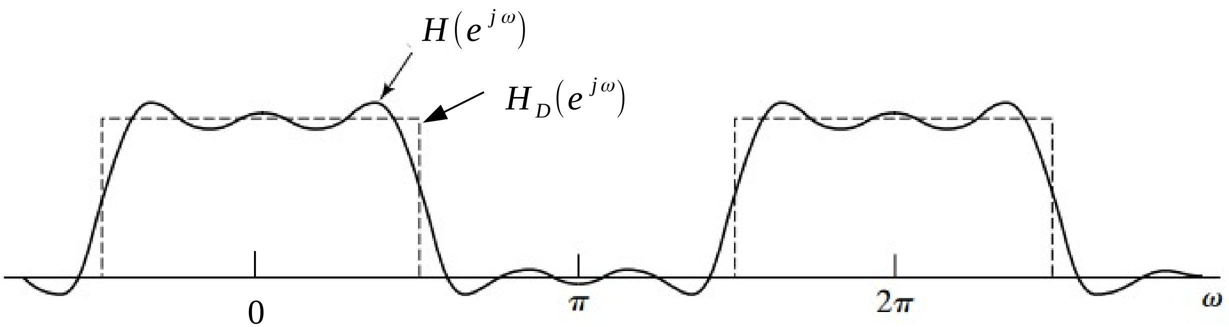
M = 26

$$w(n) = \begin{cases} 1 & -M/2 \le n \le M/2 \\ 0 & \text{otherwise} \end{cases}$$



- Infinite signal multiplied by a finite length window
- Actual frequency response, $H(e^{j\omega})$, in comparison to desired response, $H_D(e^{j\omega})$

→ The windowed version is smeared version of desired response (Gibbs phenomenon)

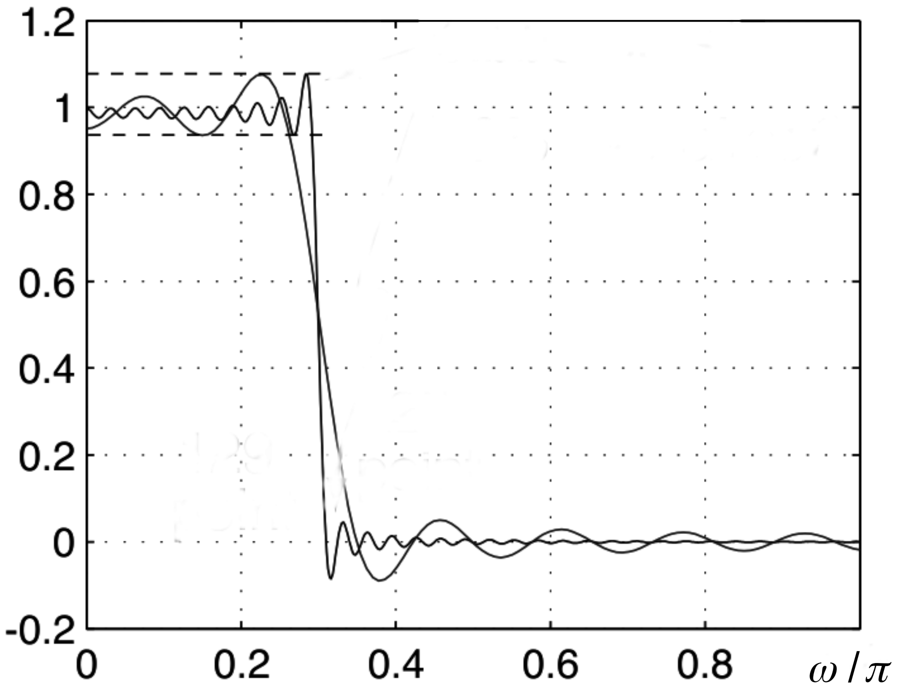


Design by windowing

- Truncated ideal filters have Gibbs' ears
- Example

  $2(M/2) + 1 = 27$, $2(M/2) + 1 = 129$
- Increasing filter length produces narrower ears and reduces Integral-Squared Error (ISE)
- Rectangular window provides the best approximation to a desired frequency response (in the sense of minimum mean-squared error)
- But, causes large errors around discontinuities

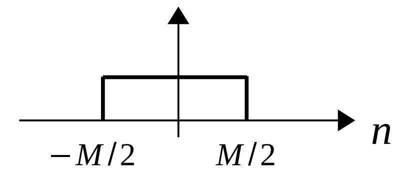
  (→ *filters are not optimal by minimax criterion*)



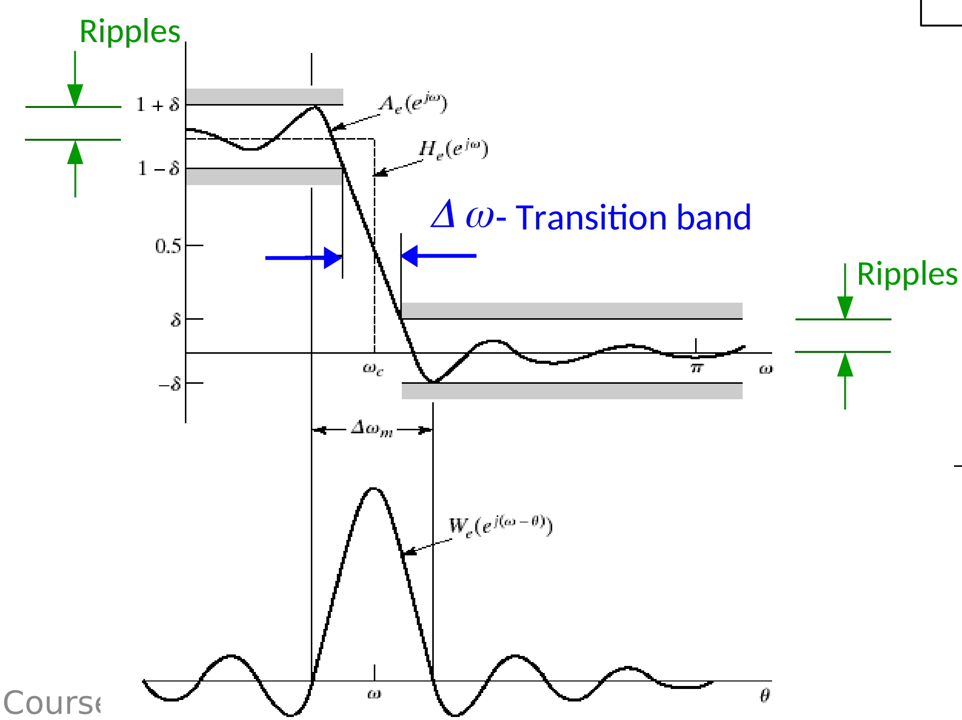
Design by windowing

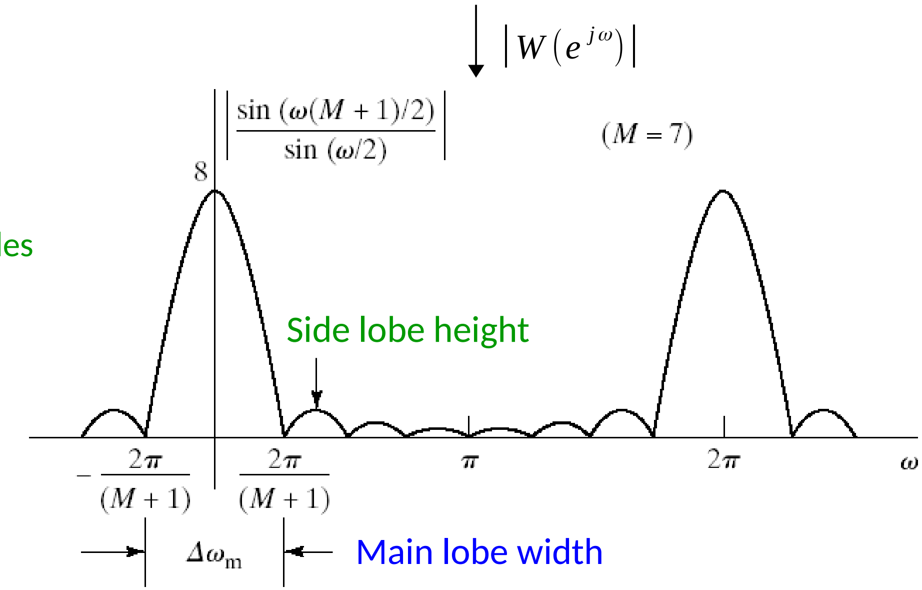
- Rectangular window
- Main lobe width determines transition band
- Side lobe height (invariant with length) determines ripples

$$w(n) = \begin{cases} 1 & -\frac{M}{2} \le n \le \frac{M}{2} \\ 0 & \text{otherwise} \end{cases}$$



$$W(e^{j\omega}) = \sum_{n=-M/2}^{M/2} e^{-j\omega n} = \frac{1-e^{-j\omega(M+1)}}{1-e^{j\omega}} = \frac{\sin\left((M+1)\frac{\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$





[Oppenheim, Schafer]

Design by windowing

- Windows for FIR filters
- Even though the rectangular window provides the best mean-squared error approximation to a desired frequency response, irregular distribution of error (irregular amplitudes of ripples) bother us
- Other windows differ in
  - Main lobe → Influence on transition band width
  - Side lobes → Influence on size of ripples near transition band
- Prefer windows with narrow main lobe and low side lobe height
- Rectangular window (high side lobes) $w(n) = 1 \qquad 0 \le n \le M$
- Hanning (Hann) (double width main lobe) $w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{M+1}\right)$
- Hamming (reduces first side lobe) $w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{M+1}\right)$
- Blackman (triple width main lobe) $w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{M+1}\right) + 0.08\cos\left(\frac{4\pi n}{M+1}\right)$

Design by windowing

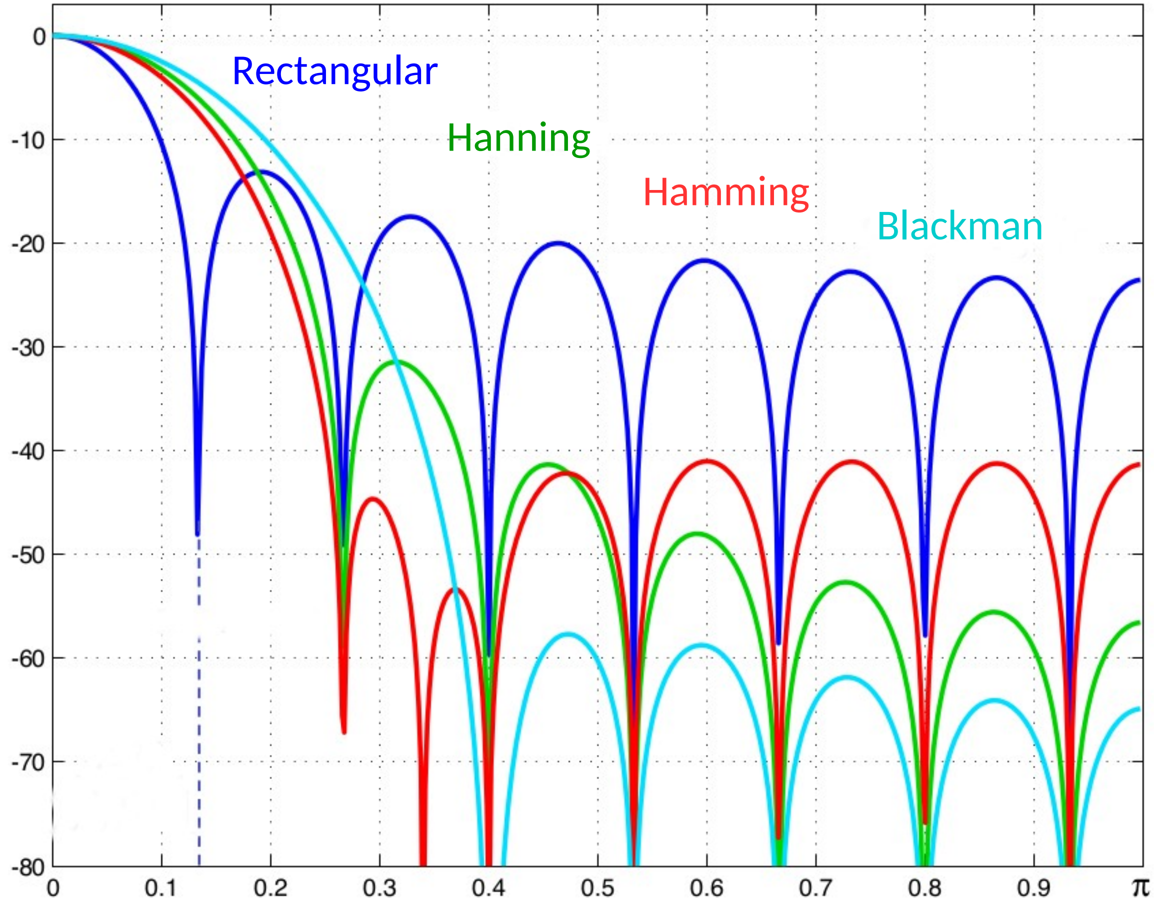
- Windows for FIR filters

- Width of main lobe

Rectangular	$4\pi/(M+1)$
Hanning	$8\pi/(M+1)$
Hamming	$8\pi/(M+1)$
Blackman	$12\pi/(M+1)$

- Peak side lobe amplitude

Rectangular	-13dB
Hanning	-31dB
Hamming	-41dB
Blackman	-57dB



Design by windowing

- Example
- Design a 25 point FIR lowpass filter with a cutoff of 500 Hz (*Fs* = 8k smp/s)
- No specific transition or ripple requirements → Compromise: Hamming
- Convert the frequency to rad/smp

$$\omega_C = \frac{500}{8000} \text{ x } 2\pi = 0.125\pi$$

1. Get ideal filter impulse response

$$\omega_C = 0.125\pi \quad \Rightarrow \quad h_D = \frac{\sin(0.125\pi n)}{\pi n}$$

2. Get window, *M+1* = 25 → *M/2* = 12

$$w(n) = 0.54 + 0.46\cos(2\pi n / 25), \quad -12 \le n \le 12$$

3. Apply window:

$$h(n) = d_D(n) \,.\, w(n) = \frac{\sin(0.125\pi n)}{\pi n}(0.54 + 0.46\cos(2\pi n / 25)), \quad -12 \le n \le 12$$

- Matlab: b = fir1(n, Wn, ftype, window);

$$H(z) = \sum_{k=0}^{M} b_k z^{-k} \rightarrow b_k = h(n)$$

Minimax Weighted Error

- Filter design by windows is simple but not optimal

- Object of optimization: FIR filters with generalized linear phase

  - Symmetric (or antisymmetric) impulse response

$$A_e(e^{j\omega}) = \sum_{n=-L}^{L} h_e(n) e^{-j\omega n}$$

- Design by iterative optimization
- Minimizing the maximum weighted approximation error

  → minimax (or Chebyshev) criterion between desired, $H_D(e^{j\omega})$, and actual, $A_e(e^{j\omega})$, frequency response

  => *Equiripple approximation*

$$\min_{\{h_e(n):\, 0 \le n \le L\}} \left(\max_{\omega \in F} |E(\omega)| \right)$$

$$E(\omega) = W(\omega)\left[H_D(e^{j\omega}) - A_e(e^{j\omega})\right]$$

$E(\omega)$ - Error

$W(\omega)$ - Weight

$H_D(e^{j\omega})$ - Desired

$A_e(e^{j\omega})$ - Actual

Design by iterative optimization

- The most popular method: Parks-McClellan algorithm
- The algorithm
  1. Define passband and stop band regions
  2. Define order, and error weights
  3. Calculate $A_e(e^{j\omega})$ and check the result
  4. Repeat previous step until optimal solution
- The algorithm is using (Chebyshev polynomial approximation, Remez exchange algorithm, alternation theorem)

- Matlab:

```
>> b = firpm(n, f, a, w, type);    % Optimal FIR digital filter
>>                                 % n: Order
>>                                 % f, a:  Desired amplitude response
>>                                 % w: Weights
>>                                 % ftype = 'low' | 'bandpass' | 'high' | 'stop'
>> ...
```

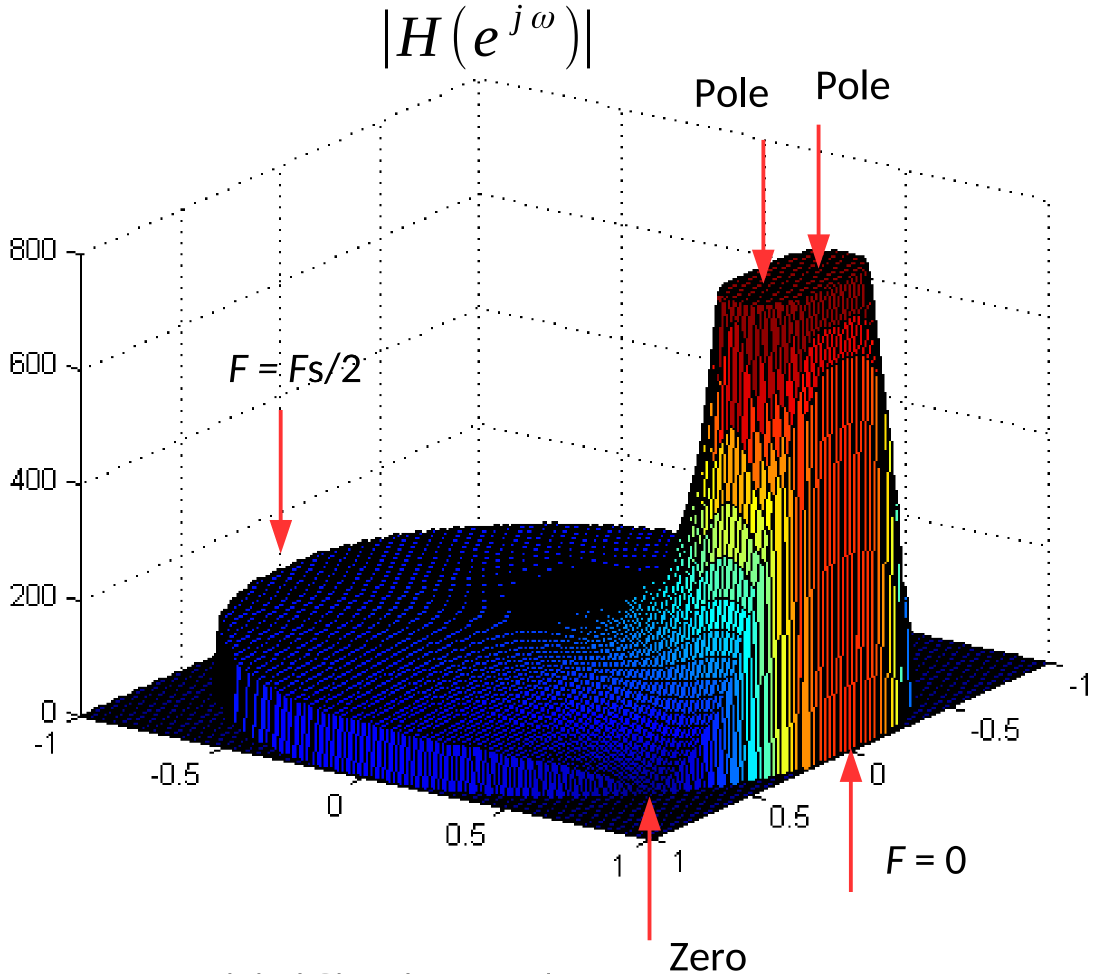
$$H(z) = \sum_{k=0}^{M} b_k z^{-k}$$

(Additional materials)

- Design by positioning zeros and poles, example
- Design via analog prototypes
- Design using optimization methods
- Least Integral-Squared Error
- Design by windowing
- Minimax Weighted Error
- Design by iterative optimization
- Design via analog prototypes, example
- Bilinear transformation method

Design by positioning zeros and poles, example

- Example, lowpass IIR filter
  - Poles: 0.8 ± j0.2
  - Zeros: 0.75 ± j0.66

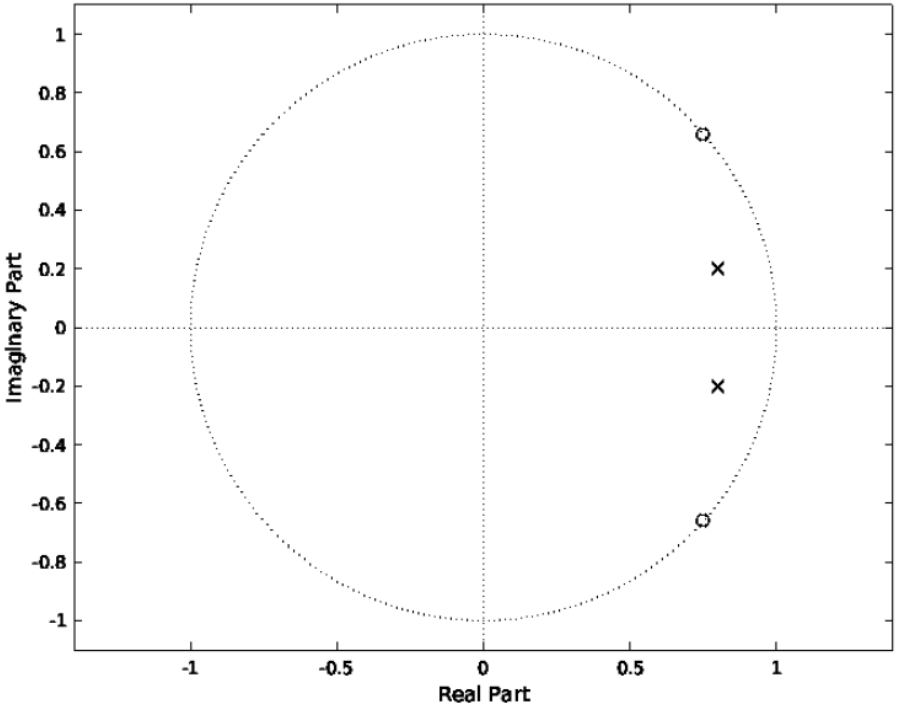


Design by positioning zeros and poles, example

- Example, lowpass IIR filter
  - Poles: 0.8 ± j0.2
  - Zeros: 0.75 ± j0.66

- MATLAB

```
>> P(1,1) = 0.8 + 0.2i;
>> P(2,1) = 0.8 - 0.2i;
>> Z(1,1) = 0.75 + 0.66i;
>> Z(2,1) = 0.75 – 0.66i;
>>
>> zplane(Z, P)
>>
>> [b,a] = zp2tf(Z, P , 1)
b = 1.0000   -1.5000    0.9981
a = 1.0000   -1.6000    0.6800
>>
```



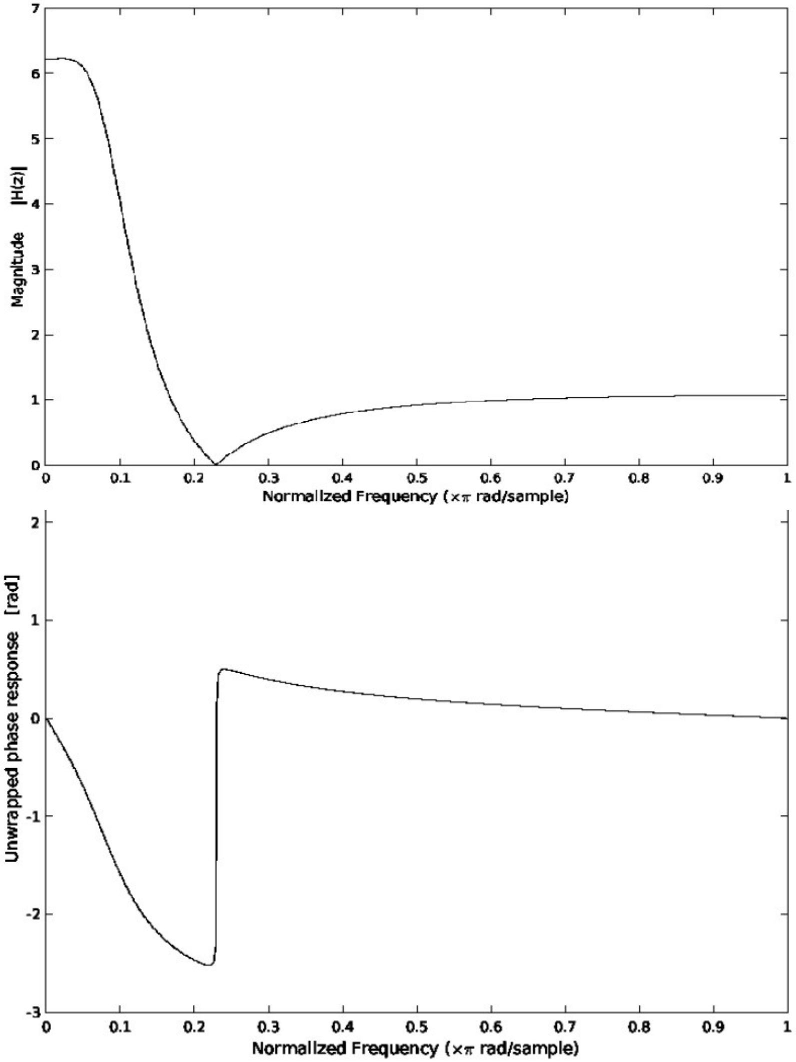
$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - 1.5\,z^{-1} + 0.9981\,z^{-2}}{1 - 1.6\,z^{-1} + 0.6800\,z^{-2}}$$

Design by positioning zeros and poles, example

- Example, lowpass IIR filter
  - Poles: 0.8 ± *j*0.2
  - Zeros: 0.75 ± *j*0.66

- MATLAB

```
>>
>> [h,w] = freqz(b, a, 512);
>> plot(w/pi, abs(h));
>> ...
>> [phi,w] = phasez(b, a, 512)
>> plot(w/pi, phi);
>> ...
>>
```



Design via analog prototypes

- MATLAB

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

```
>> % Filter design method:  Analog prototyping
>>       % Frequency transformation and filter discretization from the poles
>>       % and zeros of a classical prototype filter in the continuous domain
>>
>> [b, a] = butter(n, Wn, ftype);          % Butterworth filter
>>                             % Order: 2n,    Wn: Normalized cuttof frequencies, ftype: filter type
>> [b, a] = cheby1(n, Rp, Wp, ftype);   %  Chebyshev Type I filter
>>                             % Order: 2n,    Rp: decibels of peak-to-peak passband ripple
>> [b, a] = cheby2(n, Rs, Ws, ftype);    % Chebyshev Type II filter
>>                             % Order: 2n,    Rs: deciBels of stopband attenuation down
>>                             %                        from the peak passband value
>> [b, a] = ellip(n, Rp, Rs, Wp, ftype); % Elliptic filter,    Order: 2n
>> ...                         %  ftype = 'low' | 'bandpass' | 'high' | 'stop'
>>                            % Recall also
>> [Z, P, K] = tf2zpk(b, a);                  % Z: zeros,  P: poles,  K: gain
>> y = filter(b, a, x);   % Filter,  y – Output signal,  x – Input signal
```

Design using optimization methods

- Digital filter design as optimization problem
- Object of optimization (frequency response of a system)

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}} = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}}$$

Desired: $H_D(e^{j\omega})$ Actually obtained: $H(e^{j\omega})$

- Optimization criterion (unweighted squared error criterion)

  - Least-squares fit (Least Integral-Squared Error - ISE)

$$\min_{b_0,\ldots,b_M,\, a_1,\ldots,b_N} \underbrace{\frac{1}{2\pi}\int_{-\pi}^{\pi} |H_D(e^{j\omega}) - H(e^{j\omega})|^2 \, d\omega}_{F(b_0,\ldots,b_M,\, a_1,\ldots,a_N)} = \epsilon^2$$

Design using optimization methods

- MATLAB

```
>> % Filter design method:  By approximating a piecewise linear magnitude response
>> % Using a least-squares fit to a specified frequency response
>>
>> [b, a] = yulewalk(n, f, m);    % Recursive IIR digital filter
>>                                % Order: n
>>                                % f, m:  Desired amplitude response
```

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

Least Integral-Squared Error

- Given desired (ideal) frequency response, $H_D(e^{j\omega})$, what is the best finite *h(n)* of length *M+1*, *n = -M/2 .. M/2*, to approximate it?
- Object of optimization (frequency response of an FIR system)

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}} = b_0 + b_1 z^{-1} + \dots + b_M z^{-M}$$

Desired: $H_D(e^{j\omega})$ Actually obtained: $H(e^{j\omega})$

- Optimization criterion (Least Integral-Squared Error – ISE)

$$\min_{b_0,\dots,b_M} \underbrace{\frac{1}{2\pi}\int_{-\pi}^{\pi} |H_D(e^{j\omega}) - H(e^{j\omega})|^2 \, d\omega}_{F(b_0,\dots,b_M)} = \epsilon^2$$

$$H_D(e^{j\omega}) = \mathrm{DTFT}[h_D(n)]$$
$$H(e^{j\omega}) = \mathrm{DTFT}[h(n)]$$

- Since *h(n)* exists only for *n = -M/2 .. M/2*

$$h(n) = h_D(n), \quad -M/2 \le n \le M/2$$

→ The best finite *h(n)* to approximate $H_D(e^{j\omega})$ in the sense of minimum mean-squared error approximation is truncated $h_D(n)$

Least Integral-Squared Error

- Given desired (ideal) frequency response, $H_D(e^{j\omega})$, what is the best finite *h(n)* of length *M+1, n = -M/2 .. M/2,* to approximate it?
- Try to minimize least Integral-Squared Error (ISE) of frequency responses (unweighted squared error criterion)

$$\min \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_D(e^{j\omega}) - H(e^{j\omega})|^2 \, d\omega = \epsilon^2$$

$$H(e^{j\omega}) = \mathrm{DTFT}[h(n)]$$
$$H_D(e^{j\omega}) = \mathrm{DTFT}[h_D(n)]$$

- Since *h(n)* exists only for *n = -M/2 .. M/2*

$$\Rightarrow \sum_{n=-M/2}^{M/2} |h_D(n) - h(n)|^2 + \sum_{-\infty}^{-M/2-1} |h_D(n)|^2 + \sum_{M/2+1}^{\infty} |h_D(n)|^2 = \epsilon^2 \qquad \text{(By Parseval)}$$

i.e., the ISE is minimized by $h(n) = h_D(n), \quad -M/2 \le n \le M/2$

→ Thus, truncated IDTFT of desired frequency response, $H_D(e^{j\omega})$, best approximates the desired frequency response (in the sense of minimum mean-squared error)

→ The best finite *h(n)* to approximate the desired frequency response (in the sense of minimum mean-squared error) is truncated $h_D(n)$

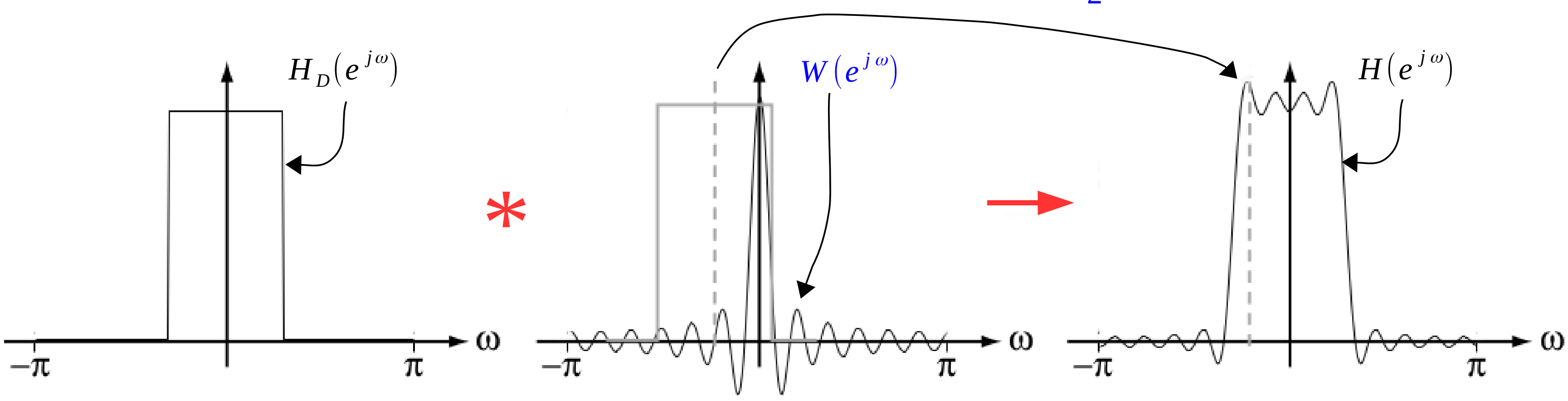
Design by windowing

- Where Gibbs phenomenon comes from?
- Multiplication in time domain is convolution in frequency domain

$$h_D(n)\,.\,w(n) \quad \leftrightarrow \quad \frac{1}{2\pi}\int_{-\pi}^{\pi} H_D(e^{j\omega})\, W(e^{j(\omega-\theta)})\, d\theta$$

- Frequency response of truncated response, $H(e^{j\omega})$, is convolution of ideal frequency response, $H_D(e^{j\omega})$, and frequency response of rectangular window, $W(e^{j\omega})$

$$W(e^{j\omega}) = \sum_{n=-M/2}^{M/2} e^{-j\omega n} = \frac{\sin\left((M+1)\frac{\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$



Design by windowing

- MATLAB

```
>> % Filter design method:  Windowing ideal impulse response
>>       % Using the specified window
>>
>> b = fir1(n, Wn, ftype, window)
>>                        % n: Order, Wn: Normalized cuttof frequencies, ftype: filter type
>> ...                    % ftype = 'low' | 'bandpass' | 'high' | 'stop'
>>                        % window = rectwin(), hann(), hamming(), blackman(), ...
```

$$H(z) = \sum_{k=0}^{M} b_k z^{-k}$$

Minimax Weighted Error

- Filter design by windows is simple but not optimal
- Alternative criteria can give better results
- Object of optimization: FIR filters with generalized linear phase

  - Type I linear-phase FIR filter (order *M*, even, length *M*+1, odd)

$$h_e(n) = h_e(-n) \qquad A_e(e^{j\omega}) = \sum_{n=-L}^{L} h_e(n) e^{-j\omega n} = h_e(0) + \sum_{n=1}^{L} 2\, h_e(n) \cos(\omega n), \quad L = M/2$$

- Optimization criterion (minimax or Chebyshev criterion)

  - Minimax fit (minimizing the maximum weighted approximation error)

    => *Equiripple approximation*

$$\min_{\{h_e(n):\, 0 \le n \le L\}} \left(\max_{\omega \in F} |E(\omega)| \right)$$

$$E(\omega) = W(\omega)\,[H_D(e^{j\omega}) - A_e(e^{j\omega})]$$

$E(\omega)$ - Error

$W(\omega)$ - Weight

$H_D(e^{j\omega})$ - Desired

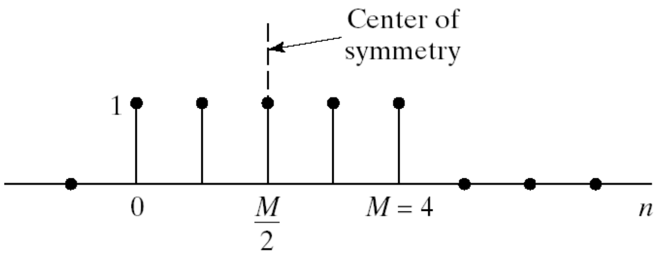
F is closed subset of $0 \le \omega \le \pi$ such that $0 \le \omega \le \omega_p$ and $\omega_s \le \omega \le \pi$

Design by iterative optimization

- The most popular method: Parks-McClellan algorithm
- Consider the design of Type I FIR filter (order *M*, even, length *M*+1, odd)

$$h_e(n) = h_e(-n) \qquad A_e(e^{j\omega}) = h_e(0) + \sum_{n=1}^{L} 2\, h_e(n)\cos(\omega n), \;\; L = M/2$$

- Can be delayed by sufficient amount (*M/2*) to make it causal



$$h(n) = h_e(n - M/2) = h(M-n) \qquad H(e^{j\omega}) = A_e(e^{j\omega})\, e^{-j\omega M/2}$$

- Goal is to approximate the desired response $H_D(e^{j\omega})$ with $A_e(e^{j\omega})$

[Oppenheim, Schafer]

Design by iterative optimization

- The most popular method: Parks-McClellan algorithm

- The algorithm
  1. Define passband and stop band regions
  2. Define order, and error weights
  3. Calculate $A_e(e^{j\omega})$ and check the result
  4. Repeat previous step until optimal solution

- It uses
  - Chebyshev polynomial approximation theory
  - Remez exchange algorithm (procedure of iterative approaching to optimal solution, 3. and 4.)
  - Alternation theorem (determines when the solution is optimal)

Design by iterative optimization

- MATLAB

```
>> % Filter design method:  Parks-McClellan algorithm
>> % Using equiripple approach over sub-bands of the frequency range
>>
>> b = firpm(n, f, a, w, type);   % Optimal FIR digital filter
>>                                % n: Order
>>                                % f, a:  Desired amplitude response
>>                                % w: Weights
>>                                % ftype = 'low' | 'bandpass' | 'high' | 'stop'
>> ...
```

Design via analog prototypes, example

- IIR filter design steps
  - Choose analog filter family
    * Butterworth
    * Chebyshev Type I
    * Chebyshev Type II
    * Elliptic
  - Choose analog-digital transformation method
    * Impulse invariance (aliasing problem)
    * Bilinear transformation (nonlinear transformation, frequency warping)
  - Procedure
    * Transform digital filter specifications to equivalent analog filter specifications
    * Design analog filter
    * Transform analog filter to digital filter

Design via analog prototypes, example

- Filter type: discrete-time lowpass filter
- Design method: the bilinear transformation
- The specifications for the filter to be designed
  - Passband frequency: $\omega_p = 0.1\,\pi$, Stop band frequency: $\omega_s = 0.3\,\pi$
  - Passband ripple: $-1\,dB \le |H(e^{j\omega})| \le 0\,dB, \quad |\omega| \le \omega_p$
  - Stop band ripple: $|H(e^{j\omega})| \le -20\,dB, \quad \omega_s \le |\omega| \le \pi$
  - In terms of parameters δ_1 and δ_2

$$1 - \sigma_1 = 10^{-1/20} = 0.89125 \qquad \sigma_2 = 10^{-20/20} = 0.1$$

  - Define new parameters

$$k_1 = \frac{1}{(1-\sigma_1)^2} = 0.79433 \qquad k_2 = \frac{1}{\sigma_2^2} - 1 = 99$$

  - Selected filter: Butterworth filter

$$|H_a(j\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2N}}$$

$|H_a(j\Omega)|^2$ equals 1.0 when $\Omega = 0$ and 1/2 when $\Omega = \Omega_c$

Design via analog prototypes, example

- Converting the critical frequencies ω_p and ω_s to their continuous-time counterparts produces

$$\Omega_p = 2\tan\left(\frac{\omega_p}{2}\right) = 0.3168 \ \text{rad/smp}$$

$$\Omega_s = 2\tan\left(\frac{\omega_s}{2}\right) = 1.0191 \ \text{rad/smp}$$

- Determining the order of the filter by the design equation

$$N = \frac{1}{2}\frac{\log\left(\frac{k_2}{k_1}\right)}{\log\left(\frac{\Omega_s}{\Omega_p}\right)} = 2.4546 \quad \rightarrow \quad N = 3, \quad 2N = 6$$

- Critical (cutoff) frequency

$$\Omega_c = \frac{\Omega_s}{k_2^{\left(\frac{1}{2N}\right)}} = 0.4738 \ \text{rad/smp}$$

Design via analog prototypes, example

- MATLAB

```
>> Rp = 1;            % Passband ripple
>> Rs = 20;           % Stop band ripple
>> Wp = 0.3168;       % Passband frequency (continuous time)
>> Ws = 1.0191;       % Stop band frequency (continuous case)
>> [N, Wn] = buttord(Wp, Ws, Rp, Rs, 's')
>>                    % Filter order, N,  and Critical (cutoff) frequency, Ωc,  i.e., Wn
>>                    % N = 3,  Wn = 4.738
>>  ...
```

Design via analog prototypes, example

- Determine transfer function of analog filter

$$|H_a(s)|^2 = \frac{1}{1 + (s/j\Omega_c)^{2N}}$$

- Poles are given by

$$s_k = \Omega_c e^{j\left(\frac{\pi}{2} + \frac{\pi}{2N} + \frac{2\pi k}{2N}\right)}, \quad k = 0, 1, \ldots, 2N-1$$

- Take *N* poles with negative real parts for *Ha(s)*

$$s_0 = -0.2369 + j0.4103 = 0.4738 e^{j2\pi/3}$$
$$s_1 = -0.4738 + j0.0000 = 0.4738 e^{j\pi}$$
$$s_2 = -0.2369 - j0.4103 = 0.4738 e^{-j2\pi/3}$$

- MATLAB

```
>> [Z, P, K] = butter(N, Wn, 's')
P =
 -0.2369 + 0.4103i
 -0.2369 - 0.4103i
 -0.4738 + 0.0000i
K =   0.1064     % Gain
>>
```

Design via analog prototypes, example

- Transfer function of analog filter

$$H_a(s) = \frac{\Omega_c^3}{(s-s_0)(s-s_1)(s-s_2)}$$

- Transform zeros and poles from s-plane (continuous) to z-plane

$$H_d(z) = H_a\left[2\frac{1-z^{-1}}{1+z^{-1}}\right] = \frac{\Omega_c^3(1+z^{-1})^3}{8(1-z^{-1}-s_0)(1-z^{-1}-s_1)(1-z^{-1}-s_2)}$$

- MATLAB

```
>> [num, den] = zp2tf(Z, P, K)
num =
     0      0      0      0.1064
den =
 1.000   0.9476   0.4490   0.1064
>>
```

- MATLAB

```
>> [Zd, Pd, Kd] = bilinear(Z, P, K,1)
Zd =
  -1   -1   -1
Pd =
  0.7300 + 0.3173i
  0.7300  - 0.3173i
  0.6169 + 0.0000i
Kd =
   0.0083   % Gain
>>
```

Design via analog prototypes, example

- Transfer function of digital filter

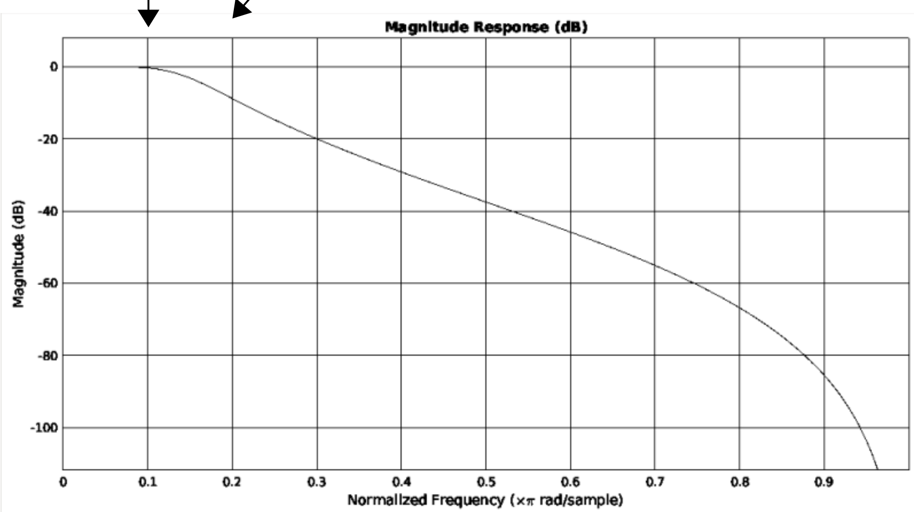
$$H_d(z) = \frac{0.0083 + 0.0249\,z^{-1} + 0.0249\,z^{-2} + 0.0083\,z^{-3}}{1.0000 - 2.0769\,z^{-1} + 1.5343\,z^{-2} - 0.3909\,z^{-3}}$$

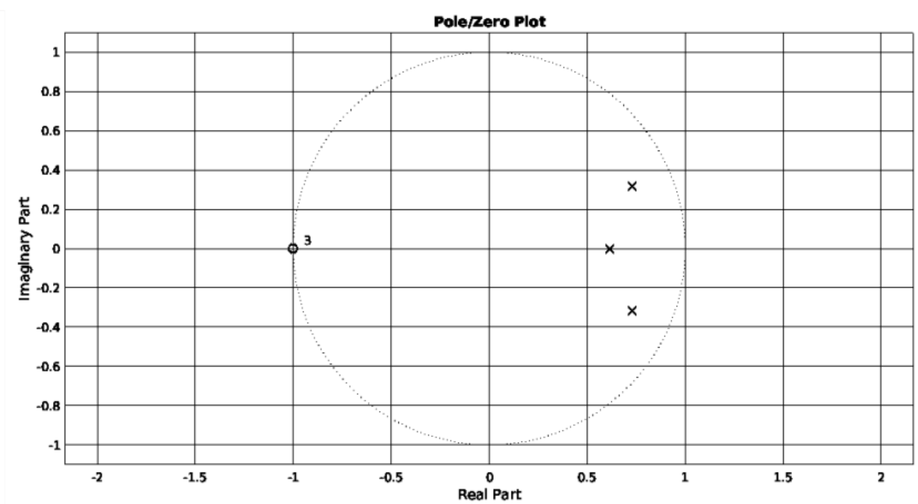
$$H_d(z) = 0.0083\,\frac{1 + 3\,z^{-1} + 3\,z^{-2} + 1\,z^{-3}}{1 - 2.0769\,z^{-1} + 1.5343\,z^{-2} - 0.3909\,z^{-3}}$$

$\omega_p(-1\,dB) = 0.1\,\pi$ $\omega_s(-15\,dB) = 0.3\,\pi$

- MATLAB

```
>> [numd, dend] = zp2tf(Zd, Pd, Kd)
numd =
   0.0083    0.0249    0.0249    0.0083
dend =
  1.0000   -2.0769    1.5342   -0.3909
>> fvtool(numd, dend);
>> y = filter(numd, dend, x);
```





Bilinear transformation method

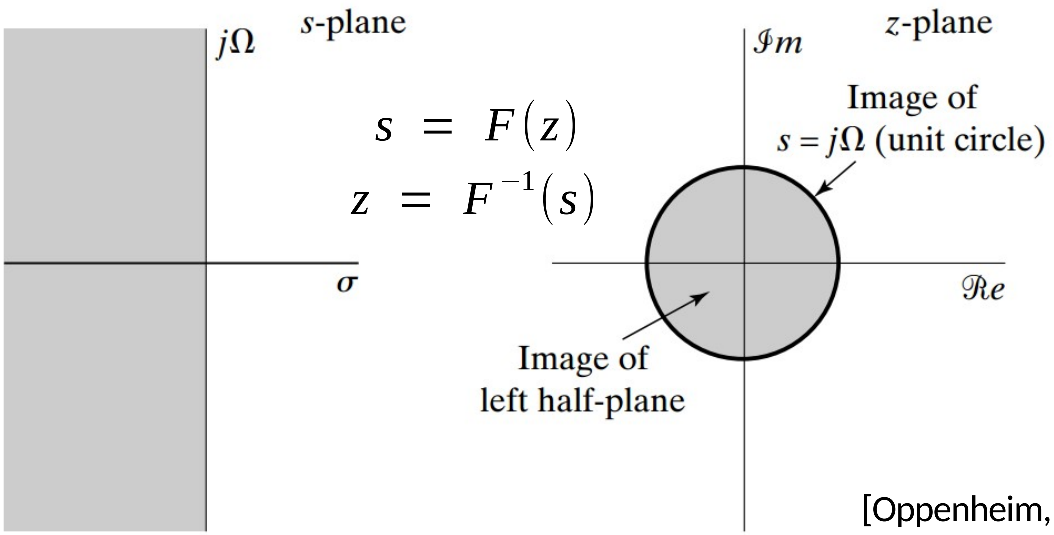
- How we can map continuous-time filters to discrete-time domain?

→ transformation $H_c(s) \rightarrow H(z)$, $s = \sigma + j\Omega$, where Ω is the analog frequency

$$H(z) = H_a(s)|_{s=F(z)}$$

where $s = F(z)$ maps s-plane ↔ z-plane

→ every value of $H(z)$ is a value of $H_c(s)$ somewhere on the s-plane and vice-versa



[Oppenheim, Schafer]

Bilinear transformation method

- Let *Hc(s)* be the transfer function of the prototype analog filter
- The transfer function *H(z)* of the digital filter is obtained by substituting

$$s = F(z) = \frac{2}{T_d}\frac{1 - z^{-1}}{1 + z^{-1}} \qquad s = \sigma + j\Omega$$

into the expression of *Hc(s)*

$$H(z) = H_a(s)|_{s=F(z)} = H_a\left[\frac{2}{T_d}\frac{1 - z^{-1}}{1 + z^{-1}}\right]$$

- Solving for *z*

$$z = \frac{1 + \frac{T_d}{2}s}{1 - \frac{T_d}{2}s} = \frac{1 + \sigma\frac{T_d}{2} + j\Omega\frac{T_d}{2}}{1 - \sigma\frac{T_d}{2} - j\Omega\frac{T_d}{2}} \qquad s = \sigma + j\Omega$$

Bilinear transformation method

- On the unit circle, $|z| = 1$, $(\sigma = 0)$, $z = e^{j\omega}$, the transform becomes

$$z = \frac{1 + j\Omega \frac{T_d}{2}}{1 - j\Omega \frac{T_d}{2}} = e^{j\omega}$$

- To derive relation between discrete-time ω and continuous-time frequency Ω

$$z = e^{j\omega} \quad => \quad s = \frac{2}{T_d}\frac{1 - e^{-j\omega}}{1 + e^{-j\omega}} = \sigma + j\Omega = \frac{2j}{T_d}\tan\left(\frac{\omega}{2}\right)$$

yielding

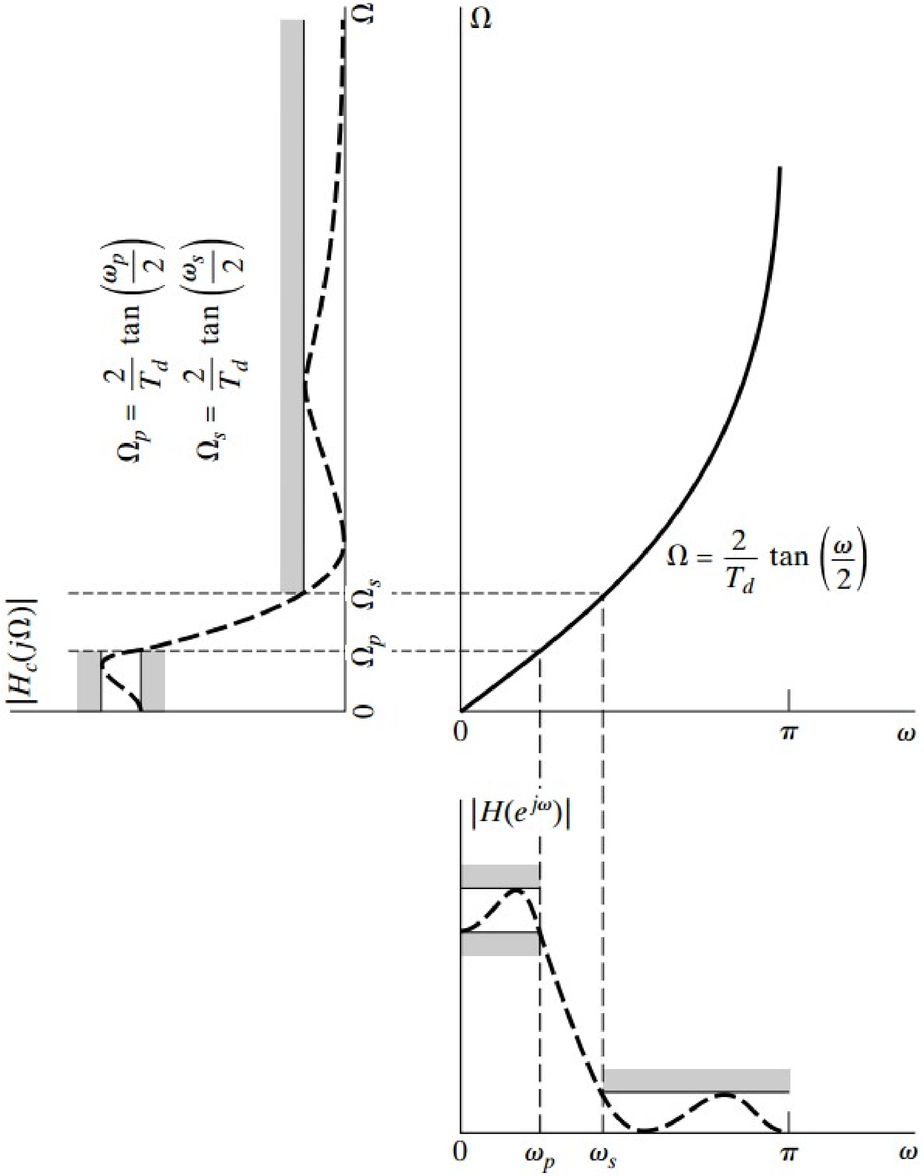
$$\Omega = \frac{2}{T_d}\tan\left(\frac{\omega}{2}\right) \qquad \omega = 2\tan^{-1}\left(\frac{\Omega T_d}{2}\right)$$

Bilinear transformation method

- Infinite range of continuous-time frequency $-\infty < \Omega < \infty$ is mapped to finite discrete-time frequency range $-\pi < \omega < \pi$

- Bilinear transform makes frequency warping (warped frequency axis) but leaves the same gain and phase

$$H(e^{j\omega}) = H_a(j\Omega)\,|_{\omega = 2\tan^{-1}(\Omega T_d / 2)}$$



Bilinear transformation method

- The design parameter T_d is usually set to equal 1 to simplify the expressions

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