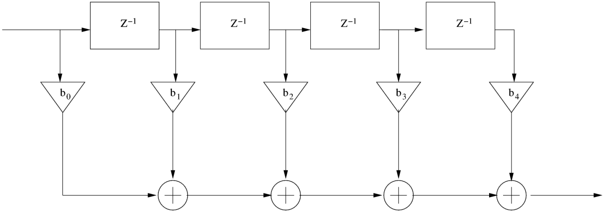
EDISP (FILTlect)
(English) Digital Signal Processing
Filters & filter design

April 28, 2015

A simple FIR filter



- Transversal structure – implements linear convolution

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + b_3 x(n-3) + b_4 x(n-4)$$

$$Y(z) = b_0 X(z) + b_1 X(z) z^{-1} + b_2 X(z) z^{-2} + b_3 X(z) z^{-3} + b_4 X(z) z^{-4}$$

$$Y(z) = X(z) \left(b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4} \right)$$

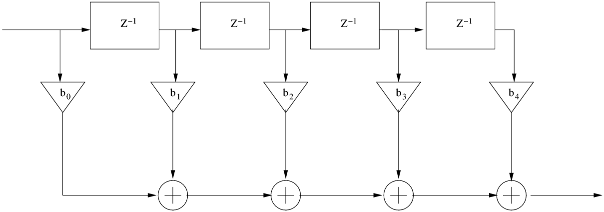
$$H(z) = Y(z)/X(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}$$

- Impulse response

$$h(n) = b_0 \delta(n) + b_1 \delta(n-1) + b_2 \delta(n-2) + b_3 \delta(n-3) + b_4 \delta(n-4)$$

A simple FIR filter

Frequency response



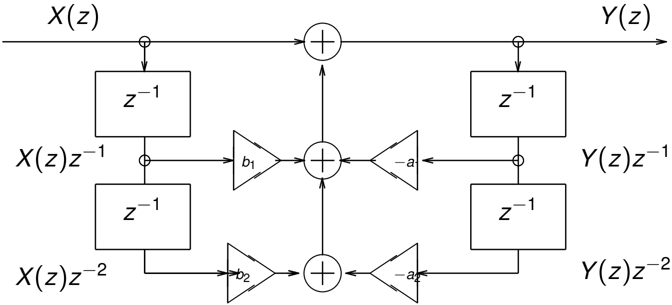
$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}$$

$$H(e^{j\theta}) = b_0 + b_1 e^{-j\theta} + b_2 e^{-j2\theta} + b_3 e^{-j3\theta} + b_4 e^{-j4\theta}$$

$$H(e^{j\theta}) = e^{-j2\theta}\left(b_0 e^{+j2\theta} + b_1 e^{+j\theta} + b_2 + b_3 e^{-j\theta} + b_4 e^{-j2\theta}\right)$$

- It is easy to make the part $(\ldots)$ real-valued by introducing symmetry of coefficients
- The polynomial of 4th order will have 4 roots (=zeros of the transfer function)
- Zeros at the unit circle = notches in frequency response

A simple IIR system (biquadratic section)

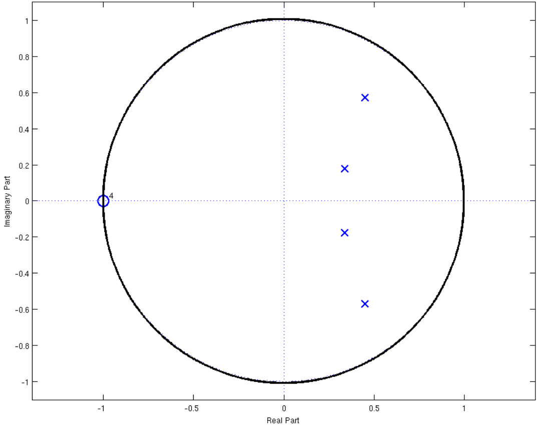


$$\sum_{m=0}^{2} a_m Y(z) z^{-m} = \sum_{k=0}^{2} b_k X(z) z^{-k}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{2} b_k z^{-k}}{\sum_{m=0}^{2} a_m z^{-m}}$$

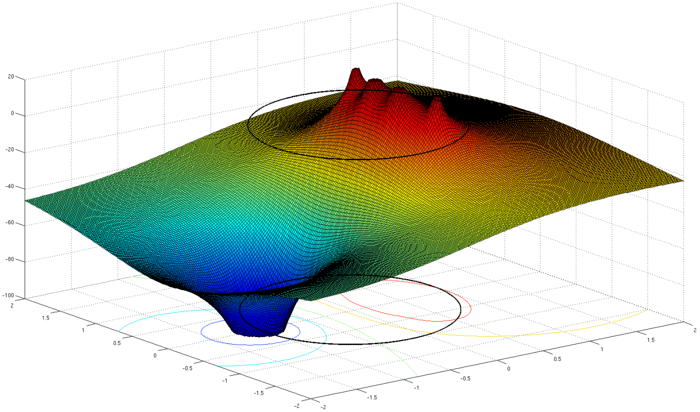
- Numerator = a polynomial of 2nd order $\longrightarrow$ two zeros of transfer function
- Denominator = a polynomial of 2nd order $\longrightarrow$ two poles of transfer function

A fourth order IIR system



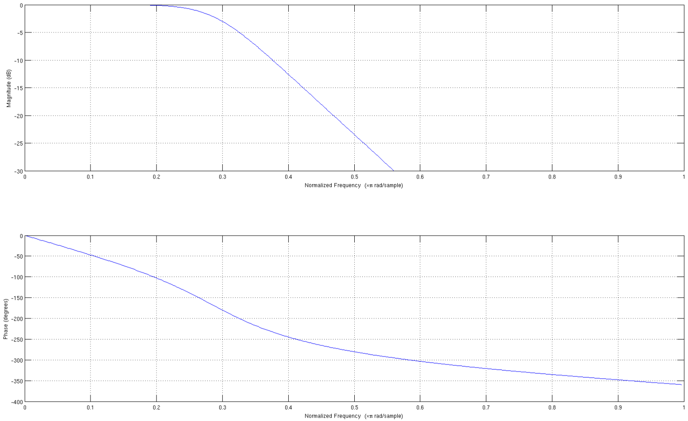
- Four poles in the vicinity of $\theta = 0$, one four-fold zero at $\theta = \pi$

A fourth order IIR system



- Z-plane viewed in 3D

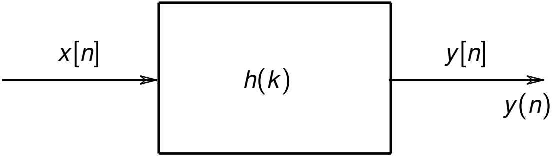
A fourth order IIR system



- Four poles in the vicinity of $\theta = 0$, one four-fold zero at $\theta = \pi$
- Poles keep passband up
- Zeros form the stopband

Frequency properties of a DT system

An LTI system is described with its impulse response



$$y(n) = \sum_{k=-\infty}^{\infty} h(k) \cdot x(n-k)$$

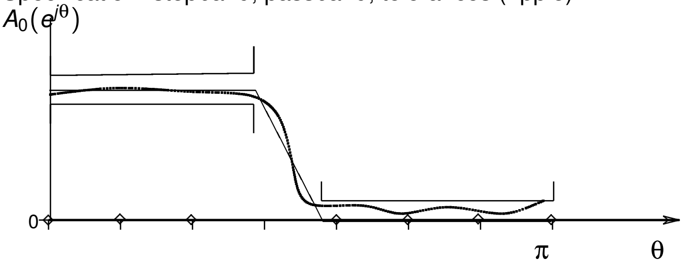
which is a description in time domain — but we are interested in its properties in the frequency domain (frequency response)

$$H(e^{j\theta}) = H(z)\big|_{z=e^{j\theta}} = \sum_{n=-\infty}^{\infty} h(n) e^{-jn\theta}$$

Magnitude of fr. response	$A(\theta)$	$=$	$\lvert H(e^{j\theta})\rvert$
Phase of fr. response	$\varphi(\theta)$	$=$	$arg[H(e^{j\theta})]$
Group delay	$\tau(\theta)$	$=$	$-d\varphi(\theta)/d\theta$

Filter design

- Specification: stopband, passband, tolerances (ripple)



- Approximation: find best rational function which fits specifications

$$\frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}} \quad \text{(IIR)}$$

or

$$b_0 + b_1 z^{-1} + \ldots + b_M z^{-M} \quad \text{(FIR)}$$

determine order and coefficients, check stability

- Implementation: structure, noise, hardware/software ...

Filter stability

We may check stability:

- from impulse response $\sum_{k=-\infty}^{\infty} |h(k)| < \infty$
- at first glance: **FIR is always stable** (see above)
- from $H(z)$: a pole d_k produces a term

$$\frac{A_k}{1 - d_k z^{-1}}, \quad A_k = (1 - d_k z^{-1}) \cdot X(z)\Big|_{z=d_k}$$

  in the partial fraction expansion of $H(z)$;

  $\frac{1}{1-d_k z^{-1}}$ is a Z transform of $d_k^n u(n)$,
  which is a stable term in $h(n)$ if $|d_k| < 1$.

  $\longrightarrow$ **all poles must be inside unit circle** $|z| = 1$ (for a stable causal system)
  outside for an anticausal one

- by time-domain analysis by hand (recommended only as last resort)

Filter stability pitfalls

Side remarks on mathematics

A fundamental formula (*absolute convergence*): $\sum_{k=-\infty}^{\infty} |h(k)| < \infty$ is sometimes misunderstood:

- It is NOT enough to check $h(k) \longrightarrow 0$ – a counterexample: $\sum_{k=0}^{\infty} 1/k$ diverges
- It is NOT enough to check it without absolute value: $\sum_{k=-\infty}^{\infty} h(k) < \infty$ – $\sum_{k=0}^{\infty} \frac{(-1)^{k+1}}{k}$ converges to $\ln 2$, but if you apply absolute value you will get $\sum_{k=0}^{\infty} 1/k$ which diverges

For the last case do a math experiment: convolve $h(k) = \frac{(-1)^{k+1}}{k}$

- with $x(n) = 1$ – you will get an $y(n)$ approaching $\ln 2$ for large n
- with $x(n) = (-1)^n$ – you will get a divergent $y(n)$

Filter types

LP lowpass – a basic type

HP highpass – how to make HP from LP? (Hint: $h_{LP} \cdot (-1)^n$)

BP bandpass – combine LP with HP

BS bandstop

notch a very narrow bandstop (e.g with a zero on the unit circle)

allpass – usually used for correcting phase response

Filter design in practice

Plan

- FIR - window method (LP example, BP/HP howto)
- FIR - optimization methods (Parks-McClellan, called also Remez)
- IIR - bilinear transformation

- IIR - impulse/step response invariance (next lecture)
- IIR - optimization methods (next lecture)

FIR filter design – window method

- Ideal filter: $A_0(\theta) = \begin{cases} 1 & \text{for} \quad |\theta| < \theta_p \\ 0 & \text{for} \quad \theta_p < |\theta| \leq \pi \end{cases}$ and zero phase
- Impulse response:

$$h_0(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_0\left(e^{j\theta}\right) e^{jn\theta}\, d\theta \; = \; \frac{\theta_p}{\pi} \, \frac{\sin n\theta_p}{n\theta_p}$$

is non-causal and infinite!

- Make it finite: $h_P[n] \; = \; h_0[n]\, g[n]$ ($g[n] = 0$ for $|n| > P$)
- Shift it to be causal – delay by P samples: $h[n] = h_P[n-P]$

$\longrightarrow$ finally we obtain

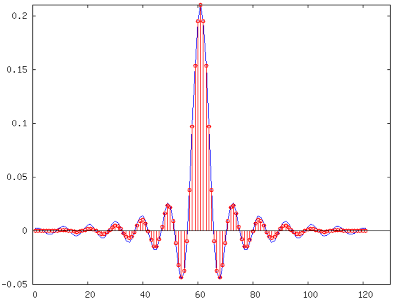
$$H(z) \; = \; \sum_{n=0}^{2P} h(n)\, z^{-n} \; = \; z^{-P} H_P(z)$$

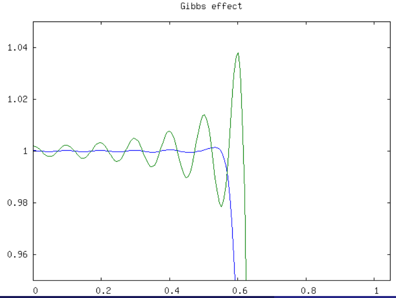
FIR LP filter by window method

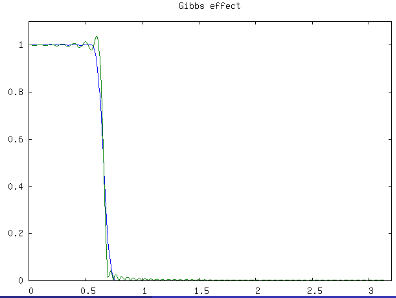
LP filter - pass from $-\theta_p$ to $+\theta_p$

$$h_0(n) = \frac{1}{2\pi}\int_{-\theta_p}^{\theta_p} e^{jn\theta}\, d\theta = \frac{\theta_p}{\pi}\,\frac{\sin n\theta_p}{n\theta_p}$$

Cut at order 120. Shift to be causal.







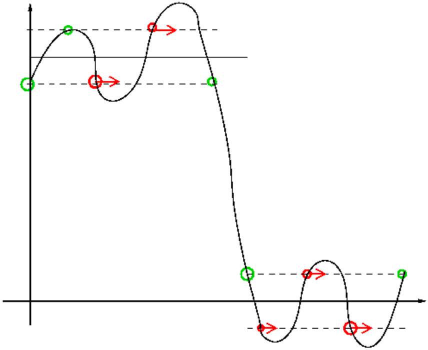
FIR - optimization methods

Window method - simple, easy, all under strict control. But is it “best” filter for given order?

yes a rectangular window gives best approximation in the MS sense

no *we know about problems (Gibbs effect) at the discontinuities*
so we try to cheat with Windows

So, Parks & Mc Clellan (1972) used Chebyshev (minimax) approximation on discrete set of points in θ. They applied E. Ya. Remez (1934) algorithm.



Approx 12 iterations needed.

IIR - bilinear transformation

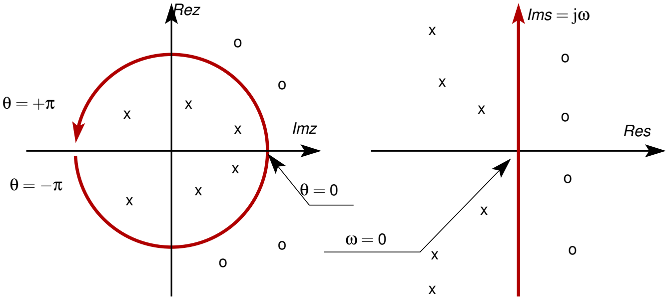
We use analog filter prototype:

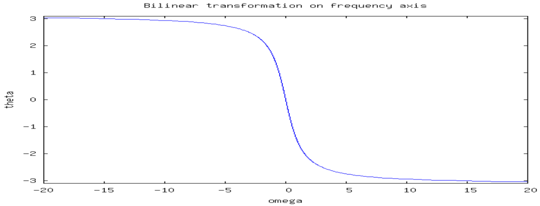
- good theory
- prototype polynomials $\longrightarrow$ known properties
- tables, methods, algorithms $\longrightarrow$ well known and fast

"Copy" a CT prototype $H(s)$ to DT domain $H(z)$:

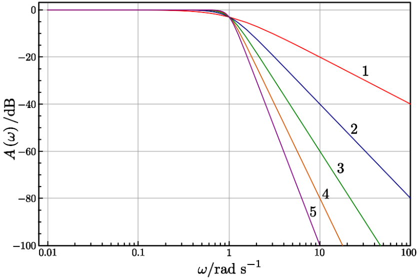
- $\longrightarrow$ substitute $s = \frac{2}{T_d}\frac{1-z^{-1}}{1+z^{-1}}$ (trapezoidal inetgration of $H(s)$ with step T_d)
- roll the $j\omega$ line to $e^{j\omega}$ circle
- A point θ is mapped from $\omega = \frac{2}{T_d} tan(\theta/2)$
- $\longrightarrow$ we need to pre-warp our frequency characteristics from θ to ω
- Stability $\longrightarrow$ left half-plane transformed into inside of unit circle (OK!)

Bilinear transformation: z-plane and s-plane

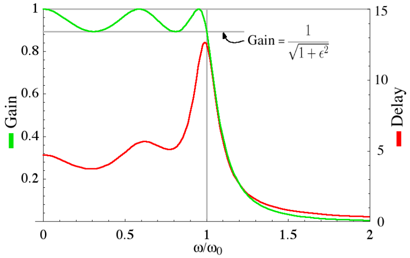




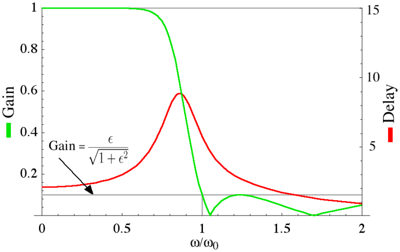
IIR - bilinear transformation - analog prototypes

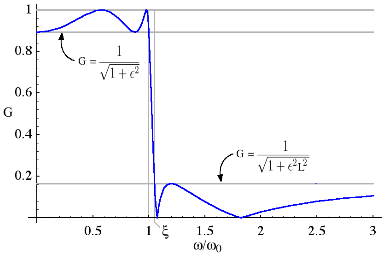


Butterworth (max. flat amplitude)



Chebyshev type I





IIR and FIR in Matlab

Utilities

- Filtering: `y=filter(B,A,x);`
  - B - numerator coefficients
  - A - denominator coefficients (if FIR $\longrightarrow A = [1]$)
  - x - input samples vector
- Filter frequency response: `[h, w]=freqz(B, A);`
  - w frequency values (0 to π),
  - `abs(h)` Magnitude of response
  - `angle(h)` Phase of response ($-\pi$ to π)
- Filter group delay: `[gd, w]=grpdelay(B, A);`

- Filter design specification: frequency from 0.0 ($\longrightarrow$ zero) to 1.0 ($\longrightarrow f_s/2$)
- Window method (FIR): `B = FIR2(N,F,A[, window]);`

  N: order
  F: frequency points
  A: amplitude characteristics at points specified by F
  window: e.g. Bartlett(N+1) or chebwin(N+1, R)

- IIR bilinear method (Butterworth as example):

  `[N, wn]=buttord(Wp, Ws, Rp, Rs);`

  Wp, Ws: passband freq, stopband freq,
  Rp, Rs: ripple in passband, ripple in stopband
  N, wn: order and 3dB point warped and adjusted

  `[B,A]=butter(N, wn);`

  does the polynomial design and bilinear transform.