Analog IIR Filter Design

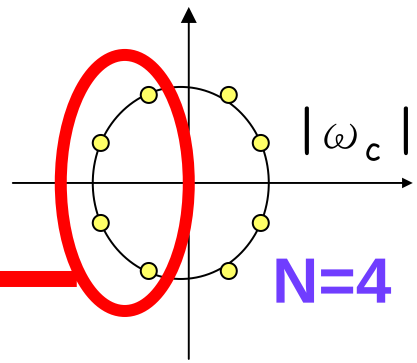
Commonly used analog filters :

- Lowpass Butterworth filters
  all-pole filters characterized by magnitude response.
  (N=filter order)

$$G(j\omega)=|H(j\omega)|^2=\frac{1}{1+(\frac{\omega}{\omega_c})^{2N}}$$

$$G(s)=H(s)H(-s)=\frac{1}{1+(\frac{-s^2}{\omega_c^2})^N}$$

Poles of H(s)H(-s) are equally spaced points on a circle of radius ω_c in s-plane



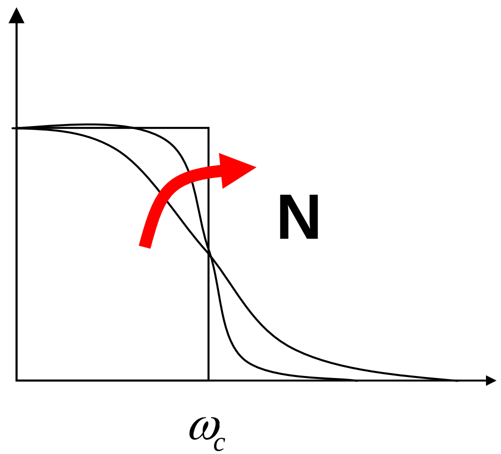
poles of H(s)

N=4

Butterworth Filters

- Lowpass Butterworth filters

  monotonic in pass-band & stop-band



\`maximum flat response': (2N-1) derivatives are zero at
$\omega = 0$ and $\omega = \infty$

DEPT. OF ELECTRONICS ENGINEERING & INST. OF ELECTRONICS NCTU

Analog IIR Filter Design

Commonly used analog filters :

- Lowpass Chebyshev filters (type-I)

all-pole filters characterized by magnitude response

(N=filter order)

$$G(j\omega)=|H(j\omega)|^2=\frac{1}{1+\varepsilon^2 T_N^2(\frac{\omega}{\omega_c})}$$

$$G(s)=H(s)H(-s)$$

ε is related to passband ripple

$T_N(x)$ are Chebyshev polynomials:

$$T_0(x)=1$$
$$T_1(x)=x$$
$$T_2(x)=2x^2-1$$
$$\ldots$$
$$T_N(x)=2xT_{N-1}(x)-T_{N-2}(x)$$

Chebyshev & Elliptic Filters

- Lowpass Chebyshev filters (type-I)
  - All-pole filters, poles of H(s)H(-s) are on ellipse in s-plane
  - Equiripple in the pass-band
  - Monotone in the stop-band
- Lowpass Chebyshev filters (type-II)
  - Pole-zero filters based on Chebyshev polynomials
  - Monotone in the pass-band
  - Equiripple in the stop-band

$$\left|H(j\omega)\right|^2 = \frac{1}{1+\varepsilon^2 U_N(\frac{\omega}{\omega_c})}$$

- Lowpass Elliptic (Cauer) filters
  - Pole-zero filters based on Jacobian elliptic functions
  - Equiripple in the pass-band and stop-band
  - (hence) yield smallest-order for given set of specs

DEPT. OF ELECTRONICS ENGINEERING & INST. OF ELECTRONICS NCTU

Analog IIR Filter Design

Frequency Transformations :

- Principle : prototype low-pass filter (e.g. cut-off frequency = 1 rad/sec) is transformed to properly scaled low-pass, high-pass, band-pass, band-stop,… filter

- example: replacing s by $\frac{s}{\omega_C}$ moves cut-off frequency to ω_C

- example: replacing s by $\frac{\omega_C}{s}$ turns LP into HP, with cut-off ω_C frequency

- example: replacing s by $\frac{s^2+\omega_1\omega_2}{s(\omega_2-\omega_1)}$ turns LP into BP

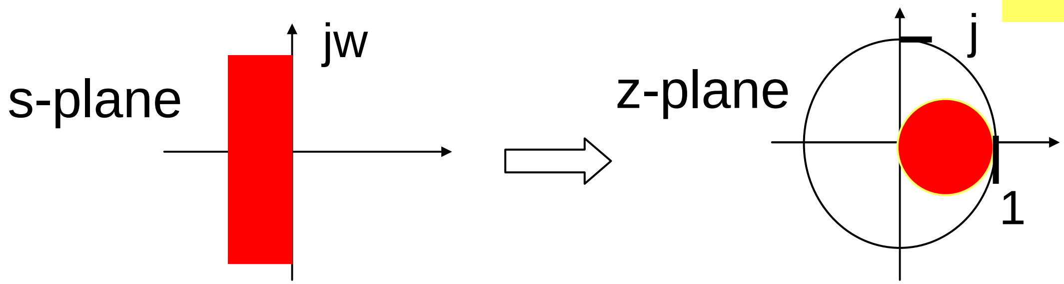
Analog -> Digital

- Principle :
  - design analog filter (LP/HP/BP/…), and then convert it to a digital filter.
- Conversion methods:
  - convert differential equation into difference equation
  - convert continuous-time impulse response into discrete-time impulse response
  - convert transfer function H(s) into transfer function H(z)
- Requirement:
  - the left-half plane of the s-plane should map into the inside of the unit circle in the z-plane, so that a stable analog filter is converted into a stable digital filter.

Analog -> Digital

(I) convert differential equation into difference equation :

- in a difference equation, a derivative dy/dt is replaced by a 'backward difference' (y(kT)-y(kT-T))/T=(y[k]-y[k-1])/T, where T=sampling interval.
- similarly, a second derivative, and so on.
- eventually (details omitted), this corresponds to replacing s by (1-1/z)/T in $H_a(s)$ (=analog transfer function) : $H(z) = H_a(s)\big|_{s=\frac{1-z^{-1}}{T}}$



$s = 0 \Rightarrow z = 1$

$s = \infty \Rightarrow z = 0$

- stable analog filters are mapped into stable digital filters, but pole location for digital filter confined to only a small region (o.k. only for LP or BP)

Analog -> Digital

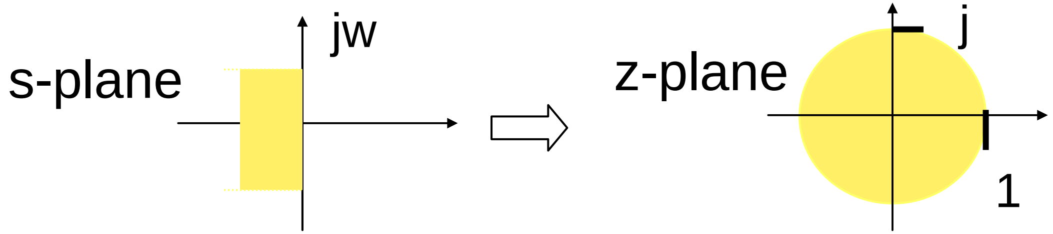
(II) convert continuous-time impulse response into discrete-time impulse response :

- given continuous-time impulse response $h_c(t)$, discrete-time impulse response is $h[k] = h_c(kT_d)$ where T_d=sampling interval.
- eventually (details omitted) this corresponds to a (many-to-one) mapping

$$z = e^{sT_d}$$

$$s = 0 \Rightarrow z = 1$$

$$s = \pm j\pi / T_d \Rightarrow z = -1$$



- aliasing (!) if continuous-time response has significant frequency content above the Nyquist frequency (i.e. not bandlimited)

Example: Filter Design by Impulse Invariance

If $h_c(t)$ is the impulse response of continuous-time filter, and $h_c(nT_d)$ is equally spaced samples of it.

Let $$h[n] = T_d h_c(nT_d) \qquad (7.4)$$

then the corresponding frequency responses meet following equation:

$$H(e^{j\omega}) = T_d \cdot \frac{1}{T_d} \sum_{k=-\infty}^{\infty} H_c(j\frac{\omega}{T_d} + j\frac{2\pi}{T_d}k) \qquad (7.5)$$

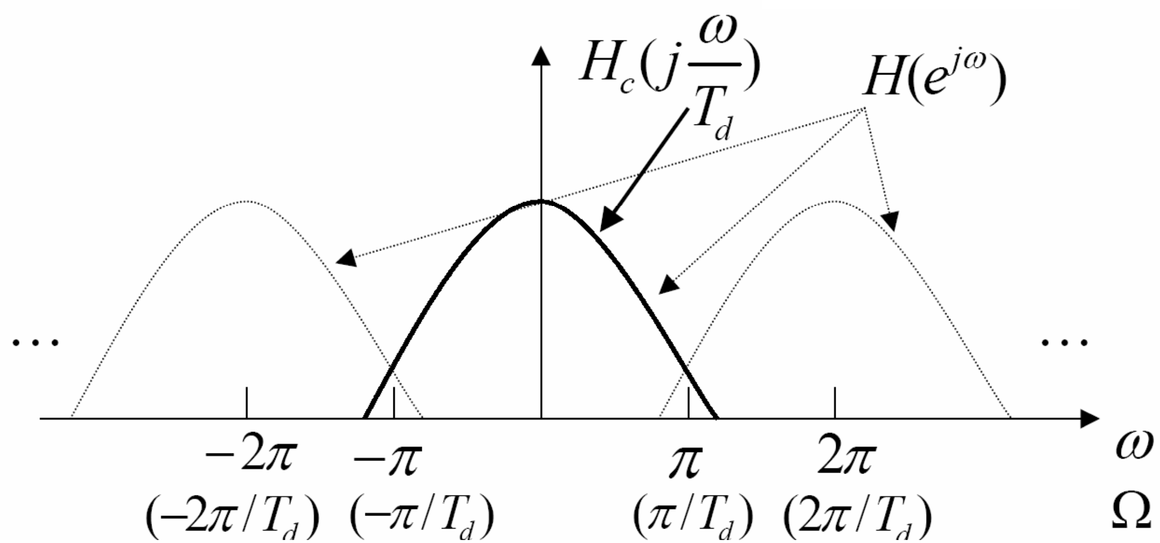
If the continuous-time filter is band-limited, so that

$$H_c(j\Omega) = 0, \quad |\Omega| \geq \pi / T_d \qquad (7.6)$$

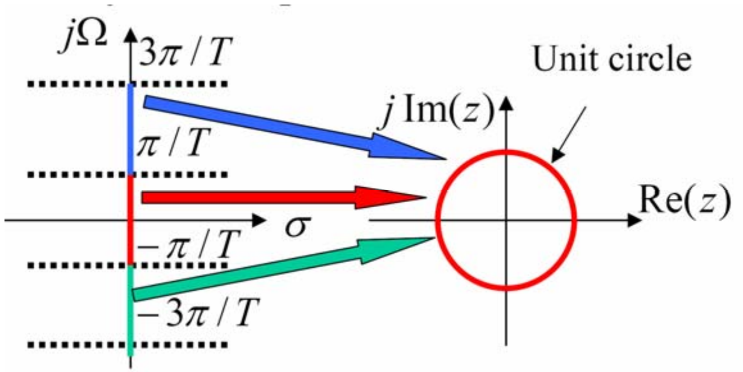
then

$$H(e^{j\omega}) = H_c(j\frac{\omega}{T_d}), \quad |\omega| \leq \pi \qquad (7.7)$$

If the continuous-time filter is not band-limited, then the interference (aliasing) between successive terms exists .



Many-to-one mapping



Example: A Low-Pass Filter

Specifications:

$$0.89125 \le \left|H(e^{j\omega})\right| \le 1, \quad 0 \le |\omega| \le 0.2\pi \qquad (7.13a)$$

$$\left|H(e^{j\omega})\right| \le 0.17783, \qquad 0.3\pi \le |\omega| \le \pi \qquad (7.13b)$$

Choose $T_d = 1$ (i.e. $\omega = \Omega$), from Eq.(7.7) we obtain

$$H(e^{j\omega}) = H_c(j\frac{\omega}{T_d}), \quad |\omega| \le \pi$$

Then $$0.89125 \le \left|H_c(j\Omega)\right| \le 1, \quad 0 \le |\Omega| \le 0.2\pi$$

$$\left|H_c(j\Omega)\right| \le 0.17783, \qquad 0.3\pi \le |\Omega| \le \pi$$

Let $\Omega_p = 0.2\pi$, and $\Omega_s = 0.3\pi$, then

$$\left|H_c(j0.2\pi)\right| \ge 0.89125$$

$$\left|H_c(j0.3\pi)\right| \le 0.17783,$$

By using Butterworth filter, then $|H_c(j\Omega)|^2 = \dfrac{1}{1+(\Omega/\Omega_c)^{2N}}$

Consequently,

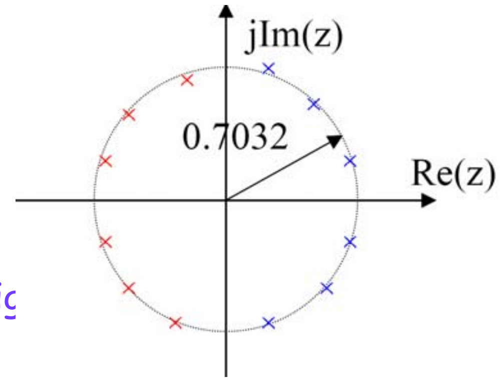
$$1+\left(\frac{0.2\pi}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.89125}\right)^2 \qquad 1+\left(\frac{0.3\pi}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.17783}\right)^2$$

Then

$$\left(\frac{0.2\pi}{0.3\pi}\right)^{2N} = \frac{\left(\frac{1}{0.89125}\right)^2 - 1}{\left(\frac{1}{0.17783}\right)^2 - 1} \Rightarrow \quad N=5.8858$$

Since N must be integer. So, N=6. And we obtain Ω_c=0.7032

We can obtain 12 poles of $|H_c(s)|^2$. They are uniformly distributed in angle on a circle of radius Ω_c=0.7032



so $H_c(s) = \dfrac{K_0}{\prod_{k=1}^{N}(s - s_k)}$ K_0=0.12093

so that

$$H_c(s) = \frac{0.12093}{(s^2 + 0.3640s + 0.4945)(s^2 + 0.9945s + 0.4945)(s^2 + 1.3585s + 0.4945)}$$

Discrete Filter

$$H(z) = \frac{0.2871 - 0.4466z^{-1}}{1 - 1.2971z^{-1} + 0.6949z^{-2}} + \frac{-2.1428 + 1.1455z^{-1}}{1 - 1.0691z^{-1} + 0.3699z^{-2}} + \frac{1.8557 - 0.6303z^{-1}}{1 - 0.9972z^{-1} + 0.2570z^{-2}}. \qquad (7.19)$$

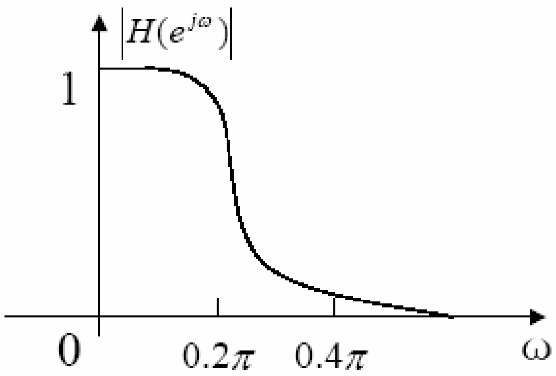


Figure 7.6 (b) Frequency response of sixth-order Butterworth filter

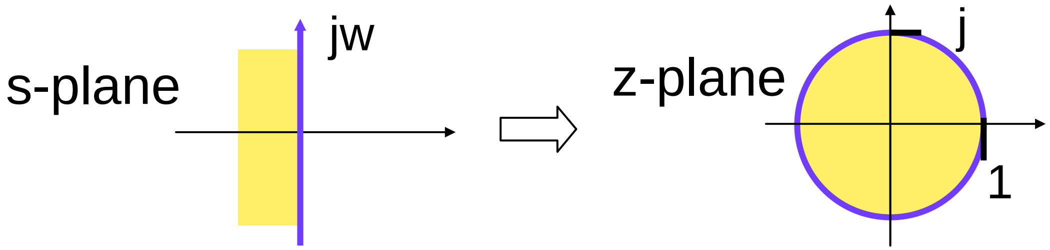
Analog -> Digital

- (III) convert continuous-time system transfer function into discrete-time system transfer function : Bilinear Transform
  - mapping that transforms (whole!) jw-axis of the s-plane into unit circle in the z-plane only once, i.e. that avoids aliasing of the frequency components.

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

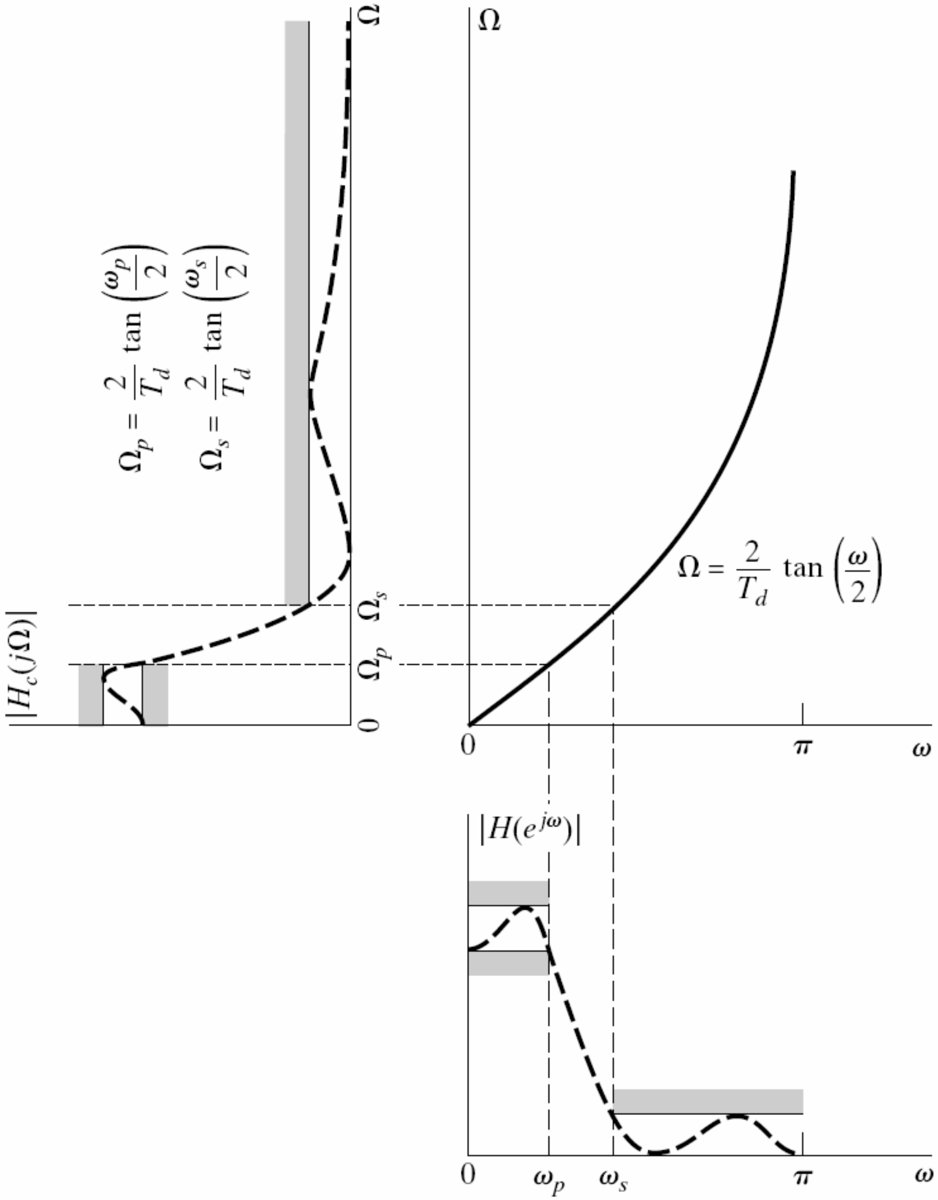
$$s = 0 \Rightarrow z = 1$$

$$s = j.\infty \Rightarrow z = -1$$



  - for low-frequencies, this is an approximation of $z = e^{sT}$
  - for high frequencies : significant frequency compression (\`warping')
  - sometimes pre-compensated by 'pre-warping'

Non-linear transform



$$s = \frac{2}{T_d}\left(\frac{1-e^{-j\omega}}{1+e^{-j\omega}}\right)$$

$$s = \sigma + j\Omega = \frac{2}{T_d}\left[\frac{e^{-j\omega/2}(e^{j\omega/2} - e^{-j\omega/2})}{e^{-j\omega/2}(e^{j\omega/2} + e^{-j\omega/2})}\right]$$

$$= \frac{2}{T_d}\left[\frac{2j\sin(\omega/2)}{2\cos(\omega/2)}\right] = \frac{2j}{T_d}\tan(\omega/2)$$

If $\sigma = 0$, then $\Omega = \frac{2}{T_d}\tan(\omega/2)$

The bilinear transformation avoids the problem of aliasing problem because it maps the entire imaginary axis of the s-plane onto the unit circle in the z-plane. The price paid for this, however, is the nonlinear compression the frequency axis (warping).

Conclusions/Software

- IIR filter design considerably more complicated than FIR design (stability, phase response, etc..)
- (Fortunately) IIR Filter design abundantly available in commercial software
- Matlab:

[b,a]=butter/cheby1/cheby2/ellip(n,…,Wn),

IIR LP/HP/BP/BS design based on analog prototypes, pre-warping, bilinear transform, …