Module 4

Digital Filters - Implementation and Design

Contents

- Signal Flow Graphs
  - Basic filtering operations
- Digital Filter Structures
  - Direct form FIR and IIR filters
  - Filter transposition
  - Linear phase FIR filter structures
  - Finite precision effects
- FIR and IIR Filter Design Techniques
  - Windows
  - Bilinear transformations

Reading:

Chapter 7, Proakis and Manolakis

Design and Implementation

- Filter Design
  - Given some frequency response specification, determine the type, order and coefficients of a digital filter to best meet the specification.
- Filter Implementation
  - Given some filter design, including the type, order and coefficients of the filter, determine a way of implementing the filtering operation using appropriate hardware or software.

- A filter can be specified by a difference equation

$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

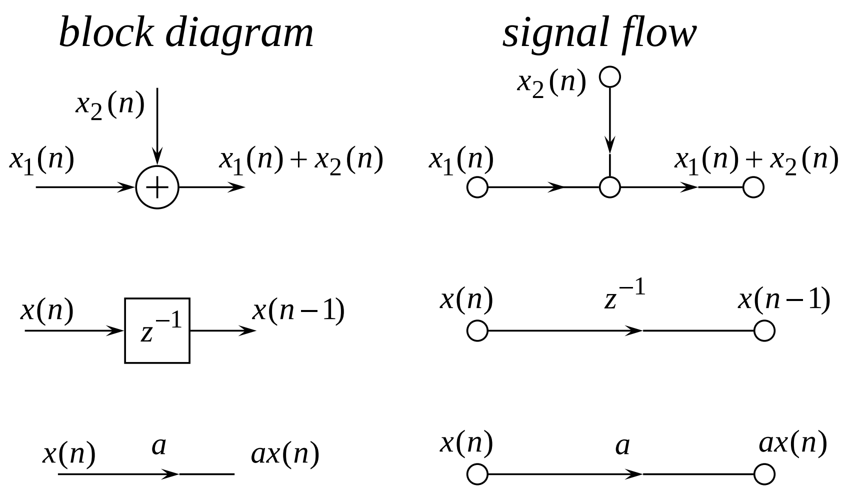
- or by a system function

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

- Such filters can be implemented in many different ways
  - equivalent in function for infinite precision computation
  - not equivalent for fixed point computation
    - some may be more robust to rounding effects

Signal Flow Graphs

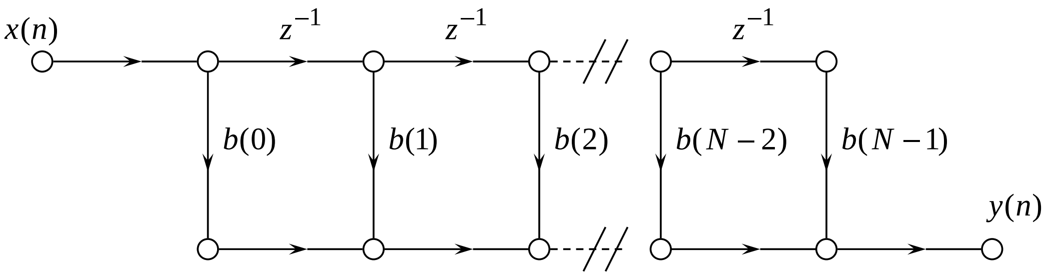
- Filters can be described in terms of 3 basic operations
- addition
- delay
- multiplication by a constant



Finite Impulse Response Filters (FIR)

- System function $H(z)=\sum_{k=0}^{M} b_k z^{-k}$
- Difference equation $y(n)=\sum_{k=0}^{M} b_k x(n-k)$
  - Output is weighted sum of current and previous M inputs
  - The filter has <u>order</u> M
  - The filter has $M+1$ <u>taps</u>, i.e. impulse response is of length $M+1$
  - $H(z)$ is a polynomial in z^{-1} of order M
  - $H(z)$ has M <u>poles</u> at $z = 0$ and M <u>zeros</u> at positions in the z-plane determined by the coefficients b_k

◆ Direct Form FIR Filter



- Also known as moving average (MA) and non-recursive

Infinite Impulse Response Filters (IIR)

◆ System function

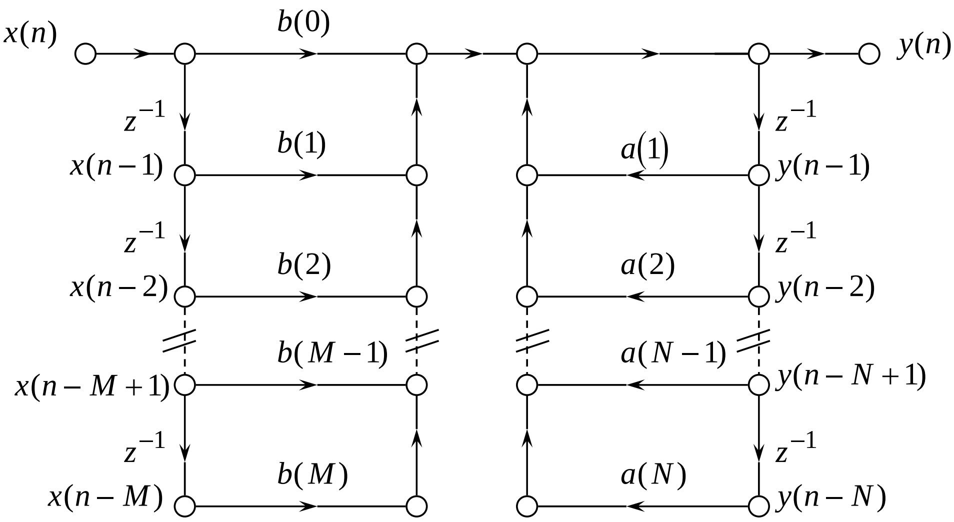
$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

◆ Difference equation

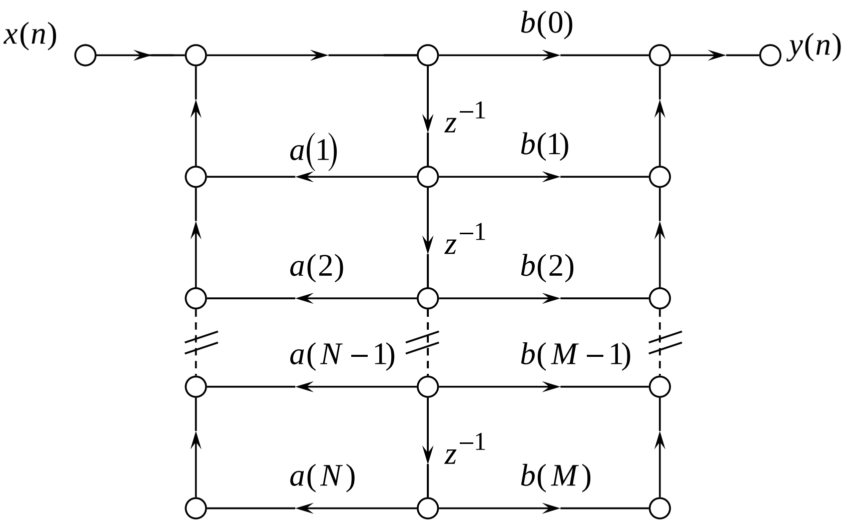
$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

- Output is weighted sum of current and previous M inputs and previous N outputs
- $H(z)$ has N poles at positions determined by a_k and M zeros at positions in the z-plane determined by b_k

◆ Direct Form 1 IIR Filter (for *M*=*N*)



◆ Direct Form 2 IIR Filter (for *M*=*N*)



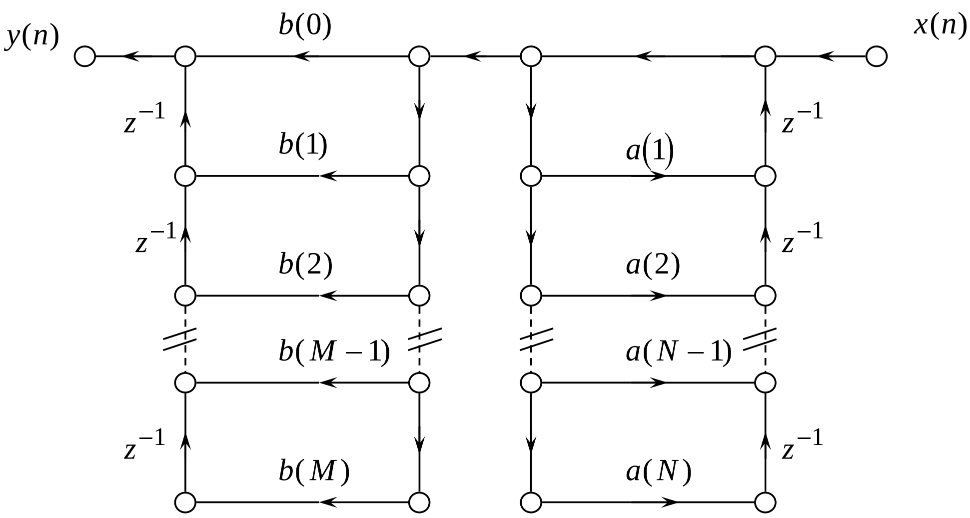
- Canonical Form
  - Filter structures which have the minimum number of delay elements are said to be in canonical form
    - i.e., minimum number of branches labelled z^{-1}
  - Minimum number of delays given by maximum of (M, N)

- Many other filter structures can be obtained by re-formulating the difference equations
  - two forms will be studied in this course: transposed form and linear phase FIR
  - many other forms will not be studied, e.g., cascade form, parallel form etc.

Transposed Forms

- Transposing a signal flow graph
  - reverse the directions of all branches
  - swap input and output
  - transfer function is unchanged

- Transposed Direct Form 1 IIR Filter (for M=N)
  - Note: drawn "backwards" for easy comparison with non-transposed form. Normally redrawn with input on left.



Linear Phase FIR Filters

- For causal linear phase FIR filters, the coefficients are symmetric $$h(n) = h(N-1-n)$$
- Linear phase filters do not introduce any phase distortion
  - they only introduce delay
  - The delay introduced is $(N-1)/2$ samples
- Zeros occur in mirror image pairs
  - if z_0 is a zero, then $1/z_0$ is also a zero
- Symmetry leads to efficient implementations
  - N/2 multiplications (N even) or (N+1)/2 multiplications (N odd) per output sample instead of N for the general case.

- The filter's z transform can be written

$$
\begin{aligned}
H(z) &= \sum_{n=0}^{N-1} h(n) z^{-n} \\
&= \sum_{n=0}^{(N/2)-1} h(n)\left[z^{-n} + z^{-(N-1-n)}\right], \; N \text{ even} \\
&= \sum_{n=0}^{[(N-1)/2]-1} h(n)\left[z^{-n} + z^{-(N-1-n)}\right] + h\left(N-1/2\right) z^{-[(N-1)/2]}, \; N \text{ odd}
\end{aligned}
$$

- and hence

$$
\begin{aligned}
H(e^{j\omega}) &= e^{-j\omega[(N-1)/2]}\left\{\sum_{n=0}^{(N/2)-1} 2h(n)\cos\left[\omega\left(n - \frac{N-1}{2}\right)\right]\right\}, \; N \text{ even} \\
&= e^{-j\omega[(N-1)/2]}\left\{h\left(\frac{N-1}{2}\right) + \sum_{n=0}^{(N-3)/2} 2h(n)\cos\left[\omega\left(n - \frac{N-1}{2}\right)\right]\right\}, \; N \text{ odd}
\end{aligned}
$$

FIR / IIR Pros and Cons

- FIR
  - ✓ Linear Phase
    - constant group delay at any frequency
  - ✓ Good CAD support for design
  - ✓ Can't be unstable
    - good for adaptive filters
  - ✓ Robust to numerical errors
    - eg: rounding in fixed point arithmetic
  - ✗ Large number of taps required for accurate frequency selectivity
    - high computational load
- IIR
  - ✓ based on "well-known" analogue concepts
  - ✗ Non-linear phase
    - delay varies with frequency
  - ✓ Good CAD support for design
  - ✗ Can be unstable
  - ✗ Rounding errors can accumulate and cause serious inaccuracies
    - limit cycles
  - ✓ Small number of taps required
    - low computational load

Finite Precision Effects

- DSP algorithms are implemented in hardware that can represent numbers only to finite precision
  - e.g., 16 bit precision for Texas Instruments TMS320C5x

- This results in
  - rounding of arithmetic operations
  - rounding of filter coefficients

- Rounding of filter coefficients
  - effective position of poles and zeros moved
    - frequency response errors introduced
    - IIR filters may go unstable

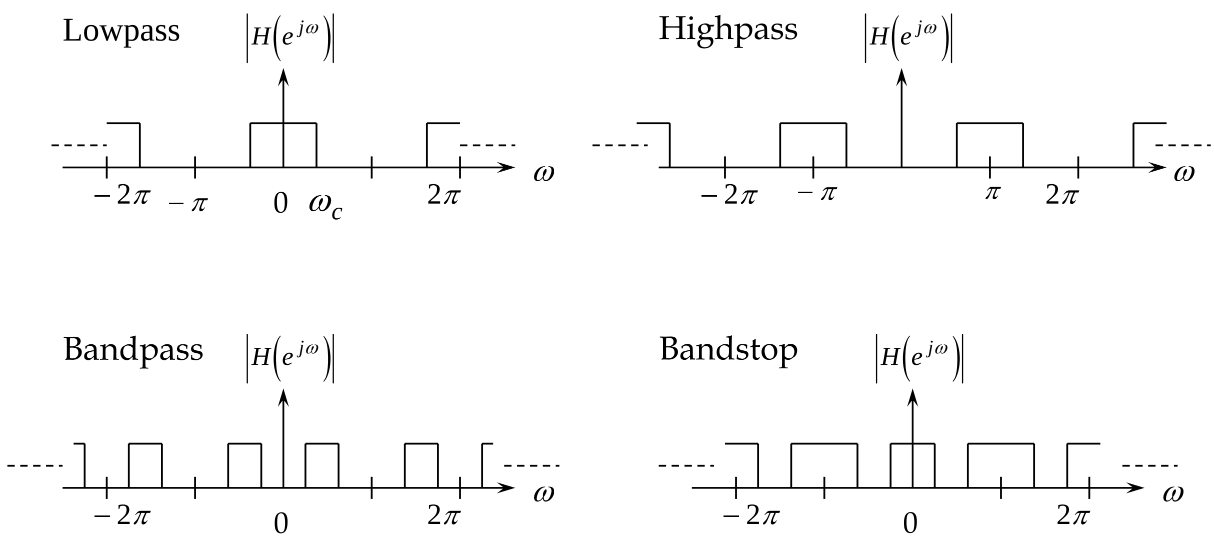
- analysis can show some structures are more robust than others
  - FIR filter implementations normally use
    - direct form
    - cascade of 2nd order sections
  - IIR filter implementations normally use
    - cascade form
    - parallel form
    - NOT direct form

- Rounding of arithmetic operations
  - may cause <u>limit cycles</u>
- Example
  - filter's system function $$H(z) = \frac{1}{1+0.9z^{-1}}$$
  - input signal $x(n) = 10\delta(n)$
  - ideal response $y(n) = 10(-0.9)^n$
  - actual response for integer rounding of arithmetic

$$y(n) = x(n) - 0.9y(n-1)$$
$$= \{10,-9,8,-7,6,-5,5,-5,5,-5, \text{ etc } \}$$

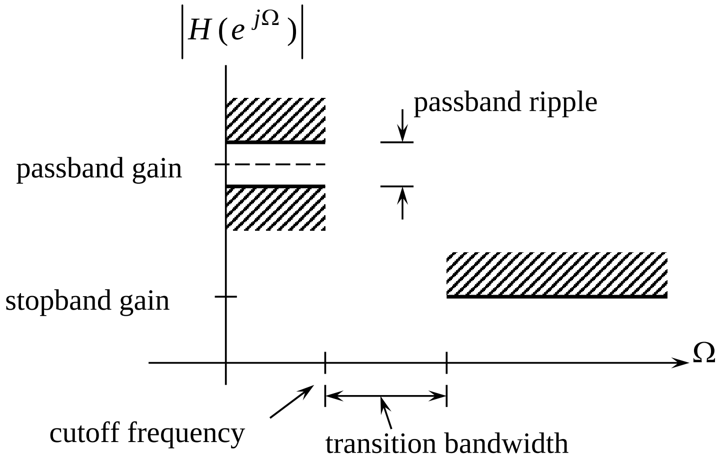
> This is called a limit cycle.
> The pole has effectively moved from $z = -0.9$ to $z = -1$.
> The result is oscillation at $f_s/2$.

Ideal Digital Filters



Filter Specification

- Aim of filter design
  - Given some frequency response specification, determine the type, order and coefficients of a digital filter to best meet the specification.



FIR Filter Design

- Order determination
  - order is
    - proportional to stopband attenuation
    - inversely proportional to transition bandwidth
    - often determined by approximate rule

- Coefficients of an FIR filter are also the impulse response

$$Z\left\{\{h(0),h(1),\ldots,h(N-1)\}\right\} = \sum_{n=0}^{N-1} h(n)z^{-n}$$

  - if we know the impulse response of the required filter then we can easily design the filter

“Ideal” FIR filters

- In general, an ideal (continuous) frequency response is related to an (infinite) impulse response by the Fourier Series

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d(n) e^{-jn\omega}$$

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{jn\omega} d\omega$$

- The coefficients of an “ideal” FIR filter can therefore be found from the Fourier Series coefficients of the desired frequency response.

- Not practical because
  - the impulse response cannot be infinite
  - the impulse response must be causal
  - maybe don’t need the frequency response to be specified for all (continuous) values of ω

Frequency Sampling

- truncation of the impulse response introduces errors
  - truncation of the impulse response is equivalent to sampling of the frequency response
- the truncated impulse response can be obtained directly from the DFT of the desired frequency response

$$H(k) = H_d(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N}k} = \sum_{n=0}^{N-1} h(n) e^{-j\frac{2\pi}{N}nk} \qquad k = 0,1,2,\ldots,N-1$$

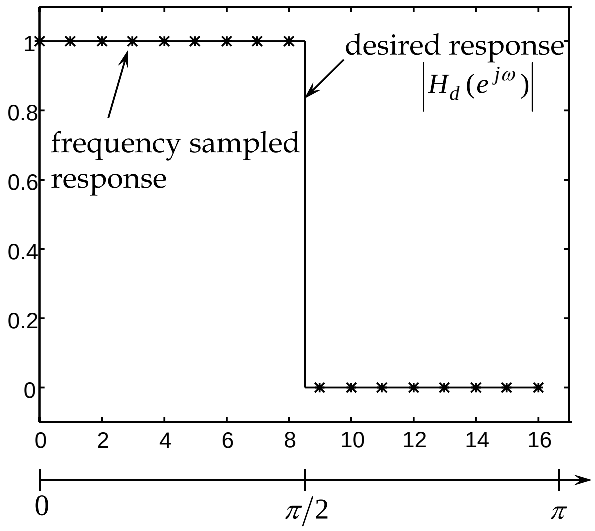
$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k) e^{j\frac{2\pi}{N}nk} \qquad n = 0,1,2,\ldots,N-1$$

- N-1 is the order of the FIR filter
- The frequency response has been sampled at N points around the unit circle
  - The frequency response of filter designed in this way will only be exactly correct only at these points

◆ Example

- Lowpass filter
- Number of taps: 33

$$\left|H(e^{j\omega})\right| = \begin{cases} 1, & |\omega| < \pi/2 \\ 0, & \text{otherwise} \end{cases}$$

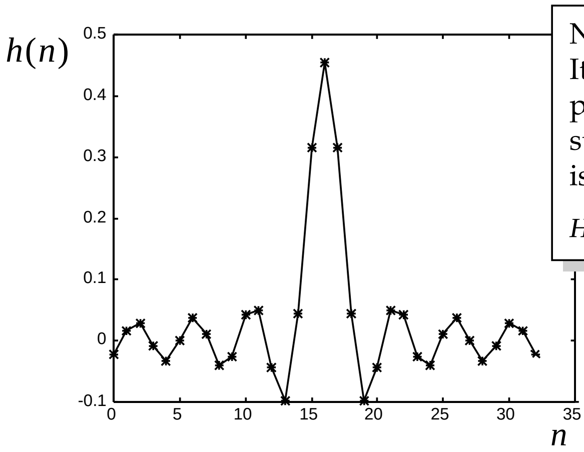


- For the ideal filter (from Fourier Series)

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi/2}^{\pi/2} e^{jn\omega}\,d\omega = \tfrac{1}{2}\,\frac{\sin(n\pi/2)}{n\pi/2} \qquad n = -\infty,\ldots,\infty$$

- For the truncated filter (from IDFT)

$$h(n) = \frac{1}{33}\sum_{k=0}^{32} H(k)e^{j\frac{2\pi}{N}nk}, \; n = 0,1,2,\ldots,32$$



Note that this result is causal. It is obtained using a linear phase assumption for the filter such that the delay of the filter is given by $(N-1)/2$ such that $H(e^{j\omega}) = \left|H(e^{j\omega})\right| \cdot e^{-j\omega(N-1)/2}$

Windowing

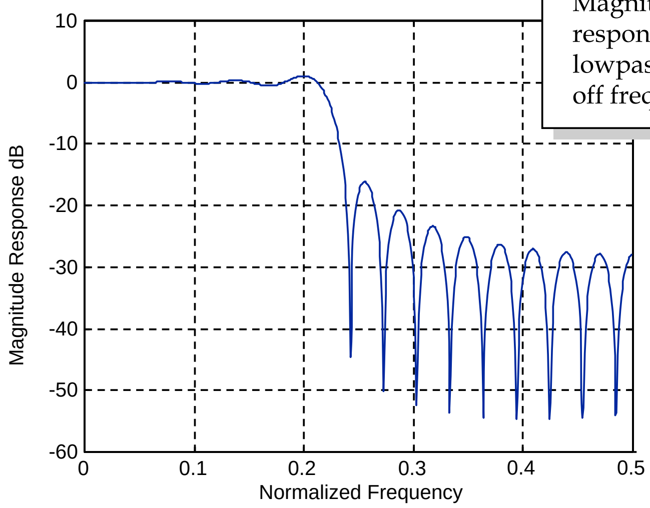
- The truncation of the impulse response is equivalent to multiplication of the ideal (infinite) impulse response by a square window $w(n)$

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$
$$= h_d(n)w(n)$$

  - Square window function

$$w(n) = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

- Effect of multiplying impulse response by window
  - convolution of ideal frequency response with Fourier transform of window
  - Fourier transform of square window is sinc
    - expect to see high side-lobes and ripples in the frequency response of the filter designed using square window



Magnitude of frequency response for the 33 taps lowpass filter with cut-off frequency of $\pi/2$

◆ Other window functions

- Hamming window

$$w(n) = \begin{cases} 0.54 + 0.46\cos\left(\dfrac{n\pi}{I}\right), & -I \le n \le I \\ 0, & \text{otherwise} \end{cases}$$

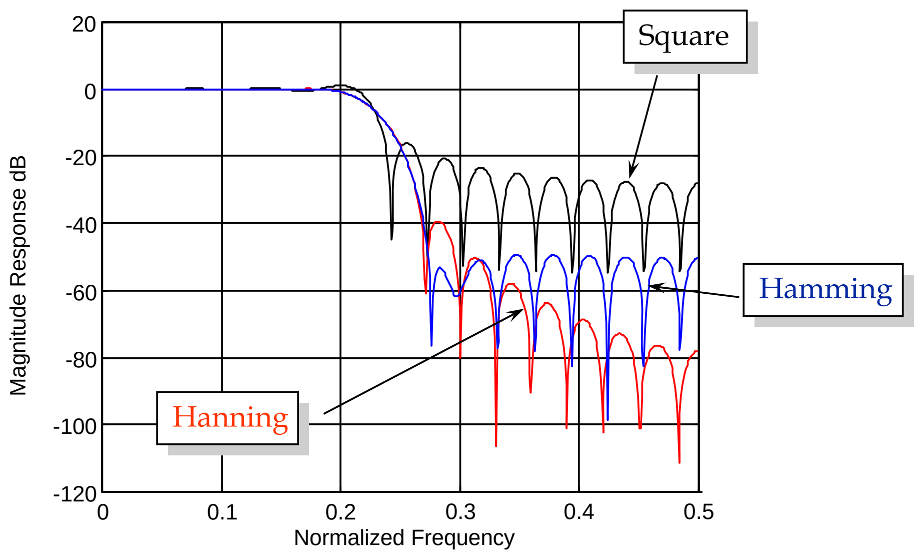
- Hanning window

$$w(n) = \begin{cases} 0.5 + 0.5\cos\left(\dfrac{n\pi}{I}\right), & -I \le n \le I \\ 0, & \text{otherwise} \end{cases}$$

- (Several others)

- Use of raised cosine-type windows (Hamming or Hanning) gives better stopband attenuation but wider transition band

◆ Filter magnitude responses for square, Hamming and Hanning windows



IIR Filter Design

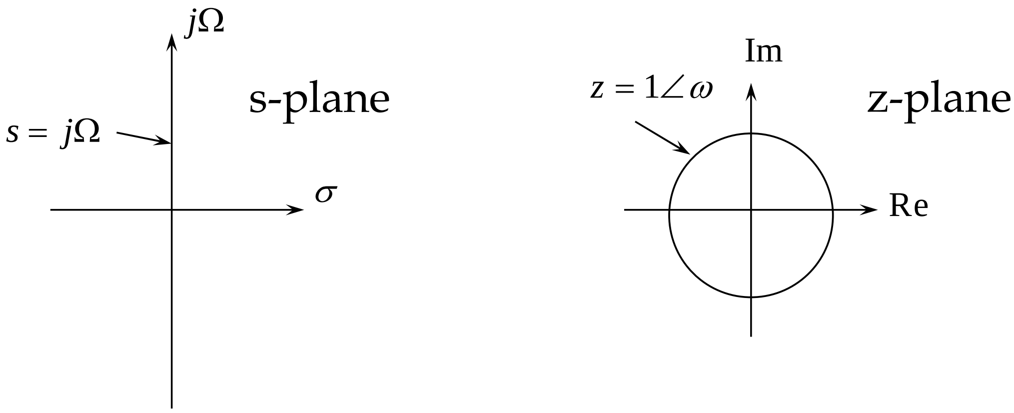
- Normally done by transforming a continuous-time design to discrete-time
  - many "classical" continuous-time filters are known and coefficients tabulated
    - Butterworth
    - Chebyshev
    - Elliptic, etc.
  - possible transformations are
    - Impulse invariant transformation
    - Bilinear transformation

- Butterworth Filters
  - maximally flat in both passband and stopband
    - first $2N-1$ derivatives of $|H(j\Omega)|^2$ are zero at $\Omega = 0$ and $\Omega = \infty$

$$\begin{aligned} |H(j\Omega)|^2 &= \frac{1}{1+\left(\dfrac{\Omega}{\Omega_c}\right)^{2N}} \\ &= 1, \qquad \text{for } \Omega = 0 \\ &= 1/\sqrt{2}, \quad \text{for } \Omega = \Omega_c \\ &\propto 1/\Omega^N, \quad \text{for } \Omega >> \Omega_c \end{aligned}$$

  - $2N$ poles of $H(s)H(-s)$ equally spaced around a circle in the s-plane of radius Ω_c symmetrically located with respect to both real and imaginary axes
    - poles of $H(s)$ selected to be the N poles on the left half plane of s
  - coefficients of a continuous-time filter for specific order and cutoff frequency can be found
    - from analysis of above expression
    - from tables

- Transformation to discrete-time
  - wish to map $j\omega$ axis of the s-plane to the unit circle of the z-plane
  - wish to map poles and zeros on the left half plane of s-plane to the inside of unit circle in z



Impulse-Invariant Transformation

- Not widely used
- Impulse response of discrete-time filter is obtained by sampling the impulse response of continuous-time filter
  - impulse response is preserved by the mapping
  - frequency response is not preserved due to aliasing

Bilinear Transformation

- Widely used
- Avoids aliasing problem of impulse-invariant transformation

- Two elements
  - transformation $s = \dfrac{z-1}{z+1}$
  - frequency warping $\Omega = \tan(\omega/2)$

- From $s = \dfrac{z-1}{z+1}$ we can write $z = \dfrac{1+s}{1-s} = \dfrac{1+\sigma+j\Omega}{1-\sigma-j\Omega} = re^{j\theta}$

  $\sigma > 0 \Rightarrow r > 1$
  - right half plane of s maps to exterior of unit circle in z

  $\sigma < 0 \Rightarrow r < 1$
  - left half plane of s maps to interior of unit circle in z

  $\sigma = 0 \Rightarrow r = 1$
  - imaginary axis in s maps to unit circle in z

  - On the imaginary axis we have (using $z = e^{j\omega}$)

$$j\Omega = \frac{e^{j\omega}-1}{e^{j\omega}+1} = \frac{j\sin(\omega/2)}{\cos(\omega/2)} = j\tan(\omega/2)$$

  - this is the relationship between frequency in the continuous-time filter and frequency in the discrete-time filter
  - it is a non-linear relationship which is close to linear for small frequencies

Example

- Design a 2nd order Butterworth digital filter with cutoff frequency of 2 kHz and sampling frequency 20 kHz.

- Pre-warp the discrete-time frequencies to obtain the equivalent continuous-time frequencies

$$\Omega_c = \tan(\omega_c/2) = \tan(0.1 \times 2\pi/2) = 0.325$$

- Design a continuous-time Butterworth filter for cutoff at Ω_c
  - For cutoff at 1 rad/s $$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$
  - For cutoff at 0.325 rad/s $$s \to \frac{s}{0.325} \qquad \therefore H(s) = \frac{0.106}{s^2 + 0.46s + 0.106}$$

- Apply transformation $$s = \frac{z-1}{z+1} \qquad \therefore H(z) = \frac{0.068z^2 + 0.136z + 0.068}{z^2 - 1.142z + 0.413}$$