Filter Design

Jose Krause Perin

Stanford University

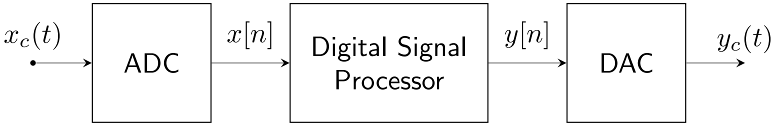
July 30, 2018

Last lecture

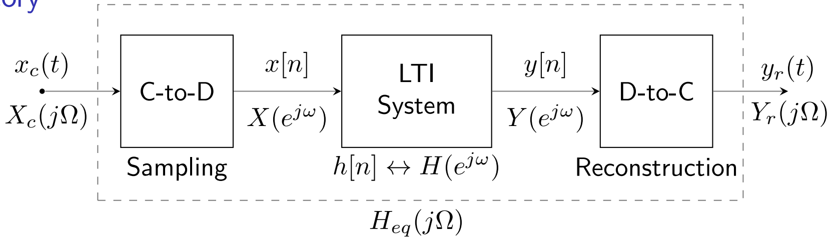
- Two's complement is a fixed-point representation that represents fractions as integers
- There's an inherent trade-off between roundoff noise and overflow/clipping
- FIR systems remain stable after coefficient quantization
- Linear phase FIR systems remain linear phase after coefficient quantization, since the impulse response remains symmetric
- Coefficient quantization may lead to instability in IIR systems, as poles may move outside the unit circle
- Similarly to quantization noise, roundoff noise is modeled by an additive uniformly distributed white noise that is independent of the input signal (the linear noise model).
- Roundoff noise is minimized by performing quantization only after accumulation, but this requires $(2B+1)$-bit adders
- In FIR structures the equivalent roundoff noise at the output is white
- IIR structures lead to roundoff noise shaping
- The least noisy IIR structure depends on the system
- Cascade and parallel forms are used to mitigate total roundoff noise

Practice and theory

In practice



DSP theory



Digital filter design

We'll cover two different design problems

1. Digital filter design from analog filter
   Given a continuous-time LTI filter defined by $h_{eq}(t) \iff H_{eq}(s)$, how to obtain the corresponding discrete-time filter $h[n] \iff H(z)$ such that

$$H(e^{j\Omega T}) \approx H_{eq}(j\Omega), |\Omega| < \Omega_s/2$$

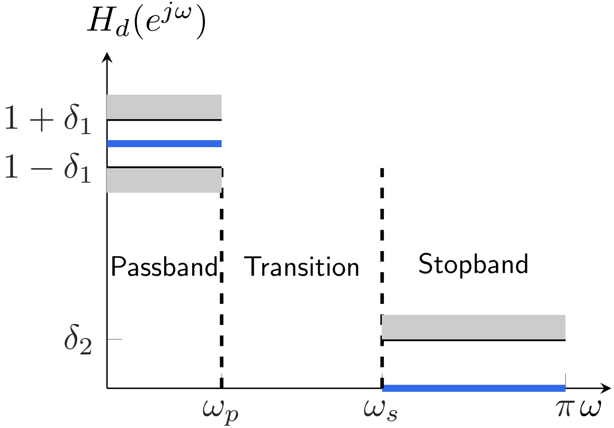
Design techniques:

- Impulse invariance
- Bilinear transformation

Design by impulse invariance can result in either FIR or IIR filters, whereas bilinear transformation generally results in IIR filters.

Digital filter design

2. Digital FIR filter design from specifications
How to find FIR $H(z)$ such that $H(e^{j\omega})$ best approximates a desired frequency response $H_d(e^{j\omega})$? Essentially a polynomial curve fitting problem.



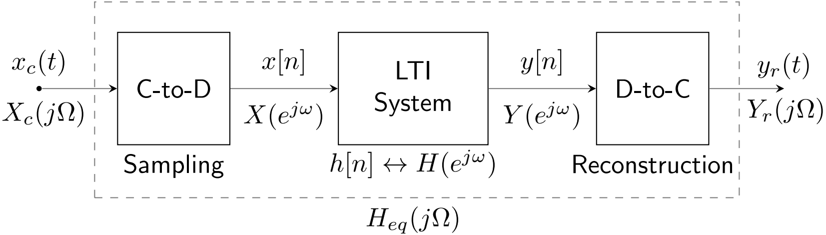
Design techniques:

- Window method
- Optimal filter design
  - Parks-McClellan algorithm
  - Least-squares algorithm

Outline

Outline

Digital processing of analog signals



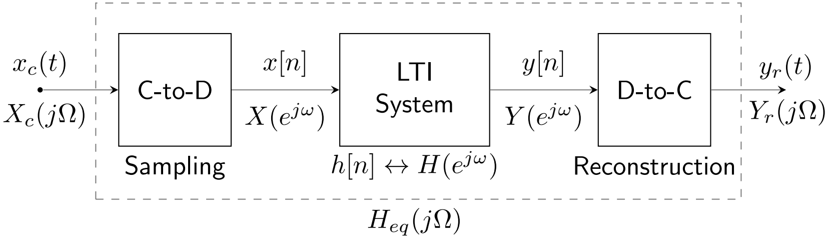
As long as there is no aliasing and that the reconstruction filter is the ideal lowpass filter these equalities hold:

$$H_{eq}(j\Omega) = \begin{cases} H(e^{j\Omega T}), & |\Omega| < \pi/T \\ 0, & |\Omega| > \pi/T \end{cases} \qquad \text{(from DSP to analog)}$$

$$H(e^{j\omega}) = H_{eq}(j\omega/T), \quad |\omega| < \pi \qquad \text{(from analog to DSP)}$$

In practice, these are good approximations.

Question: How to design $h[n] \longleftrightarrow H(z)$ if we know $h_{eq}(t) \longleftrightarrow H_{eq}(s)$?



Design $h[n]$ by sampling $h_{eq}(t)$ with period T.

$$h[n] = T h_c(nT) \qquad \text{(impulse invariance)}$$

The scaling factor T compensates for the $1/T$ attenuation in the frequency domain due to sampling

The resulting $h[n]$ depends on the sampling period T.

Impulse invariance example: lowpass Butterworth filter

Butterworth filters are **maximally flat** in the passband and are monotonic overall. The downside of Butterworth filters is their relatively slow roll-off.

For this example, consider the following 6th-order continuous-time lowpass Butterworth filter:

$$H_{eq}(s) = \frac{0.12093}{(s^2 + 0.364s + 0.4945)(s^2 + 0.9945s + 0.4945)(s^2 + 1.3385 + 0.4945)}$$

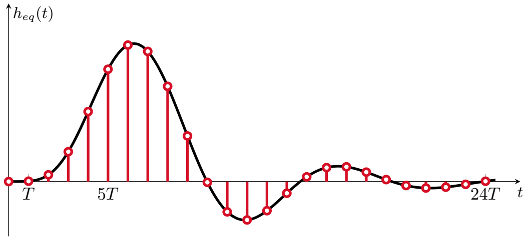
Impulse invariance example: lowpass Butterworth filter

To design an **FIR filter** by impulse invariance we must

1. Obtain the continuous-time impulse response $h_{eq}(t) \longleftrightarrow H_{eq}(s)$ (`impulse` in Matlab)
2. Sample and scale $h_{eq}(t)$ with period T and record only $M+1$ first samples

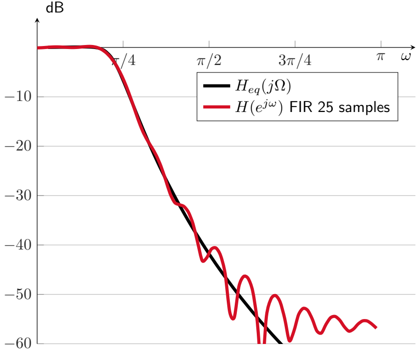
$$h[n] = \begin{cases} T h_{eq}(nT), & n = 0, \dots, M \\ 0, & \text{otherwise} \end{cases}, \qquad \text{(for causal } h_{eq}(t)\text{)}$$

$h[n]$ is the FIR filter coefficients. M is typically chosen to satisfy some energy criterion. For instance, samples must contain 95% of the signal energy.

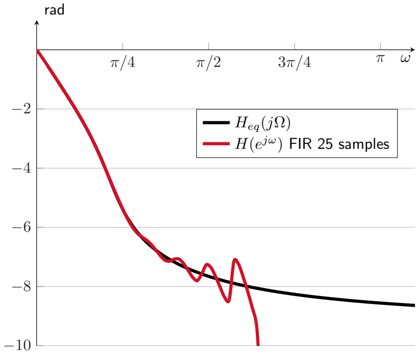


Impulse invariance example: lowpass Butterworth filter

Magnitude



Phase



Questions:

1. What would happen if we take fewer samples (smaller M)?
2. What would happen if we decrease the sampling period e.g., $T_2 = 0.5T$?

Impulse invariance example: lowpass Butterworth filter

- Designing FIR filters by impulse invariance is straightforward. Plus, FIR systems have the implementation advantages discussed in lectures 7 and 8
- **Problem:** it may require prohibitively many samples to achieve good accuracy
- IIR systems generally offer better accuracy while requiring fewer operations (coefficients)

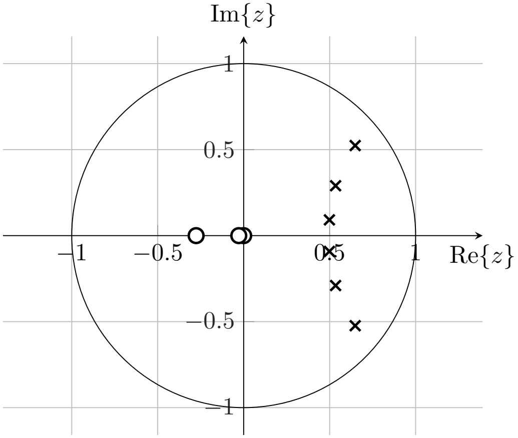
To design an **IIR filter** by impulse invariance we must

1. Invert the Laplace transform $H_{eq}(s)$ using **partial fraction expansion** to obtain $h_{eq}(t)$ analytically. Function `residue` in Matlab
2. Sample $h_{eq}(t)$: $h[n] = Th_{eq}(nT)$
3. Calculate the z-transform $H(z)$ of $h[n]$

Impulse invariance example: lowpass Butterworth filter

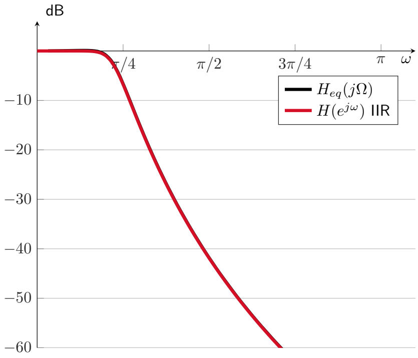
For the 6th-order Butterworth example:

$$H(z) = \frac{0.2871 - 0.4466z^{-1}}{1 - 1.2971z^{-1} + 0.6949z^{-2}} + \frac{-2.1428 + 1.1455z^{-1}}{1 - 1.0691z^{-1} + 0.3699z^{-2}} + \frac{1.8557 - 0.6303z^{-1}}{1 - 0.9972^{-1} + 0.2570z^{-2}}$$

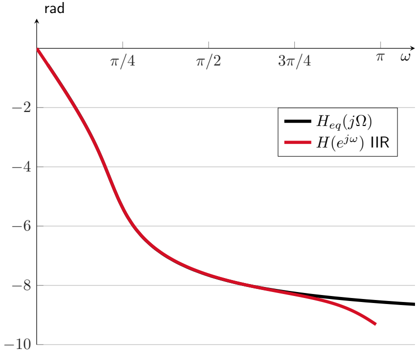


Impulse invariance example: lowpass Butterworth filter

Magnitude



Phase

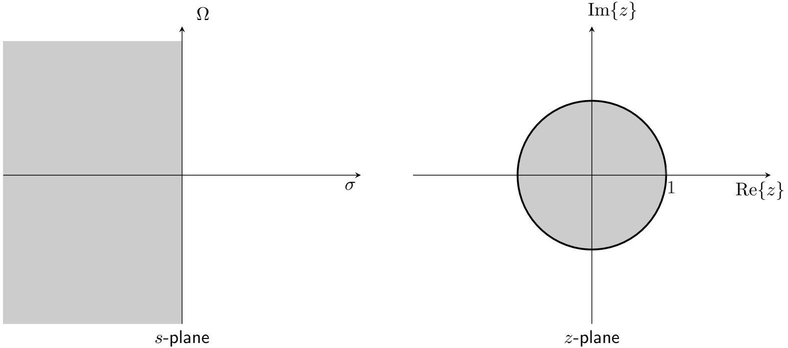


- IIR systems achieve better accuracy while requiring fewer operations (coefficients) than FIR systems.
- Similarly to FIR systems, if we change the sampling frequency the behavior of the filter changes.

Bilinear transformation

Another way to answer the question: How to design $h[n] \longleftrightarrow H(z)$ given $h_{eq}(t) \longleftrightarrow H_{eq}(s)$?
The **bilinear transformation** maps the left-hand side of the s-plane into the unit circle in the z-plane.

$$s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \qquad \text{(Bilinear transformation)}$$



Bilinear transformation

To design a digital filter from an analog filter using the bilinear transformation, we simply make the following change of variables:

$$H(z) = H_{eq}(s)\Big|_{s = \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}}}$$

The resulting $H(z)$ generally is IIR.

The bilinear transformation method is easier and more systematic than the impulse invariance method.

In Matlab: `[bz, az] = bilinear(bs, as, 1/T)`

Frequency warping

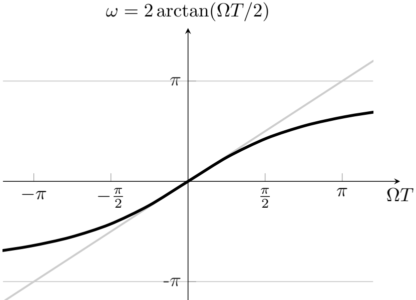
Evaluating z on the unit circle is equivalent to evaluating s on the imaginary axis $j\Omega$:

$$j\Omega = \frac{2}{T}\left(\frac{1-e^{-j\omega}}{1+e^{-j\omega}}\right) = j\frac{2}{T}\tan\omega/2$$

This results in the following relation

$$\omega = 2\arctan(\Omega T/2) \qquad \text{(frequency warping)}$$

Problem: with the bilinear transformation we no longer have the linear relation $\omega = \Omega T$. This is known as **frequency warping**.

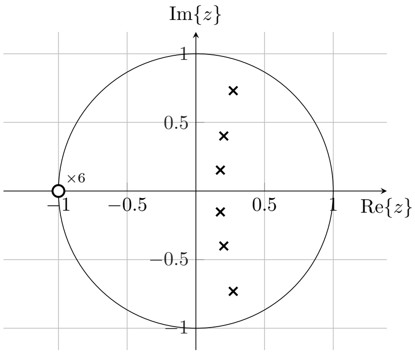


Bilinear transformation example: lowpass Butterworth filter

Revisiting the example of the 6th-order lowpass Butterworth filter
To obtain $H(z)$ we simply make:

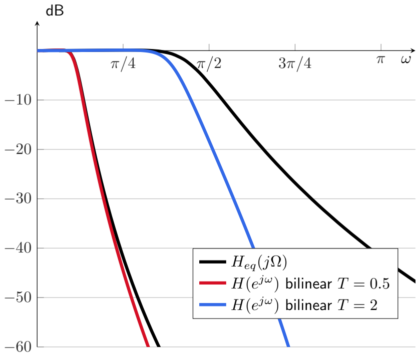
$$H(z) = H_{eq}(s)\Big|_{s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

Pole-zero diagram

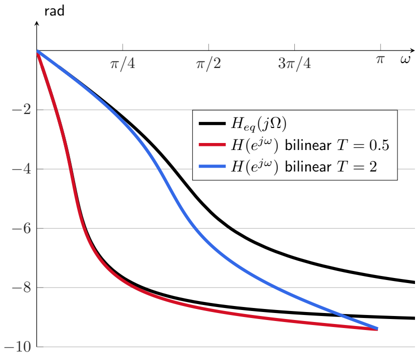


Bilinear transformation example: lowpass Butterworth filter

Magnitude



Phase



- Similarly to impulse invariance, the resulting frequency response depends on the sampling period T.
- Frequency warping leads to the disagreement between continuous-time and discrete-time filters for $\omega > 0.3\pi$

Frequency pre-warping

Frequency pre-warping mitigates the distortion caused by frequency warping by **scaling** s so that $H(e^{j\Omega_p T}) = H_{eq}(j\Omega_p)$ (no distortion) at some specified frequency Ω_p.

$$H(z) = H_{eq}(s)\Big|_{s = \frac{\Omega_p}{\tan(\Omega_p T/2)} \frac{1-z^{-1}}{1+z^{-1}}}$$

(bilinear transformation with frequency pre-warping)

Ω_p is chosen so that $H(e^{j\omega})$ will preserve a particular characteristic of $H_{eq}(j\Omega)$ e.g., Ω_p is made equal to the 3-dB bandwidth.

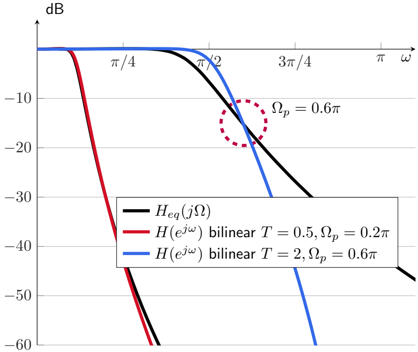
In Matlab: `[bz, az] = bilinear(bs, as, 1/T, Wp/(2*pi))`

Bilinear transformation example: lowpass Butterworth filter

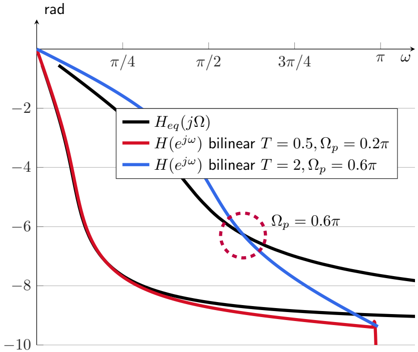
Example of bilinear transformation with frequency pre-warping

- $\Omega_p = 0.6\pi$ for $T = 2$
- $\Omega_p = 0.2\pi$ for $T = 0.5$.

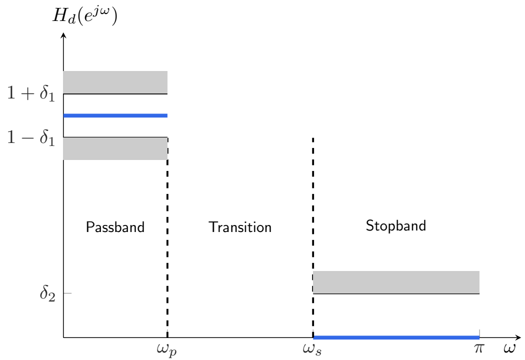
Magnitude



Phase



Common terminology



Terminology

- The filter order is equal to the largest power of z^{-1} or z
- δ_1 passband ripple
- δ_2 stopband ripple (stopband attenuation)
- ω_p passband edge frequency
- ω_s stopband edge frequency

Classic filters

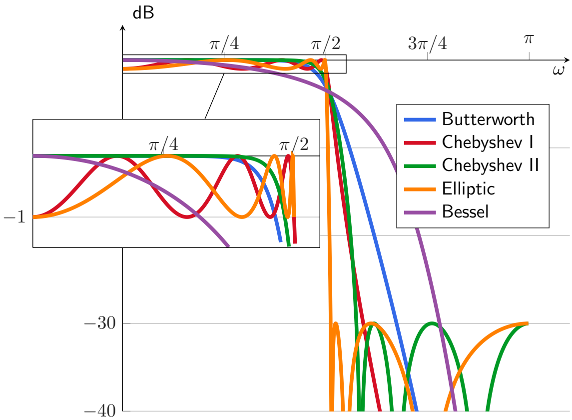
- **Butterworth:** It's monotonic in the passband and in the stopband.
  Matlab: `butter(order, w3dB/pi)`
- **Chebyshev type I:** It has equiripple frequency response in the passband and varies monotonically in stopband.
  Matlab: `cheby1(order, passband_ripple, wp/pi)`
- **Chebyshev type II:** It has equiripple frequency response in the stopband and varies monotonically in the passband.
  Matlab: `cheby2(order, stopband_attenuation, ws/pi)`
- **Elliptic:** It has equiripple frequency response in both the passband and the stopband.
  Matlab: `ellip(order, passband_ripple, stopband_attenuation, wp/pi)`
- **Bessel:** It has maximally linear phase response (constant group delay).
  Matlab function `besself` (only for continuous time)

In general (and in Matlab) these filters are first designed in continuous-time $H(s)$, and then converted to discrete-time $H(z)$ using the bilinear transformation with frequency pre-warping.

Comparison of classic filters

- All are 6th-order filters designed to have 3-dB bandwidth of $\approx \pi/2$.
- Ripple was set to 1 dB in passband
- Stopband attenuation was 30 dB.

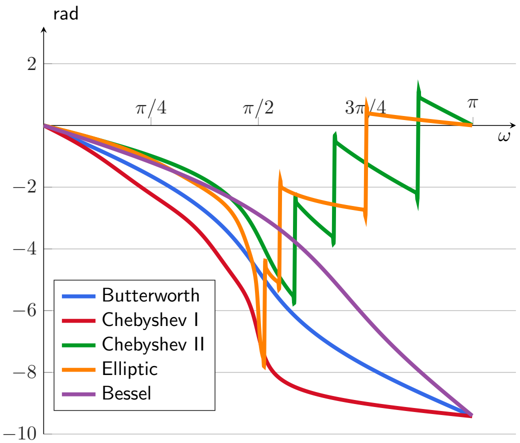
Magnitude



Comparison of classic filters

- All are 6th-order filters designed to have 3-dB bandwidth of $\approx \pi/2$.
- Ripple was set to 1dB in passband and stopband
- Stopband attenuation was 30 dB.

Phase



From lowpass to highpass, bandpass, and bandstop

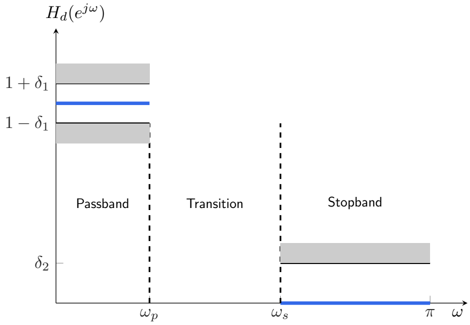
TABLE 7.1 TRANSFORMATIONS FROM A LOWPASS DIGITAL FILTER PROTOTYPE OF CUTOFF FREQUENCY θ_p TO HIGHPASS, BANDPASS, AND BANDSTOP FILTERS

Filter Type	Transformations	Associated Design Formulas
Lowpass	$Z^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$	$\alpha = \frac{\sin\left(\frac{\theta_p - \omega_p}{2}\right)}{\sin\left(\frac{\theta_p + \omega_p}{2}\right)}$ ω_p = desired cutoff frequency
Highpass	$Z^{-1} = -\frac{z^{-1} + \alpha}{1 + \alpha z^{-1}}$	$\alpha = -\frac{\cos\left(\frac{\theta_p + \omega_p}{2}\right)}{\cos\left(\frac{\theta_p - \omega_p}{2}\right)}$ ω_p = desired cutoff frequency
Bandpass	$Z^{-1} = -\frac{z^{-2} - \frac{2\alpha k}{k+1} z^{-1} + \frac{k-1}{k+1}}{\frac{k-1}{k+1} z^{-2} - \frac{2\alpha k}{k+1} z^{-1} + 1}$	$\alpha = \frac{\cos\left(\frac{\omega_{p2} + \omega_{p1}}{2}\right)}{\cos\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right)}$ $k = \cot\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right) \tan\left(\frac{\theta_p}{2}\right)$ ω_{p1} = desired lower cutoff frequency ω_{p2} = desired upper cutoff frequency
Bandstop	$Z^{-1} = \frac{z^{-2} - \frac{2\alpha}{1+k} z^{-1} + \frac{1-k}{1+k}}{\frac{1-k}{1+k} z^{-2} - \frac{2\alpha}{1+k} z^{-1} + 1}$	$\alpha = \frac{\cos\left(\frac{\omega_{p2} + \omega_{p1}}{2}\right)}{\cos\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right)}$ $k = \tan\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right) \tan\left(\frac{\theta_p}{2}\right)$ ω_{p1} = desired lower cutoff frequency ω_{p2} = desired upper cutoff frequency

Outline

Digital FIR filter design from specifications

How to find FIR $H(z)$ such that $H(e^{j\omega})$ best approximates a desired frequency response $H_d(e^{j\omega})$? Essentially a polynomial curve fitting problem.



Design techniques:

- Window method
- Optimal filter design
  - Parks-McClellan algorithm
  - Least squares

Window method

An easy way to design an FIR filter to match a desired frequency response $H_d(e^{j\omega})$ is to calculate the inverse DTFT of $H_d(e^{j\omega})$ and truncate the result to a reasonable number of samples (similar to impulse invariance):

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega \qquad \text{(inverse DTFT)}$$

Then we truncate it to have at most $M+1$ samples

$$h[n] = \begin{cases} h_d[n], & n = 0, 1, \ldots, M \\ 0, & \text{otherwise} \end{cases} \qquad \text{(truncated sequence)}$$

Another way to write truncation is

$$h[n] = w[n]h_d[n], \quad \text{where } w[n] = \begin{cases} 1, & n = 0, 1, \ldots, M \\ 0, & \text{otherwise} \end{cases} \qquad \text{(truncated sequence)}$$

$w[n]$ is the **window sequence**, which in this case is the rectangular window.

Window method

Representing truncation as $h[n] = w[n]h_d[n]$, gives us an easy way to understand what happens in the frequency domain.

Multiplication in time domain means convolution in the frequency domain:

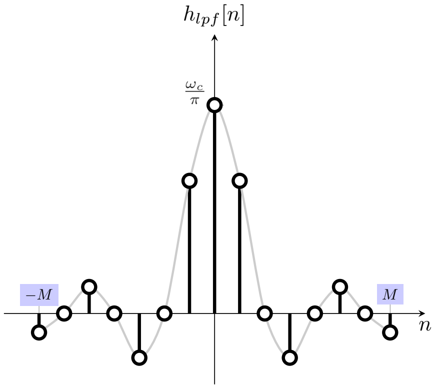
$$\begin{aligned} H(e^{j\omega}) &= \frac{1}{2\pi} W(e^{j\omega}) * H_d(e^{j\omega}) \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta \qquad \text{(convolution)} \end{aligned}$$

Problem: $H(e^{j\omega})$ will not be equal to $H_d(e^{j\omega})$. Instead, it will be a *smeared* version of the desired response $H_d(e^{j\omega})$.

Revisiting the Gibbs phenomenon

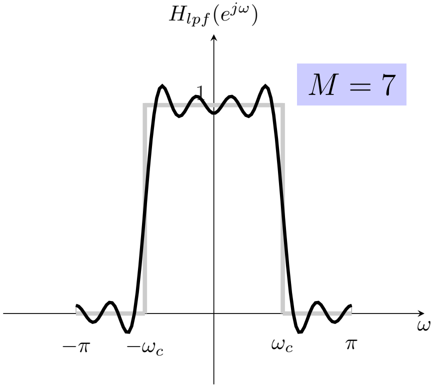
Time domain

$$h_{lpf}[n] = \frac{\sin \omega_c n}{\pi n} = \frac{\omega_c}{\pi}\text{sinc}\left(\frac{\omega_c}{\pi} n\right)$$



Frequency domain

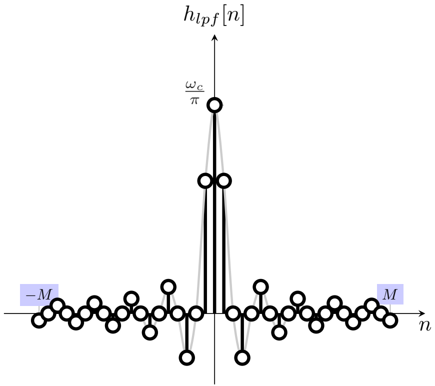
$$H_M(e^{j\omega}) = \sum_{n=-M}^{M} \frac{\sin \omega_c n}{\pi n} e^{-j\omega n}$$



Revisiting the Gibbs phenomenon

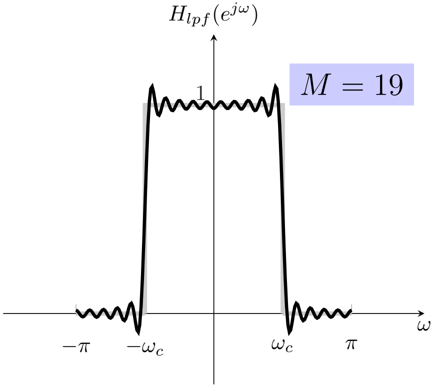
Time domain

$$h_{lpf}[n] = \frac{\sin \omega_c n}{\pi n} = \frac{\omega_c}{\pi} \text{sinc}\left(\frac{\omega_c}{\pi} n\right)$$



Frequency domain

$$H_M(e^{j\omega}) = \sum_{n=-M}^{M} \frac{\sin \omega_c n}{\pi n} e^{-j\omega n}$$



Revisiting the Gibbs phenomenon

- The Gibbs phenomenon appears when we truncate the impulse response of the ideal lowpass filter (or any discontinuous DTFT).
- In lecture 1, we attributed this to convergence issues of the DTFT for non-absolute summable sequences. The DTFT of the sinc converges only in the mean square sense, and not uniformly
- Another way to view the Gibbs phenomenon is as a result of windowing.

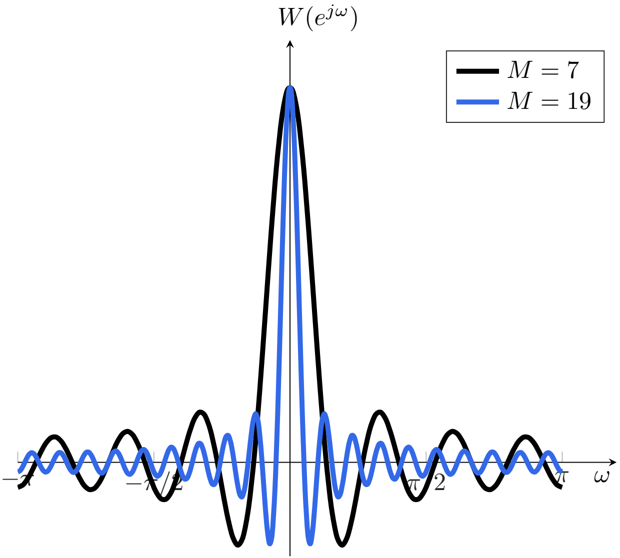
$$H(e^{j\omega}) = \frac{1}{2\pi} W(e^{j\omega}) * H_d(e^{j\omega}) \qquad \text{(convolution)}$$

- In this case the desired response $H_d(e^{j\omega})$ is the ideal lowpass filter, and the window function is

$$w[n] = \begin{cases} 1, & n = -M, -M+1, \ldots, M-1, M \\ 0, & \text{otherwise} \end{cases}$$

$$\iff W(e^{j\omega}) = \frac{\sin(\omega(2M+1)/2)}{\sin(\omega/2)}$$

Rectangular window



Rectangular window

From Fourier transform theory, we can show that the rectangular window produces $H(e^{j\omega})$ that best matches $H_d(e^{j\omega})$ in the mean-square sense. That is,

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega, \qquad \text{(mean-square error)}$$

is minimized when $w[n]$ is the rectangular window.
Question: are there other windows $w[n]$ that minimize issues with discontinuities without excessively increasing the mean-square error?

Commonly used windows

Rectangular:

$$w[n] = \begin{cases} 1, & 0 \leq n \leq M \\ 0, & \text{otherwise} \end{cases}$$

Bartlett (triangular):

$$w[n] = \begin{cases} 2n/M, & 0 \leq n \leq M/2, M \text{ even} \\ 2 - 2n/M, & M/2 < n \leq M \\ 0, & \text{otherwise} \end{cases}$$

Hann:

$$w[n] = \begin{cases} 0.5 - 0.5\cos(2\pi n/M), & 0 \leq n \leq M, \\ 0, & \text{otherwise} \end{cases}$$

Hamming:

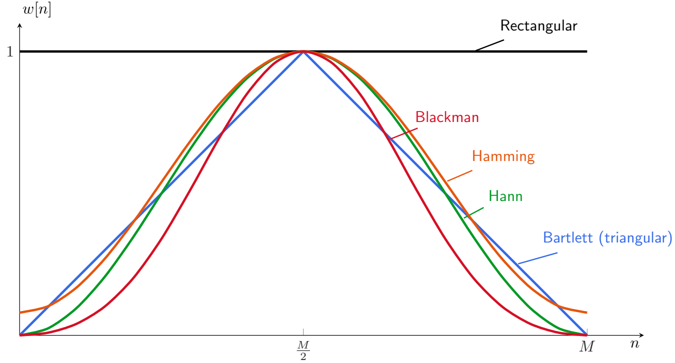
$$w[n] = \begin{cases} 0.54 - 0.46\cos(2\pi n/M), & 0 \leq n \leq M, \\ 0, & \text{otherwise} \end{cases}$$

Blackman:

$$w[n] = \begin{cases} 0.42 - 0.5\cos(2\pi n/M) + 0.08\cos(4\pi n/M), & 0 \leq n \leq M, \\ 0, & \text{otherwise} \end{cases}$$

Commonly used windows

Time domain
All windows are symmetric about $M/2$.



Note: n is discrete. These curves were plotted as continuous functions just for easier visualization.
We will revisit windows when talking about spectrum analysis (lecture 12)

Linear phase in filters designed by windowing

If the window is causal and symmetric and if the desired impulse response $h_d[n]$ is causal and symmetric, then it follows

$$w[n] = \pm w[M-n] \qquad \text{(causal and symmetric window)}$$
$$h_d[n] = \pm h_d[M-n] \qquad \text{(causal and symmetric } h_d[n]\text{)}$$
$$h[n] = w[n]h_d[n] = \pm w[M-n]h_d[M-n] = \pm h[M-n] \qquad \text{(causal and symmetric } h[n]\text{)}$$

Therefore, $h[n]$ is either even or odd symmetric and consequently $H(e^{j\omega})$ has generalized linear phase.

Kaiser window

It's typically desired that the window be maximally concentrated around $\omega = 0$ (small sidelobe area).
The **Kaiser window** offers a nearly optimal trade-off between main-lobe width and side-lobe area.

$$w[n] = \begin{cases} \dfrac{I_0\left(\beta\sqrt{1-(n-\alpha)^2/\alpha^2}\right)}{I_0(\beta)}, & 0 \le n \le L-1 \\ 0, & \text{otherwise} \end{cases},$$

where $\alpha = (L-1)/2$, β is a design parameter, and $I_0(\cdot)$ is the **modified Bessel function of first kind and order 0**.
See section 7.5.3 of the textbook for recommendations on values of β for lowpass filter design.

Summary on FIR filter design by the window method

1. From the desired frequency response $H_d(e^{j\omega})$ calculate the desired impulse response $h_d[n]$.
2. Choose the filter order M and the window $w[n]$. Then,

$$h[n] = \begin{cases} h_d[n]w[n], & n = 0, \ldots, M \\ 0, & \text{otherwise} \end{cases} \qquad \text{(for } h_d[n] \text{ causal)}$$

   Kaiser window depends on parameters β and M. Other windows only depend on M.
3. Linear phase is guaranteed if $h_d[n]$ and $w[n]$ are symmetric

In Matlab:
`fir1` uses Hamming window by default. Other windows can be passed as parameters:

```
>> fir1(M, wc/pi, 'lowpass', kaiser(M+1, beta))
```

designs a lowpass FIR filter of order M and cutoff frequency ω_c using the window method with Kaiser window with parameter β

Optimal FIR filter design

- Though straightforward, filter design by windowing is sub-optimal in the sense that it compromises accuracy for better *handling* of discontinuities in $H_d(e^{j\omega})$.
- More importantly, there was no well-defined metric to evaluate filters

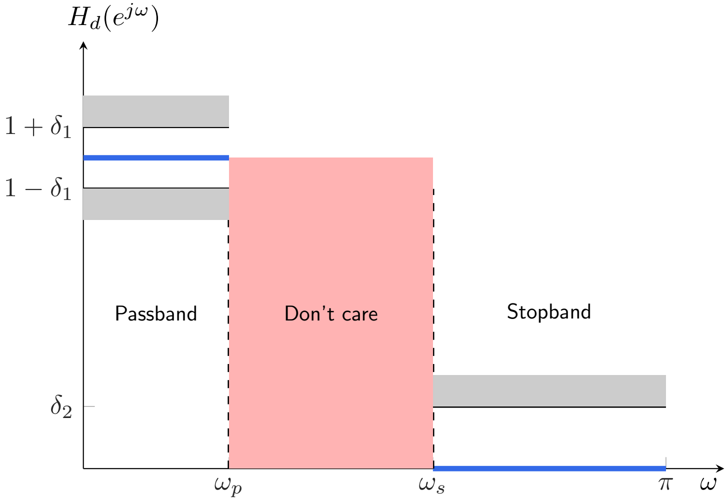
A sensible choice for evaluation metric is the **weighted error**:

$$E(\omega) = W(\omega)\Big(H_d(e^{j\omega}) - H(e^{j\omega})\Big), \qquad \text{(weighted error)}$$

where $0 \leq W(\omega) \leq 1$ is the **weight function**.

- Generally, we choose either $W(\omega) = 1$ or $W(\omega) = 0$ over a certain frequency band.
- Making $W(\omega) = 0$ over a certain band means that we don't care about the error in that band. Generally, we choose $W(\omega) = 0$ around discontinuities of $H_d(e^{j\omega})$ i.e., transition bands.

Optimal FIR filter design



$W(\omega \leq \omega_p) = 1$ $\quad W(\omega_p < \omega < \omega_s) = 0$ $\quad W(\omega \geq \omega_s) = 1$

Matrix notation

It is hard to build efficient algorithms to deal with continuous ω.
We will *sample* the weighted error $E(\omega)$ for a set of N frequencies $\{\omega_1, \ldots, \omega_N\}$ and write everything in matrix notation:

$$E(\omega) = W(\omega)\Big(H_d(e^{j\omega}) - H(e^{j\omega})\Big) \qquad \text{(continuous weighted error)}$$

$$e = W(d - Qh) \qquad \text{(matrix notation)}$$

- e is the error vector $e_i = E(\omega_i)$
- W is a diagonal matrix defined as $W_{ii} = W(\omega_i)$
- d is the desired frequency response vector: $d_i = H_d(e^{j\omega_i})$
- h is the FIR filter coefficients vector $h_i = h[i]$. This is the vector we want to find.
  Note: If the filter has linear phase, $h[n]$ is symmetric, so we only need to compute the coefficients $h[0], \ldots, h[\lfloor M/2 \rfloor]$

Matrix notation

- Q is the matrix:

$$Q = \begin{bmatrix} 2\cos(\omega_1(\frac{M}{2})) & 2\cos(\omega_1(\frac{M}{2}-1)) & \dots & 2\cos(\omega_1) & 1 \\ 2\cos(\omega_2(\frac{M}{2})) & 2\cos(\omega_2(\frac{M}{2}-1)) & \dots & 2\cos(\omega_2) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 2\cos(\omega_N(\frac{M}{2})) & 2\cos(\omega_N(\frac{M}{2}-1)) & \dots & 2\cos(\omega_N) & 1 \end{bmatrix}_{N \times \frac{M}{2}+1}$$

for $h[n]$ even symmetric and M even.
This comes from the relation

$$H(e^{j\omega}) = \sum_{m=0}^{M} h[m]e^{j\omega m} = e^{-j\omega\frac{M}{2}}\left(1 + \sum_{n=0}^{\frac{M}{2}-1} 2h[n]\cos(\omega(M/2-n))\right).$$

(DTFT of symmetryic FIR $h[n]$)

Note: in matrix Q we have disregarded the term $e^{j\omega M/2}$. This way matrix Q will be purely real. Ignoring the terms $e^{j\omega M/2}$ is equivalent to disregarding the constraint that the filter must be causal. This is not a problem because we can always time-shift the result and make it causal. **Questions:** how would matrix Q change for $h[n]$ even symmetric and M odd? What about $h[n]$ odd symmetric?

Generalized linear phase in optimal FIR filter design

Even symmetry $h[n] = h[M-n]$

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

Odd symmetry $h[n] = -h[M-n]$

$$H(e^{j\omega}) = \begin{cases} e^{-j\omega\frac{M}{2}}\left(1 + \sum_{n=0}^{\frac{M}{2}-1} 2jh[n]\sin(\omega(M/2-n))\right), & M \text{ even} \\ e^{-j\omega\frac{M}{2}} \sum_{n=0}^{\frac{M-1}{2}} 2jh[n]\sin(\omega(M/2-n)), & M \text{ odd} \end{cases}$$

Optimal FIR filter design

Question: how to find the coefficients $h[0], \ldots, h[\lfloor \frac{M}{2} \rfloor]$ (the vector h)?

Two algorithms:

1. Parks-McClellan algorithm: minimizes the maximum weighted error

$$\min_{h[n]} \max_{\omega} E(\omega) \qquad \text{(min-max problem)}$$

$$\min_{h} \max_{i} |e_i| \qquad \text{(in matrix notation)}$$

   `firpm` in Matlab.

2. Least squares: minimizes the mean-square weighted error

$$\min_{h[n]} \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(\omega)|^2 d\omega \qquad \text{(least squares)}$$

$$\min_{h} ||e||_2^2 \qquad \text{(in matrix notation)}$$

   `firls` in Matlab.

Parks-McClellan algorithm

The **Parks-McClellan algorithm** finds the filter coefficients that minimize the maximum weighted error:

$$\min_{h[n]} \max_{\omega} E(\omega) \qquad \text{(min-max problem)}$$

$$\min_{h} \max_{i} |e_i| \qquad \text{(in matrix notation)}$$

- This problem is also known as the Chebyshev approximation problem
- Traditionally, this problem is solved by using the **alternation theorem** and the **Remez exchange** algorithm to iteratively find the impulse response that minimizes the maximum weighted error over a set of closed intervals in the frequency domain.
- We can also recast this problem as a **linear program** and use standard convex optimization packages to solve it.

Parks-McClellan algorithm as a linear program

$$\min_h \max_i |e_i| \qquad \text{(min-max problem)}$$

We can rewrite this optimization problem as

$$\begin{aligned} \min_u \quad & u \\ \text{subject to} \quad & -u \le e_i \le u, \quad i = 1, \dots, N \end{aligned} \qquad \text{(equivalent linear program)}$$

u is just a dummy scalar variable, and $e = W(d - Qh)$.

In CVX for Matlab:

```
cvx_begin
    variable u(1)
    variable h(floor(M/2)+1)
    minimize u
    subject to -u <= W*(d - Q*h) <= u
cvx_end
```

It will return `u` and the vector `h`, which is what we really want.

Least-squares algorithm

The **least-squares algorithm** finds the filter coefficients that minimize the mean-square weighted error:

$$\min_{h[n]} \quad \frac{1}{2\pi}\int_{-\pi}^{\pi} |E(\omega)|^2 d\omega \qquad \text{(mean square weighted error)}$$

$$\min_{h} \quad ||W(d - Qh)||_2^2 \qquad \text{(in matrix notation)}$$

$$\min_{h} \quad ||Ah - b||_2^2 \qquad \text{(change of variables } A = WQ \text{ and } b = Wd\text{)}$$

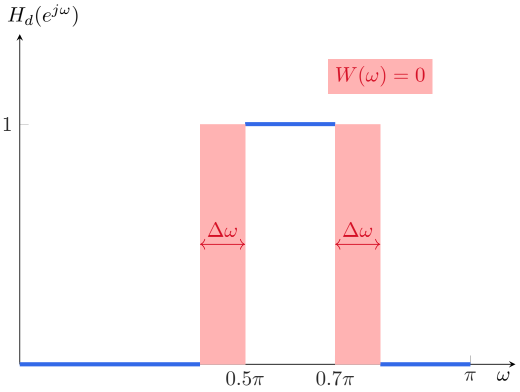
Problems of the form $\min_h ||Ah - b||_2^2$ are referred to as **least-squares problems** and they have analytical solution:

$$h = A^\dagger b \qquad \text{(least-squares solution)}$$

$A^\dagger = (A^H A)^{-1} A^H$ is the **Moore-Penrose pseudoinverse** (`pinv` in Matlab).
Note: $A^H = (A^*)^T$ is the **Hermitian** (conjugate transpose matrix), since A could be complex.

Example: optimal bandpass FIR design

We want to design an FIR bandpass filter with the following desired response $H_d(e^{j\omega})$
The weight function is zero in the **transition bands**. Hence, we don't care about the error in those regions.



See code on Canvas/Files/Matlab/optimal_fir_design_example.m.

Non-linear phase FIR filter design using least squares

Many applications do not require linear phase FIR filters. In fact, in some applications the filter must have non-linear phase e.g., linear equalization (HW#5)
To design non-linear phase FIR filters using the least-squares algorithm, we just need to redefine matrix Q:

$$Q = \begin{bmatrix} 1 & e^{-j\omega_1} & e^{-j2\omega_1} & \dots & e^{-jM\omega_1} \\ 1 & e^{-j\omega_2} & e^{-j2\omega_2} & \dots & e^{-jM\omega_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j\omega_N} & e^{-j2\omega_N} & \dots & e^{-jM\omega_N} \end{bmatrix}_{N\times M+1}$$

where $\omega_1, \dots, \omega_N$ are evenly spaced frequencies in the interval $[-\pi, \pi]$.
Note that

$$(Qh)_k = \sum_{m=0}^{M} h[m]e^{-j\omega_k m} = H(e^{j\omega_k m}) \qquad \text{(the DTFT of } h[m] \text{ at frequency } \omega_k\text{)}$$

Therefore, the matrix-vector product Qh gives $H(e^{j\omega})$ at N frequencies $\omega_1, \dots, \omega_N$.

Non-linear phase FIR filter design using least squares

Now that we have the redefined matrix Q, we can apply the least-squares algorithm as usual

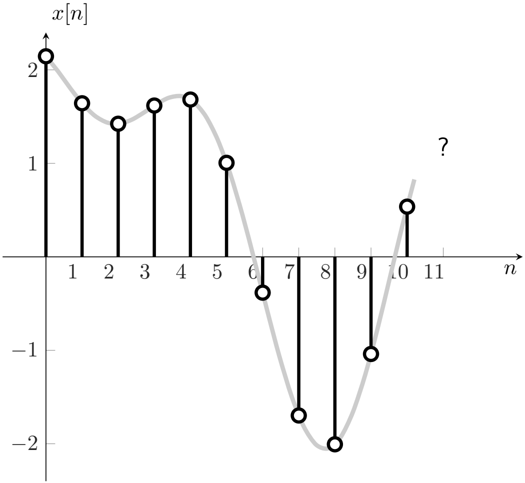
$$h = A^{\dagger} b \qquad \text{(least squares solution)}$$

where $A = WQ$ and $b = Wd$.

Important: d and W have to be defined for the same frequencies used in calculating Q.
If $H_d(e^{j\omega})$ is **Hermitian symmetric** i.e., $H_d(e^{j\omega}) = H_d^*(e^{-j\omega})$, then h will be purely real.

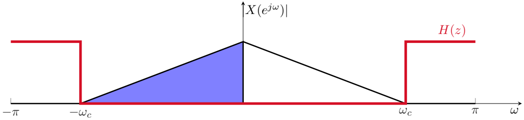
Example: predicting band-limited signals

Question: how to predict the next sample from previous samples?



Example: predicting band-limited signals

Suppose our band-limited signal is such that



Mathematically,

$$e[n] = \sum_{m=0}^{M} h[m]x[n-m] \qquad \text{(filter output)}$$

$$0 \approx h[0]x[n] + \sum_{m=1}^{M} h[m]x[n-m]$$

$$x[n] \approx -\frac{1}{h[0]} \sum_{m=1}^{M} h[m]x[n-m] \qquad \text{(prediction based on } M \text{ previous samples)}$$

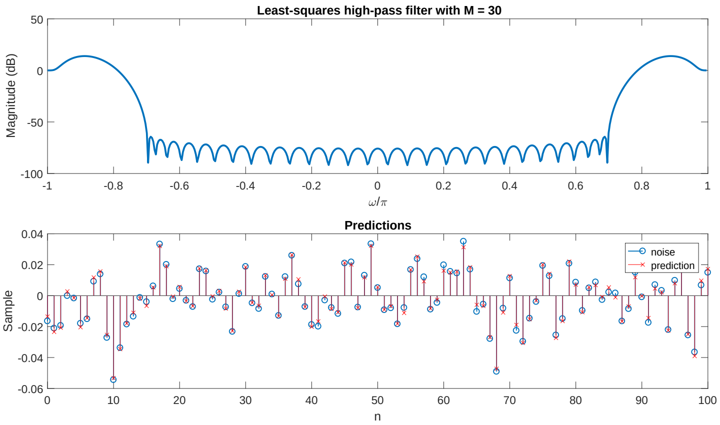
Conclusion: designing a good predictive filter for band-limited signals boils down to designing a good high-pass filter.

This method was first proposed by Vaidyanathan in 1987

Example: predicting band-limited noise

This is an example of prediction of a Gaussian noise with PSD:

$$\Phi_{xx}(e^{j\omega}) \approx \begin{cases} 1, & |\omega| \leq 0.7\pi \\ 0, & 0.7 < |\omega| \leq \pi \end{cases}$$



Summary

Impulse invariance

- The impulse response of the continuous-time system is sampled and scaled by T. In FIR implementations the impulse response is truncated up to a specified number of samples. In IIR implementations the discrete-time system is obtained analytically.

Bilinear transformation

- The bilinear transformation maps the left-hand side of the s-plane into the unit circle in the z-plane. This non-linear mapping leads to frequency warping, which can be mitigated by frequency pre-warping. Oversampling also mitigates frequency warping.

FIR filter design by windowing

- Design by windowing is almost an art form
- The Kaiser window is a nearly optimal choice

Optimal FIR filter design

- Optimal FIR filters minimize some characteristic of the weighted error
- The Parks-McClellan method minimizes the maximum weighted error
- The least-squares method minimizes the mean-square weighted error