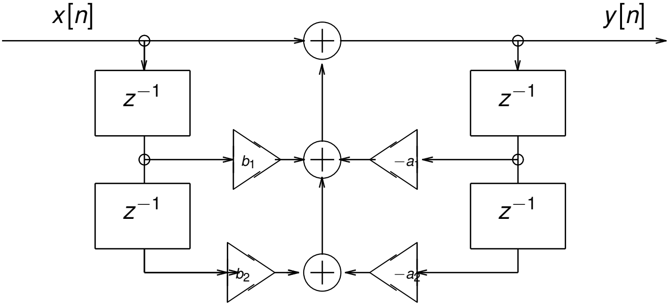
EDISP (Filters 2)
(English) Digital Signal Processing
Digital (Discrete Time) filters 2
lecture

January 30, 2011

How an LTI system filters signals

- A practical system and its difference equation
- Difference equation and $H(z)$
- Short path: system $\longrightarrow H(z)$
- System defined by $H(z)$ + harmonic signal
- Is my filter stable?

System and its difference equation



$$y(n) = x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

$$y(n) + a_1 y(n-1) + a_2 y(n-2) = x(n) + b_1 x(n-1) + b_2 x(n-2)$$

$$\sum_{m=0}^{2} a_m y(n-m) = \sum_{k=0}^{2} b_k x(n-k)$$

Difference equation and $H(z)$

z^{-1} - shift operator

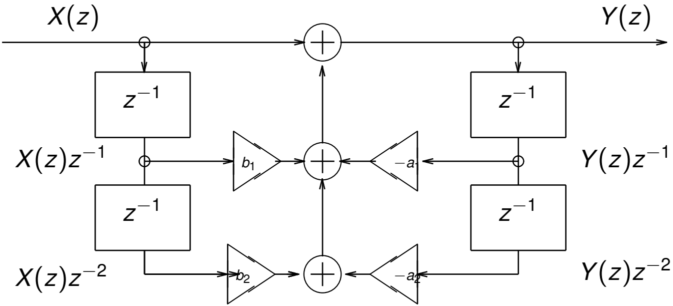
$$\sum_{m=0}^{2} a_m y(n-m) = \sum_{k=0}^{2} b_k x(n-k)$$

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

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

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

System and its $H(z)$



$$Y(z) = X(z) + b_1 X(z)z^{-1} + b_2 X(z)z^{-2} - a_1 Y(z)z^{-1} -$$

$$Y(z) + a_1 Y(z)z^{-1} + a_2 Y(z)z^{-2} = X(z) + b_1 X(z)z^{-1} + b_2 X(z)z^{-2}$$

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

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

$H(z)$ to $h(n)$ (or how to find Z^{-1})

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

$$= A \frac{\prod_{k=0}^{2}(1 - c_k z^{-1})}{\prod_{m=0}^{2} a_m z^{-m}(1 - d_m z^{-1})}$$

Zeros at $z = c_k \longrightarrow (1 - c_k z^{-1}) = \frac{z - c_k}{z - 0}$ (plus pole at $z = 0$).
Poles at $z = d_m \longrightarrow \frac{1}{(1 - d_m z^{-1})} = \frac{z - 0}{z - d_m}$ (plus a zero at $z = 0$).

System defined by $H(z)$ + harmonic signal

$$x(n) = e^{jn\theta} \longrightarrow \boxed{\text{h(n)}} \longrightarrow y(n) = ?$$

$$\begin{aligned} y(n) &= \sum_k h(k) \cdot e^{j(n-k)\theta} = \\ &= \sum_k h(k) \cdot e^{j(-k)\theta} \cdot e^{jn\theta} = \\ &= e^{jn\theta} \sum_k h(k) \cdot e^{j(-k)\theta} = \\ &= e^{jn\theta} H(\theta) \\ H(\theta) &= A(\theta) e^{j\phi(\theta)} \end{aligned}$$

If $x(n)$ is periodic - we can decompose it into harmonics (linearity).

System defined by $H(z)$ + sine/cosine signal

$$x(n) = e^{jn\theta} \longrightarrow \boxed{\text{h(n)}} \longrightarrow y(n) = e^{jn\theta} H(\theta)$$

so if $x(n) = cos(n\theta) = 1/2 \cdot (e^{jn\theta} + e^{-jn\theta})$
then $y(n) = 1/2 \cdot (e^{jn\theta} + e^{-jn\theta}) \cdot H(\theta)$

$\text{H}(\theta) = A(\theta) e^{j\phi(\theta)}$

and $y(n) = A(\theta) \cdot 1/2 \cdot (e^{jn\theta} + e^{-jn\theta}) \cdot e^{j\phi(\theta)}$
(for a real h(n) $\phi(\theta)$ is odd: $\phi(-\theta) = -\phi(\theta)$)
and $y(n) = A(\theta) \cdot 1/2 \cdot (e^{j(n\theta + \phi(\theta))} + e^{-j(n\theta + \phi(\theta))})$

y(n)=A$(\theta) \cdot cos(n\theta + \phi(\theta))$

Repeat the same with $sin()$ $\longrightarrow$ at home.

E.g. $x(n) = 3 + 5 sin(0.1\pi n)$ $\longrightarrow$ a DC component and a 0.1π harmonic signal. So $y(n) = A(0) \cdot 3 + A(0.1\pi) \cdot 5 sin(0.1\pi n + \phi(0.1\pi))$.
Note: Z-transform is *NOT* a good tool to calculate this!!

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 design in practice

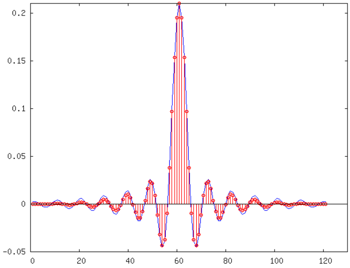
- FIR - window method (LP example, BP/HP howto)
- FIR - optimization methods (Parks-McClellan, called also Remez)
- IIR - bilinear transformation
- IIR - impulse/step response invariance (next lecture)
- IIR - optimization methods (next lecture)

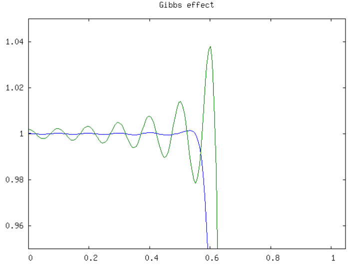
FIR LP filter by window method

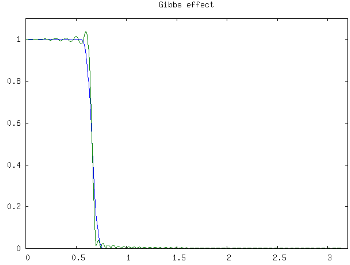
LP filter - pass from $-\theta_p$ to $+\theta_p$

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

Cut at order 120. Shift to be causal.







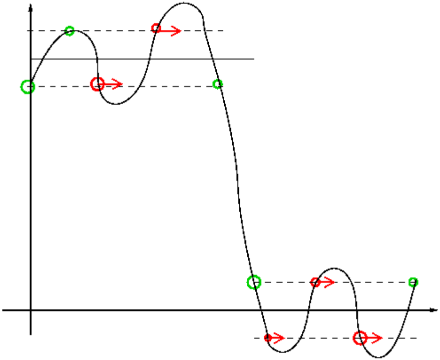
FIR - optimization methods

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.

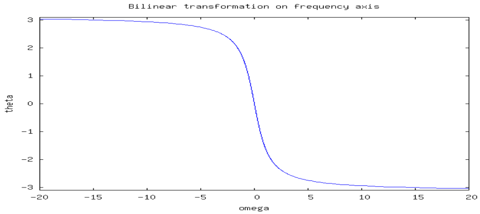
IIR - bilinear transformation

We use analog filter prototype:

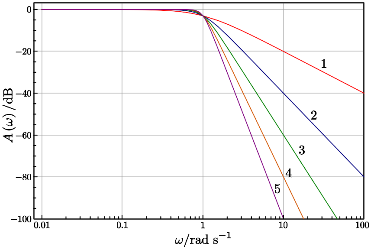
- good theory
- prototype polynomials $\longrightarrow$ known properties
- tables, methods

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

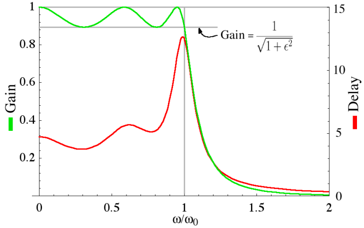
- $\longrightarrow$ substitute $s = \frac{2}{T_d}\frac{1-z^{-1}}{1+z^{-1}}$ (trapezoidal inetgration of $H(s)$ with step T_d
- roll the $j\omega$ line to $e^{j\omega}$ circle
- A point θ is mapped from $\omega = \frac{2}{T_d} tan(\theta/2)$
- $\longrightarrow$ we need to pre-warp our frequency characteristics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



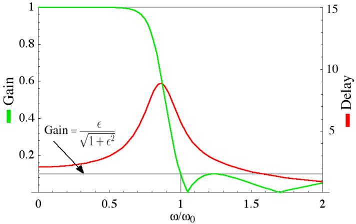
IIR - bilinear transformation - analog prototypes



Butterworth (max. flat amplitude)

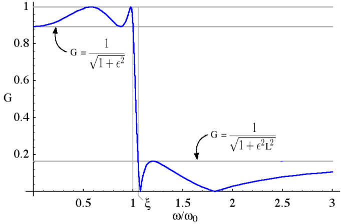


Chebyshev type I



Chebyshev type II

Bessel - maximally flat phase



Cauer (elliptical)

IIR - bilinear transformation - Matlab

- Filtering: `y=filter(B,A,x);`
  - B - numerator coefficients
  - A - denominator coefficients (if FIR $\longrightarrow A = [1]$)
  - x - input samples vector
- Filter characteristics: `[h, w]=freqz(B, A);`
  - w frequency values,
  - `abs(h)` amplitude characteristics
- 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.