http://elec3004.com

Digital Filters IIR
(& Their Corresponding Analog Filters)

ELEC 3004: **Systems**: Signals & Controls
Dr. Surya Singh

Lecture 10

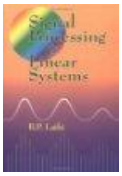
elec3004@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~elec3004/

April 4, 2017

Lecture Schedule:

Week	Date	Lecture Title
1	28-Feb	Introduction
	2-Mar	Systems Overview
2	7-Mar	Systems as Maps & Signals as Vectors
	9-Mar	Systems: Linear Differential Systems
3	14-Mar	Sampling Theory & Data Acquisition
	16-Mar	Aliasing & Antialiasing
4	21-Mar	Discrete Time Analysis & Z-Transform
	23-Mar	Second Order LTID (& Convolution Review)
5	28-Mar	Frequency Response
	~~30-Mar~~	~~Filter Analysis~~
6	4-Apr	Digital Filters (IIR) & Filter Analysis
	6-Apr	Digital Windows
7	11-Apr	Digital Filter (FIR)
	13-Apr	FFT
	18-Apr	Holiday
	20-Apr	
	25-Apr	
8	27-Apr	Active Filters & Estimation
9	2-May	Introduction to Feedback Control
	4-May	Servoregulation/PID
10	9-May	Introduction to (Digital) Control
	11-May	Digitial Control
11	16-May	Digital Control Design
	18-May	Stability
12	23-May	Digital Control Systems: Shaping the Dynamic Response
	25-May	Applications in Industry
13	30-May	System Identification & Information Theory
	1-Jun	Summary and Course Review

Follow Along Reading:



B. P. Lathi
Signal processing and linear systems
1998
TK5102.9.L38 1998

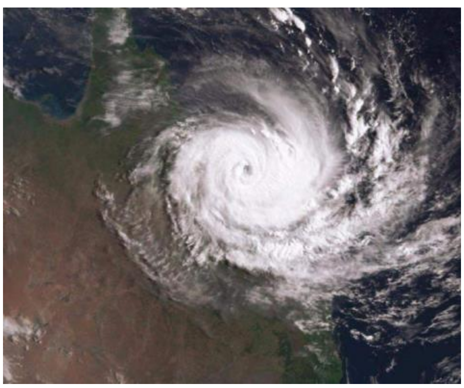
Today

- Chapter 10
  (**Discrete-Time System Analysis Using the *z*-Transform**)
  – § 10.3 Properties of DTFT
  – § 10.5 Discrete-Time Linear System analysis by DTFT
  – § 10.7 Generalization of DTFT to the $\mathcal{Z}$–Transform

- Chapter 12
  (**Frequency Response and Digital Filters**)
- § 12.1 Frequency Response of Discrete-Time Systems
- § 12.3 Digital Filters
- § 12.4 Filter Design Criteria
- § 12.7 Nonrecursive Filters

Next Time

Announcements: Cyclone Debbie

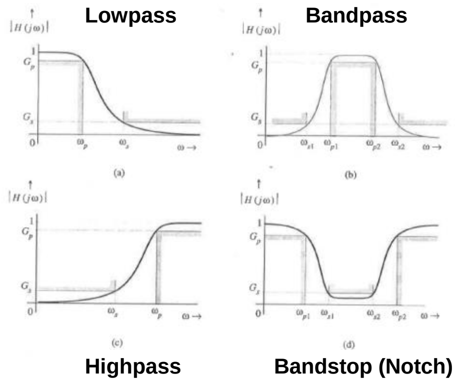


- Lecture 10: Cancelled (Sorry!)
  – We will makeup some of the material today! ☺

Sources: [L] http://www.abc.net.au/news/2017-03-28/cyclone-debbie--space-stations-capture-incredible-images/8392232 [R] Mr. Fausto Benavides

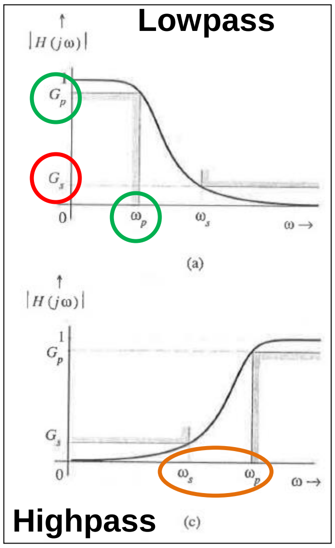
Let's Start With: (analog) Filters!

Filters



- *Frequency-shaping filters*: LTI systems that change the shape of the spectrum
- *Frequency-selective filters:* Systems that pass some frequencies undistorted and attenuate others

Filters



Specified Values:

- Gp = minimum passband gain

Typically:

$$G_p = \frac{1}{\sqrt{2}} = -3dB$$

- Gs = maximum stopband gain
  - **Low**, not zero (sorry!)
  - For realizable filters, the gain cannot be zero over a finite band (Paley-Wiener condition)
- Transition Band:

  transition from the passband to the stopband ➔ ωp≠ ωs

Filter Design & z-Transform

Filter Type	Mapping	Design Parameters
Low-pass	$z^{-1} \to \dfrac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$	$\alpha = \dfrac{\sin[(\omega_c - \omega_c')/2]}{\sin[(\omega_c + \omega_c')/2]}$ $\omega_c' =$ desired cutoff frequency
High-pass	$z^{-1} \to -\dfrac{z^{-1} + \alpha}{1 + \alpha z^{-1}}$	$\alpha = -\dfrac{\cos[(\omega_c + \omega_c')/2]}{\cos[(\omega_c - \omega_c')/2]}$ $\omega_c' =$ desired cutoff frequency
Bandpass	$z^{-1} \to -\dfrac{z^{-2} - [2\alpha\beta/(\beta+1)]z^{-1} + [(\beta-1)/(\beta+1)]}{[(\beta-1)/(\beta+1)]z^{-2} - [2\alpha\beta/(\beta+1)]z^{-1} + 1}$	$\alpha = \dfrac{\cos[(\omega_{c2} + \omega_{c1})/2]}{\cos[(\omega_{c2} - \omega_{c1})/2]}$ $\beta = \cot[(\omega_{c2} - \omega_{c1})/2]\tan(\omega_c/2)$ $\omega_{c1} =$ desired lower cutoff frequency $\omega_{c2} =$ desired upper cutoff frequency
Bandstop	$z^{-1} \to \dfrac{z^{-2} - [2\alpha/(\beta+1)]z^{-1} + [(1-\beta)/(1+\beta)]}{[(1-\beta)/(1+\beta)]z^{-2} - [2\alpha/(\beta+1)]z^{-1} + 1}$	$\alpha = \dfrac{\cos[(\omega_{c1} + \omega_{c2})/2]}{\cos[(\omega_{c1} - \omega_{c2})/2]}$ $\beta = \tan[(\omega_{c2} - \omega_{c1})/2]\tan(\omega_c/2)$ $\omega_{c1} =$ desired lower cutoff frequency $\omega_{c2} =$ desired upper cutoff frequency

Butterworth Filters

- Butterworth: Smooth in the pass-band
- The amplitude response $|H(j\omega)|$ of an n[th] order Butterworth low pass filter is given by:

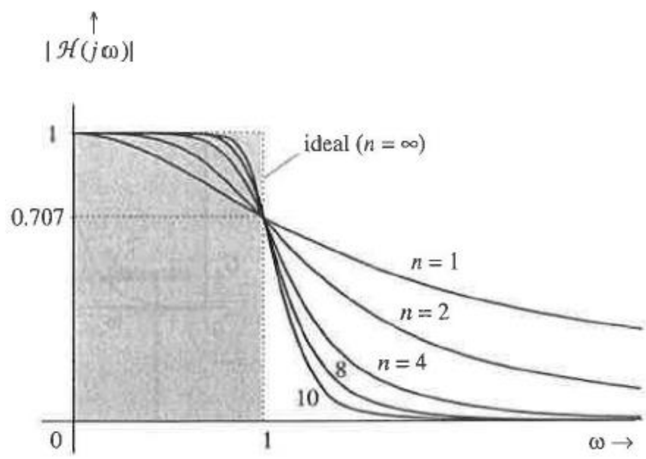
$$|H(j\omega)| = \frac{1}{\sqrt{1+\left(\frac{\omega}{\omega_c}\right)^{2n}}}$$

- The normalized case (ω_c=1)

$$|\mathcal{H}(j\omega)| = \frac{1}{\sqrt{1+\omega^{2n}}} \Rightarrow \mathcal{H}(j\omega)\mathcal{H}(-j\omega) = |\mathcal{H}(j\omega)|^2 = \frac{1}{1+\omega^{2n}}$$

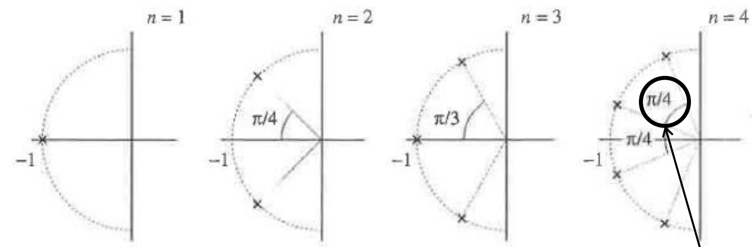
Recall that: $|H(j\omega)|^2 = H(j\omega)\,H(-j\omega)$

Butterworth Filters



Butterworth Filters of Increasing Order: Seeing this Using a Pole-Zero Diagram

- Increasing the order, increases the number of poles:

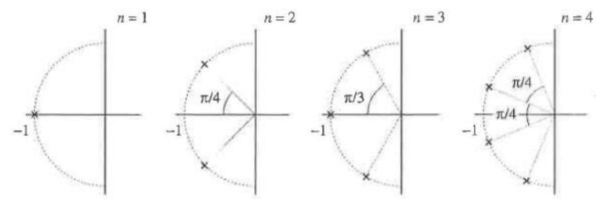


Angle between poles: $\dfrac{\pi}{n}$

➔Odd orders (n=1,3,5…):
- Have a pole on the Real Axis

➔Even orders (n=2,4,6…):
- Have a pole on the off axis

Butterworth Filters: Pole-Zero Diagram



- Since H(s) is stable and causal, its poles must lie in the LHP
- Poles of -H(s) are those in the RHP
- Poles lie on the unit circle (for a normalized filter)

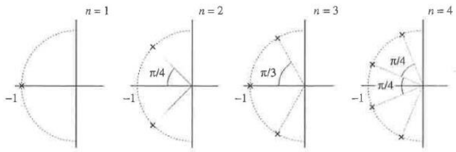
n is the order of the filter

➔$H(s) = \dfrac{1}{(s-s_1)(s-s_2)\dots(s-s_n)}$

Where:

$$s_k = e^{\frac{j\pi}{2n}(2k+n-1)}$$
$$= \cos\frac{\pi}{2n}(2k+n-1) + j\sin\frac{\pi}{2n}(2k+n-1) \qquad k = 1, 2, 3, \dots, n$$

Butterworth Filters: 4^{th} Order Filter Example



- Plugging in for n=4, k=1,…4:

$$\mathcal{H}(s) = \frac{1}{(s+0.3827-j0.9239)(s+0.3827+j0.9239)(s+0.9239-j0.3827)(s+0.9239+j0.3827)}$$
$$= \frac{1}{(s^2+0.7654s+1)(s^2+1.8478s+1)}$$
$$= \frac{1}{s^4+2.6131s^3+3.4142s^2+2.6131s+1}$$

- We can generalize → Butterworth Table

n	a_1	a_2	a_3	a_4	a_5
2	1.41421356				
3	2.00000000	2.00000000			
4	2.61312593	3.41421356	2.61312593		
5	3.23606798	5.23606798	5.23606798	3.23606798	
6	3.86370331	7.46410162	9.14162017	7.46410162	3.86370331

This is for 3dB bandwidth at ω_c=1

Butterworth Filters: Scaling Back (from Normalized)

- Start with Normalized equation & Table
- Replace ω with $\frac{\omega}{\omega_c}$ in the filter equation
- For example:
  for f_c=100Hz → ω_c=200π rad/sec

  From the Butterworth table: for n=2, $a_1=\sqrt{2}$
  Thus:

$$H(s) = \frac{1}{\left(\frac{s}{200\pi}\right)^2+\sqrt{2}\left(\frac{s}{200\pi}\right)+1}$$
$$= \frac{1}{s^2+200\pi\sqrt{2}+40{,}000\pi^2}$$

Butterworth: Determination of Filter Order

- Define $\mathbf{G_x}$ as the gain of a lowpass Butterworth filter at $\omega = \omega_x$
- Then:

$$\hat{G}_x = 20\log_{10}|H(j\omega_x)| = -10\log\left[1+\left(\frac{\omega_x}{\omega_c}\right)^{2n}\right]$$

And thus:

$$\hat{G}_p = -10\log\left[1+\left(\frac{\omega_p}{\omega_c}\right)^{2n}\right]$$

$$\hat{G}_s = -10\log\left[1+\left(\frac{\omega_s}{\omega_c}\right)^{2n}\right]$$

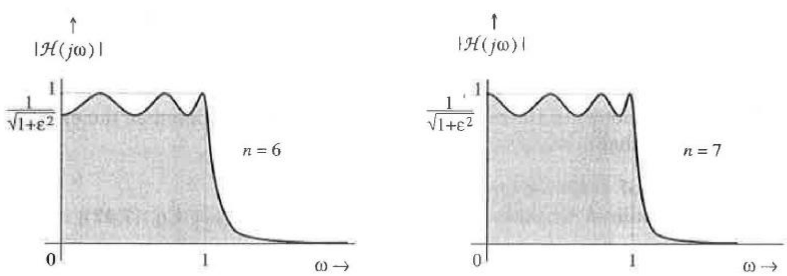
Or alternatively: $\omega_c = \dfrac{\omega_p}{\left[10^{-\hat{G}_p/10}-1\right]^{1/2n}}$ & $\omega_c = \dfrac{\omega_s}{\left[10^{-\hat{G}_s/10}-1\right]^{1/2n}}$

Solving for n gives:

$$n = \frac{\log\left[\left(10^{-\hat{G}_s/10}-1\right)/\left(10^{-\hat{G}_p/10}-1\right)\right]}{2\log(\omega_s/\omega_p)}$$

PS. See Lathi 4.10 (p. 453) for an example in MATLAB

Chebyshev Filters



- **equal-ripple:**
  Because all the ripples in the passband are of equal height
- If we reduce the ripple, the passband behaviour improves, but it does so at the cost of stopband behaviour

Chebyshev Filters

- Chebyshev Filters: Provide tighter transition bands (sharper cutoff) than the same-order Butterworth filter, but this is achieved at the expense of inferior passband behavior (rippling)

➔ For the lowpass (LP) case: at higher frequencies (in the stopband), the Chebyshev filter gain is smaller than the comparable Butterworth filter gain by about <u>***6(n - 1) dB***</u>

- The amplitude response of a normalized Chebyshev lowpass filter is:

$$|\mathcal{H}(j\omega)| = \frac{1}{\sqrt{1+\epsilon^2 {C_n}^2(\omega)}}$$

Where Cn(ω), the nth-order Chebyshev polynomial, is given by:

$$C_n(\omega) = \cos\left(n \cos^{-1} \omega\right)$$

$$C_n(\omega) = \cosh\left(n \cosh^{-1} \omega\right)$$

and where C_n is given by:

n	$C_n(\omega)$
0	1
1	ω
2	$2\omega^2 - 1$
3	$4\omega^3 - 3\omega$
4	$8\omega^4 - 8\omega^2 + 1$
5	$16\omega^5 - 20\omega^3 + 5\omega$
6	$32\omega^6 - 48\omega^4 + 18\omega^2 - 1$

Normalized Chebyshev Properties

- It's normalized: The passband is 0<ω<1
- **Amplitude response**: has **ripples** in the passband and is **smooth** (monotonic) in the stopband
- **Number of ripples**: there is a total of ***n*** maxima and minima over the passband 0<ω<1
- $C_n^2(0) = \begin{cases} 0, & n: odd \\ 1, & n: even \end{cases}$ ➔ $|H(0)| = \begin{cases} 1, & n: odd \\ \frac{1}{\sqrt{1+\epsilon^2}}, & n: even \end{cases}$
- ϵ: ripple height → $r = \sqrt{1+\epsilon^2}$
- The Amplitude at ω=1: $\frac{1}{r} = \frac{1}{\sqrt{1+\epsilon^2}}$
- For Chebyshev filters, the ripple ***r*** dB takes the place of G_p

Determination of Filter Order

- The gain is given by: $\hat{G} = -10 \log \left[1 + \epsilon^2 {C_n}^2(\omega)\right]$

Thus, the gain at ω_s is: $\epsilon^2 {C_n}^2(\omega_s) = 10^{-\hat{G}_s/10} - 1$

- Solving:

$$n = \frac{1}{\cosh^{-1}(\omega_s)} \cosh^{-1} \left[\frac{10^{-\hat{G}_s/10} - 1}{10^{\hat{r}/10} - 1} \right]^{1/2}$$

- General Case:

$$n = \frac{1}{\cosh^{-1}(\omega_s/\omega_p)} \cosh^{-1} \left[\frac{10^{-\hat{G}_s/10} - 1}{10^{\hat{r}/10} - 1} \right]^{1/2}$$

Chebyshev Pole Zero Diagram

- Whereas Butterworth poles lie on a semi-circle,
  The poles of an n^{th}-order normalized Chebyshev filter lie on a **semiellipse** of the major and minor semiaxes:

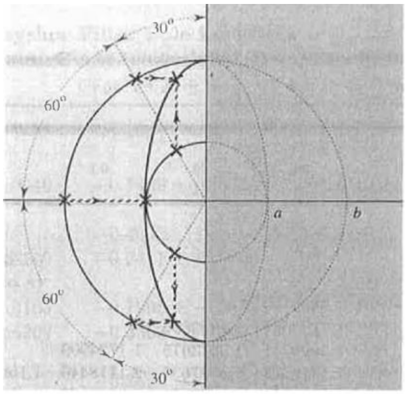
$$a = \sinh\left(\frac{1}{n}\sinh^{-1}\left(\frac{1}{\epsilon}\right)\right) \quad \& \quad b = \cosh\left(\frac{1}{n}\sinh^{-1}\left(\frac{1}{\epsilon}\right)\right)$$

And the poles are at the locations:

$$H(s) = \frac{1}{(s - s_1)(s - s_2)\ldots(s - s_n)}$$

$$s_k = -\sin\left[\frac{(2k-1)\pi}{2n}\right] \sinh x + j \cos\left[\frac{(2k-1)\pi}{2n}\right] \cosh x, \; k = 1, \ldots, n$$

Ex: Chebyshev Pole Zero Diagram for **n=3**



Procedure:

1. Draw two semicircles of radii **a** and **b** (from the previous slide).
2. Draw radial lines along the corresponding Butterworth angles (π/n) and locate the n^{th}-order Butterworth poles (shown by crosses) on the two circles.
3. The location of the k^{th} Chebyshev pole is the intersection of the horizontal projection and the vertical projection from the corresponding k^{th} Butterworth poles on the outer and the inner circle, respectively.

Chebyshev Values / Table

$$\mathcal{H}(s) = \frac{K_n}{C'_n(s)} = \frac{K_n}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}$$

$$K_n = \begin{cases} a_0 & n \text{ odd} \\ \dfrac{a_0}{\sqrt{1+\epsilon^2}} = \dfrac{a_0}{10^{\hat{r}/20}} & n \text{ even} \end{cases}$$

n	a_0	a_1	a_2	a_3
1	1.9652267			
2	1.1025103	1.0977343		
3	0.4913067	1.2384092	0.9883412	
4	0.2756276	0.7426194	1.4539248	0.9528114

1 db ripple
($\hat{r} = 1$)

Other Filter Types:
Chebyshev Type II = Inverse Chebyshev Filters

- Chebyshev filters passband has ripples and the stopband is smooth.
- **Instead:** this has **passband** have **smooth** response and **ripples** in the stopband.

➔ Exhibits maximally flat passband response and equi-ripple stopband

➔ `Cheby2` **in MATLAB**

$$|\mathcal{H}(\omega)|^2 = 1 - |\mathcal{H}_C(1/\omega)|^2 = \frac{\epsilon^2 C_n^2(1/\omega)}{1+\epsilon^2 C_n^2(1/\omega)}$$

Where: H_c is the Chebyshev filter system from before

- Passband behavior, especially for small ω, is **better** than Chebyshev
- **Smallest transition band** of the 3 filters (Butter, Cheby, Cheby2)
- Less time-delay (or phase loss) than that of the **Chebyshev**
- Both needs the **same order *n*** to meet a set of specifications.
- **$$$** (or number of elements):
  Cheby < Inverse Chebyshev < Butterworth (of the same **performance** [not order])

Other Filter Types:
Elliptic Filters (or Cauer) Filters

- Allow **ripple** in **both** the passband and the stopband,
  ➔ we can achieve **tighter** transition band

$$|\mathcal{H}(j\omega)| = \frac{1}{\sqrt{1+\epsilon^2 {R_n}^2(\omega)}}$$

Where: R_n is the n^{th}-order Chebyshev rational function determined from a given ripple spec.
ϵ controls the ripple
$Gp = \frac{1}{\sqrt{1+\varepsilon^2}}$

- Most efficient (η)
  - the **largest ratio** of the passband gain to stopband gain
  - **or** for a given ratio of passband to stopband gain, it requires the **smallest transition band**

➔ **in MATLAB:** `ellipord` followed by `ellip`

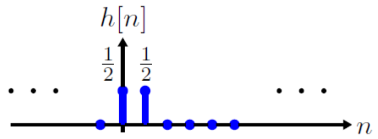
In Summary

Filter Type	Passband Ripple	Stopband Ripple	Transition Band	MATLAB Design Command
Butterworth	No	No	Loose	`butter`
Chebyshev	Yes	No	Tight	`cheby`
Chebyshev Type II (Inverse Chebyshev)	No	Yes	Tight	`cheby2`
Eliptic	Yes	Yes	Tightest	`ellip`

Almost there: (digital) **Signal Types!**

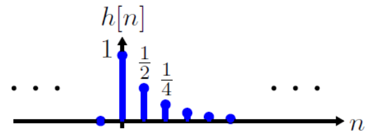
Impulse Response of Both Types

$$y[n] = \frac{1}{2}u[n-1] + \frac{1}{2}u[n]$$



'Finite impulse response" (FIR)

$$y[n] = \frac{1}{2}y[n-1] + u[n]$$



"Infinite impulse response" (IIR)

→ Digital Filters Types

FIR

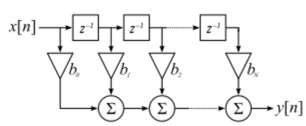
From H(z):

→ $$\begin{aligned} H(\omega) &= h_0 + h_1 e^{-\mathrm{i}\omega} + \cdots + h_{n-1} e^{-\mathrm{i}(n-1)\omega} \\ &= \sum_{t=0}^{n-1} h_t \cos t\omega - \mathrm{i} \sum_{t=0}^{n-1} h_t \sin t\omega \end{aligned}$$

→ Filter becomes a "multiply, accumulate, and delay" system:

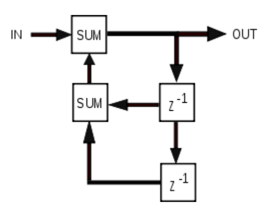
$$y(t) = \sum_{\tau=0}^{n-1} h_\tau u(t-\tau)$$

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N]$$



IIR

- Impulse response function that is non-zero over an infinite length of time.



FIR Properties

- Require no feedback.
- Are inherently stable.
- They can easily be designed to be linear phase by making the coefficient sequence symmetric
- Flexibility in shaping their magnitude response
- Very Fast Implementation (based around FFTs)

- The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency (relative to the sample rate) cutoffs are needed.

FIR as a class of LTI Filters

- Transfer function of the filter is

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

- Finite Impulse Response (FIR) Filters: (N = 0, no feedback)

➔From H(z):

$$\begin{aligned} H(\omega) &= h_0 + h_1 e^{-i\omega} + \cdots + h_{n-1} e^{-i(n-1)\omega} \\ &= \sum_{t=0}^{n-1} h_t \cos t\omega - i \sum_{t=0}^{n-1} h_t \sin t\omega \end{aligned}$$

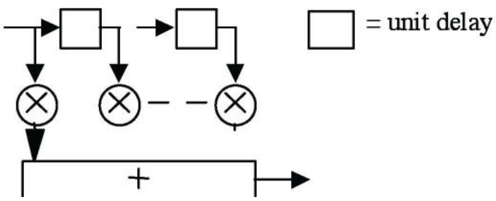
∵ H(ω) is periodic and conjugate

∴ <u>Consider ω ∈ [0, π]</u>

FIR Filters

- Let us consider an FIR filter of length M
- Order N=M-1 **(watch out!)**
- Order → number of delays

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) = \sum_{k=0}^{M-1} h(k)\, x(n-k)$$



FIR Impulse Response

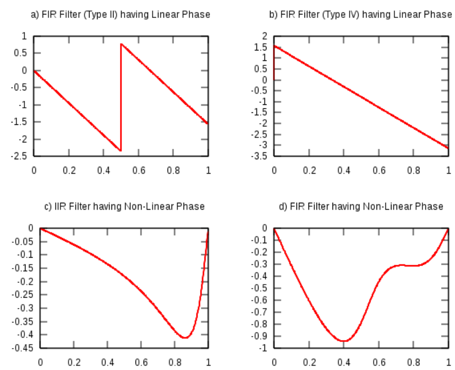
Obtain the impulse response immediately with $x(n)= \delta(n)$:

$$h(n) = y(n) = \sum_{k=0}^{M-1} b_k \delta(n-k) = b_n$$

- The impulse response is of finite length M (good!)
- FIR filters have only zeros (no poles) (as they must, N=0 !!)
  - Hence known also as **all-zero** filters
- FIR filters also known as **feedforward** or **non-recursive,** or **transversal** filters

FIR & Linear Phase

- The phase response of the filter is a linear function of frequency



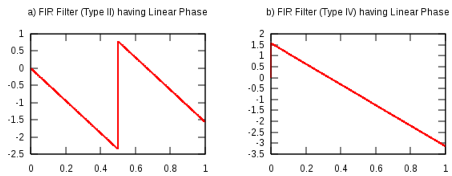
Ref: Wikipedia (Linear Phase)

- Linear phase has constant group delay, all frequency components have equal delay times. ∴ No distortion due to different time delays of different frequencies

- FIR Filters with:

$$\sum_{n=-\infty}^{\infty} h[n] \cdot \sin(\omega \cdot (n - \alpha) + \beta) = 0$$

FIR & Linear Phase → Four Types



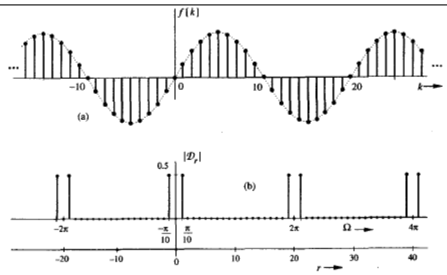
Ref: Wikipedia (Linear Phase)

Impulse response	# coefs	$H(\omega)$	Type
$h(n) = h(M-1-n)$	Odd	$e^{-j\omega(M-1)/2}\left(h\left(\frac{M-1}{2}\right) + 2\sum_{k=1}^{(M-3)/2} h\left(\frac{M-1}{2}-k\right)\cos(\omega k)\right)$	1
$h(n) = h(M-1-n)$	Even	$e^{-j\omega(M-1)/2} 2\sum_{k=1}^{(M-3)/2} h\left(\frac{M}{2}-k\right)\cos\left(\omega\left(k-\frac{1}{2}\right)\right)$	2
$h(n) = -h(M-1-n)$	Odd	$e^{-j[\omega(M-1)/2-\pi/2]}\left(2\sum_{k=1}^{(M-1)/2} h\left(\frac{M-1}{2}-k\right)\sin(\omega k)\right)$	3
$h(n) = -h(M-1-n)$	Even	$e^{-j[\omega(M-1)/2-\pi/2]} 2\sum_{k=1}^{(M-1)/2} h\left(\frac{M}{2}-k\right)\sin\left(\omega\left(k-\frac{1}{2}\right)\right)$	4

- Type 1: most versatile
- Type 2: frequency response is always 0 at ω=π
  (not suitable as a high-pass)
- Type 3 and 4: introduce a π/2 phase shift, 0 at ω=0
  (not suitable as a high-pass)

DTFT

Digital Filters → DTFT



Lathi, p. 621

- First Thought:

FFT → Crop → Un-FFT → Go!

- How to get DTFT? FFT?
- Slightly Naïve ∵
  - ○ H(ω) cannot be exactly zero over any *band* of frequencies (**Paley-Wiener Theorem)**

DTFT is a Convolution

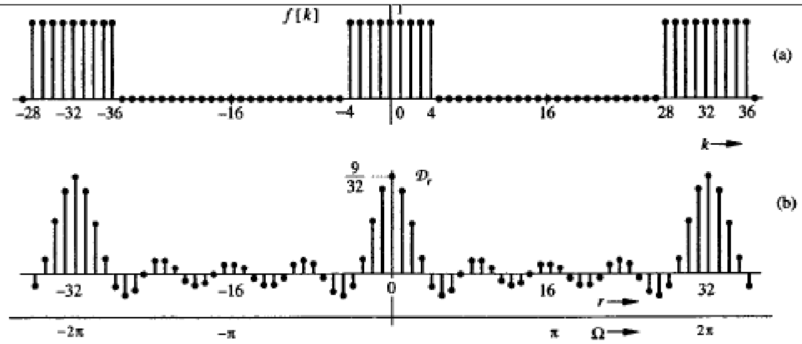


Fig. 10.2 Periodic sampled gate pulse and its Fourier spectrum. Lathi, p. 623

- The frequency response is limited to 2π
- DTFT is a convolution responses in time domain…

$$\underbrace{\mathcal{F}\{x * h\}}_{Y(\omega)} = \underbrace{\mathcal{F}\{x\}}_{X(\omega)} \cdot \underbrace{\mathcal{F}\{h\}}_{H(\omega)}$$

$$y[n] = x[n] * h[n] = \mathcal{F}^{-1}\{X(\omega) \cdot H(\omega)\},$$

DTFT ➔ z-Transform

The above results motivate the definitions of the z transform, the discrete-time Fourier transform (DTFT), and the discrete Fourier series (DFS) to be presented in this chapter and the next. In particular, if the basis functions for the input can be enumerated as

$$\phi_k[n] = z_k^n,$$

that is, if $x(t)$ can be expressed in the form of Eq. (6.1.1) as

$$x[n] = \sum_k a_k z_k^n, \tag{6.1.10}$$

then the corresponding output is simply, from Eqs. (6.1.2) and (6.1.8),

$$y[n] = \sum_k a_k H(z_k) z_k^n. \tag{6.1.11}$$

The discrete Fourier series for periodic signals is of this form, with $z_k = e^{j2\pi k/N}$. If, on the other hand, the required basis functions cannot be enumerated, we must utilize the continuum of functions $\phi[n] = z^n$ to represent $x[n]$ and $y[n]$ in the form of integrals. When z is restricted to have unit magnitude (that is, $z = e^{j\Omega}$), the resulting representation is called the *discrete-time Fourier transform*, while if z is an arbitrary complex variable, the full *z-transform* representation results.

The Discrete-Time Fourier Transform

- Synthesis:

The function $X(e^{j\Omega})$ defined by

$$\boxed{X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}} \qquad (7.1.1)$$

(if it converges) is called the *discrete-time Fourier transform* (*DTFT*) of the signal $x[n]$. In particular, if the region of convergence for the z transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

includes the unit circle, then the DTFT equals $X(z)$ evaluated on the unit circle, that is,

$$X(e^{j\Omega}) = X(z)\big|_{z=e^{j\Omega}}. \qquad (7.1.2)$$

The Discrete-Time Fourier Transform

- Analysis/Inverse:

$$\boxed{x[n] = \frac{1}{2\pi}\int_{2\pi} X(e^{j\Omega})e^{j\Omega n}\,d\Omega.}$$

- *x[n]* is the (limiting) sum of sinusoidal components of the form $\left[\frac{1}{2\pi}X\left(e^{j\Omega}\right)d\Omega\right] e^{j\Omega n}$

- Together: Forms the DTFT Pair

The Discrete-Time Fourier Transform

- Ex:

$$x[n] = a^n u[n]$$

has the z transform

$$X(z) = \frac{1}{1 - az^{-1}}, \qquad |z| > |a|,$$

and thus $X(e^{j\Omega})$ exists for $|a| < 1$ because the ROC then contains the unit circle. Specifically,

$$X(e^{j\Omega}) = \frac{1}{1 - ae^{-j\Omega}}, \qquad |a| < 1. \qquad (7.1.8)$$

The corresponding *magnitude spectrum* $|X(e^{j\Omega})|$ and *phase spectrum* $\angle X(e^{j\Omega})$ are shown in Fig. 6.8. Clearly, from the defining sum in Eq. (7.1.1), the DTFT of $x[n]$ does not converge for $|a| > 1$, and we defer until later the case of $|a| = 1$.

On the other hand, the anticausal exponential

$$w[n] = -a^n u[-n-1]$$

has the z transform

$$W(z) = \frac{1}{1 - az^{-1}}, \qquad |z| < |a|,$$

and thus $W(e^{j\Omega})$ exists for $|a| > 1$, but not for $|a| < 1$. That is,

$$W(e^{j\Omega}) = \frac{1}{1 - ae^{-j\Omega}}, \qquad |a| > 1. \qquad (7.1.9)$$

Again the case of $|a| = 1$ is deferred until later.

The Discrete-Time Fourier Transform

- Observe:
  "Kinship Of Difference Equations To Differential Equations"

$$\frac{dy}{dt} + cy(t) = x(t) \qquad (3.15a)$$

Consider uniform samples of $x(t)$ at intervals of T seconds. As usual, we use the notation $x[n]$ to denote $x(nT)$, the nth sample of $x(t)$. Similarly, $y[n]$ denotes $y[nT]$, the nth sample of $y(t)$. From the basic definition of a derivative, we can express Eq. (3.15a) at $t = nT$ as

$$\lim_{T \to 0} \frac{y[n] - y[n-1]}{T} + cy[n] = x[n]$$

Clearing the fractions and rearranging the terms yields (assuming nonzero, but very small T)

$$y[n] + \alpha y[n-1] = \beta x[n] \qquad (3.15b)$$

where

$$\alpha = \frac{-1}{1 + cT} \quad \text{and} \quad \beta = \frac{T}{1 + cT}$$

We can also express Eq. (3.15b) in advance operator form as

$$y[n+1] + \alpha y[n] = \beta x[n+1] \qquad (3.15c)$$

The Discrete-Time Fourier Transform

- Ex(2): The DTFT of the real sinusoid

$$x[n] = \sin \Omega_0 n = \frac{1}{2j}(e^{j\Omega_0 n} - e^{-j\Omega_0 n})$$

is simply

$$X(e^{j\Omega}) = 2\pi\left(\frac{1}{2j}\right)[\delta(\Omega - \Omega_0) - \delta(\Omega + \Omega_0)]$$

$$= -j\pi[\delta(\Omega - \Omega_0) - \delta(\Omega + \Omega_0)]$$

for $|\Omega|, |\Omega_0| \le \pi$, while that of the cosine signal

$$y[n] = \cos \Omega_0 n = \tfrac{1}{2}(e^{j\Omega_0 n} + e^{-j\Omega_0 n})$$

is likewise

$$Y(e^{j\Omega}) = 2\pi(\tfrac{1}{2})[\delta(\Omega - \Omega_0) + \delta(\Omega + \Omega_0)]$$

$$= \pi[\delta(\Omega - \Omega_0) + \delta(\Omega + \Omega_0)].$$

In addition, the DTFT pair for the dc signal $x[n] = 1$ is simply

$$1 \leftrightarrow 2\pi\,\delta(\Omega), \qquad |\Omega| \le \pi,$$

as opposed to the dual relationship

$$\delta[n] \leftrightarrow 1, \qquad \text{all } \Omega.$$

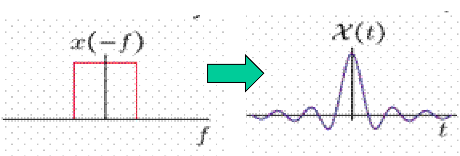
BREAK

Now: (digital) **Filters!**

Flashback: Fourier Series & Rectangular Functions

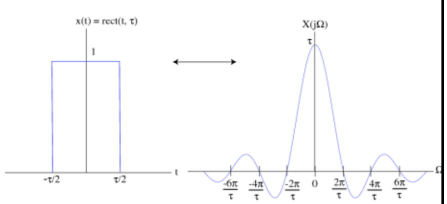
$\mathfrak{F}$: Fourier Tranform

$$\mathfrak{F}^{-1}\left\{rect\left(\frac{\omega}{2}\right)\right\} = \frac{sinc\,(t)}{\pi}$$



Ref: http://cnx.org/content/m26719/1.1/
http://www.wolframalpha.com/input/?i=IFFT%28sinc%28f%29%29

$$\mathfrak{F}\{rect\,(t)\} = sinc\left(\frac{\omega}{2}\right)$$



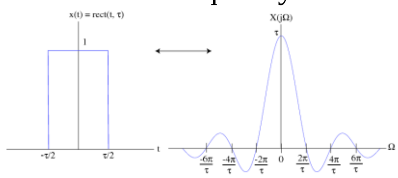
Ref: http://cnx.org/content/m32899/1.8/
http://www.thefouriertransform.com/pairs/box.php

See:

- Table 7.1 (p. 702) Entry 17 & Table 9.1 (p. 852) Entry 7

Flashback: Fourier Series & Rectangular Functions [2]

- The sinc function might look familiar
  - This is the frequency content of a square wave (box)



Ref: http://www.wolframalpha.com/input/?i=FFT%28rect%28t%29%29
http://cnx.org/content/m32899/1.8/

- This also applies to **signal reconstruction!**

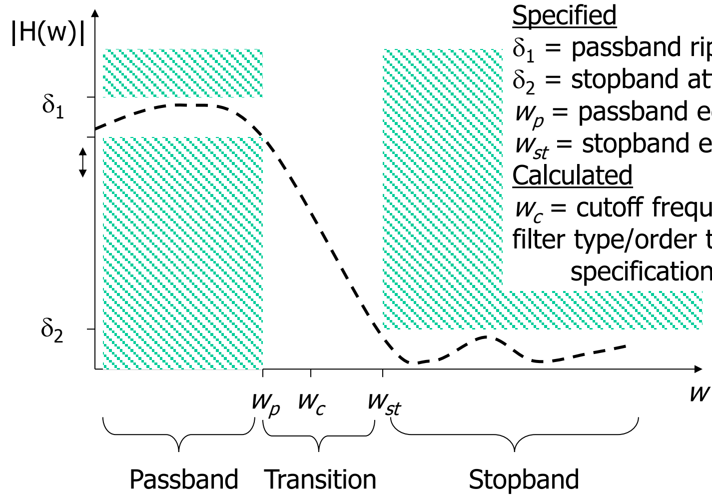
➔ Whittaker–Shannon interpolation formula
  - This says that the "better way" to go from Discrete to Continuous (i.e. D to A) is not ZOH, but rather via the sinc!

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \cdot \mathrm{sinc}\left(\frac{t-nT}{T}\right)$$

Filter Design

- Previously we have analysed
  - difference equations (y[n])
  - transfer functions (H(z))
- To obtain time/frequency domain response
  - Impulse (h[n]) or frequency (H(w)) response
- Now we have a specification
  - frequency response (filters)
  - time response (control)
- Goal to design a filter that meets specification
  - i.e., determine transfer function
  - and therefore difference equation (implementation)

Filter Specification in the Frequency Domain



Specified
δ_1 = passband ripple (dB)
δ_2 = stopband attenuation (dB)
w_p = passband edge (Hz)
w_{st} = stopband edge (Hz)
Calculated
w_c = cutoff frequency (@ 3dB)
filter type/order to meet specification

Transfer Function → Difference Equation

- Example, consider

$$H(z)=\frac{z^2-0.2z-0.08}{z^2+0.5}$$

Make H(*z*) causal × by $\frac{z^{-2}}{z^{-2}}$

- Normalise to negative powers of z (causal)
  - re-arrange and take inverse z transform

$$H(z)=\frac{1-0.2z^{-1}-0.08z^{-2}}{1+0.5z^{-2}}=\frac{Y(z)}{X(z)}$$

$$Y(z)\left(1+0.5z^{-2}\right)=X(z)\left(1-0.2z^{-1}-0.08z^{-2}\right)$$

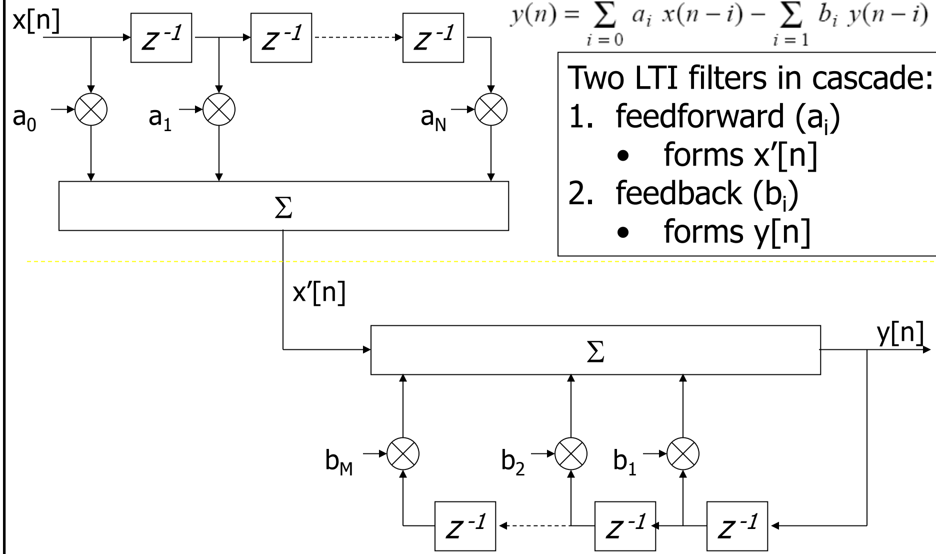
$$y[n]+0.5y[n-2]=x[n]-0.2x[n-1]-0.08x[n-2]$$

$$y[n]=x[n]-0.2x[n-1]-0.08x[n-2]-0.5y[n-2]$$

Direct Form I: Direct realisation of digital filter

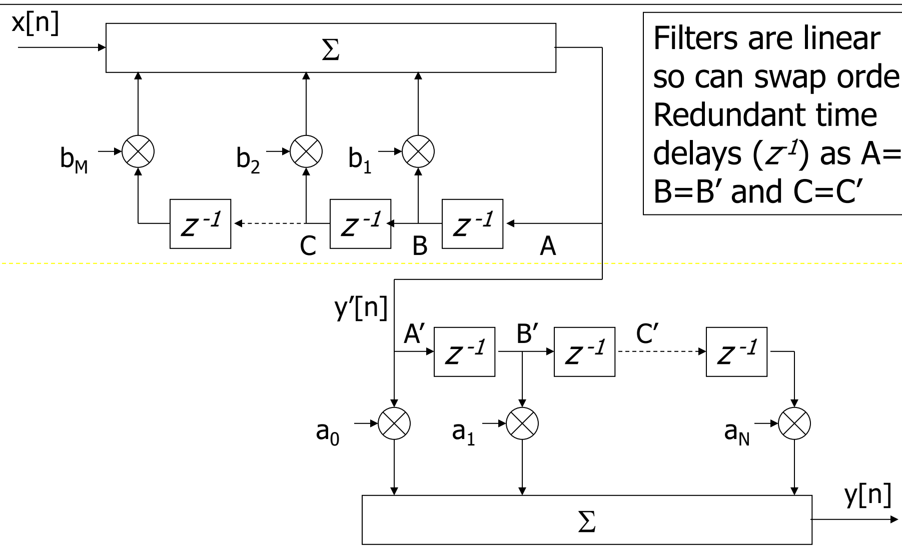
$$y(n) = \sum_{i=0}^{N} a_i\, x(n-i) \overset{+}{-} \sum_{i=1}^{M} b_i\, y(n-i)$$

Two LTI filters in cascade:
1. feedforward (a_i)
   - forms x′[n]
2. feedback (b_i)
   - forms y[n]



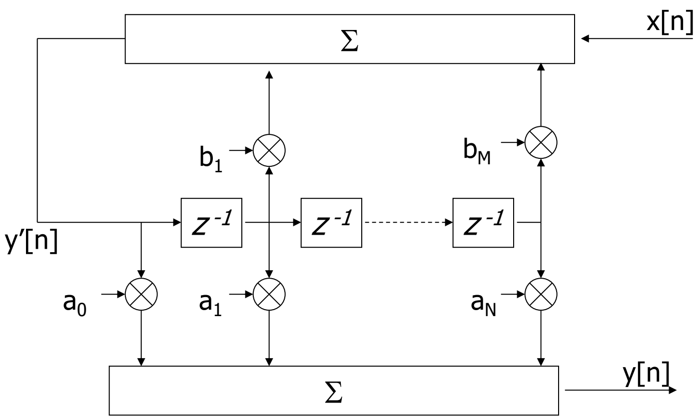
Reordered form of realisation

Filters are linear so can swap order. Redundant time delays (z^{-1}) as A=A′ B=B′ and C=C′



Note: y′[n] ≠ x′[n] of previous slide BUT y[n] = y[n] ☺ so, same filter

Direct form II: Canonical form of realisation (minimum memory)



redundant time delays removed

Derivation of Canonical Form

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N} a_i z^{-i}}{(1 - \sum_{i=1}^{M} b_i z^{-i})}$$

General form of transfer function

$$Y(z) = H(z)\, X(z)$$

Re-arranging in terms of output

$$Y(z) = \sum_{i=0}^{N} a_i z^{-i}\, Y'(z) \qquad \text{where} \qquad Y'(z) = \frac{X(z)}{(1 - \sum_{i=1}^{M} b_i z^{-i})}$$

Which as a difference equation is

Direct II $\quad y(n) = \sum_{i=0}^{N} a_i\, y'(n-i) \quad$ where $\quad y'(n) = x(n) + \sum_{i=1}^{M} b_i\, y'(n-i),$

Remember

Direct I $\quad y(n) = \sum_{i=0}^{N} a_i\, x(n-i) + \sum_{i=1}^{M} b_i\, y(n-i)$

Canonical terms
A' B' C'

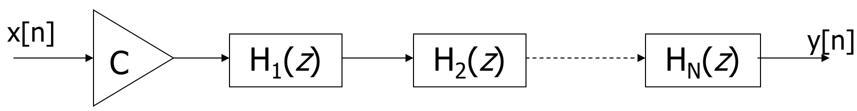
Canonical Realisation

- Direct Form I
  - Conceptually simplest realisation
  - Often less susceptible to noise
- Canonical/Direct Form II
  - Minimimum memory (storage)
- Filter design
  - Determine value of filter coefficients (all ai & bi)
  - Poles controlled by bi coefficients
    - if any bi ≠ 0 then filter IIR (recursive)
    - if all bi = 0 then filter FIR (non-recursive)
  - Zeros controlled by ai coefficients

Cascade Form

- Transfer function factorised to
  - Product of second order terms Hn(z)
  - C is a constant (gain)

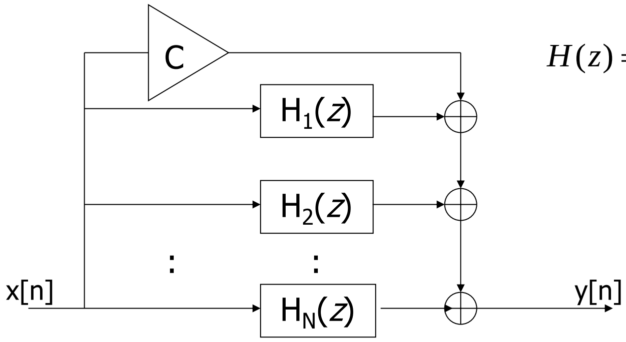
$$H(z) = C\prod_{n=1}^{N} H_n(z)$$



Most common realisation
Often assumed by many filter design packages
many 2nd order sections have integer coefficients

Parallel Form

- Transfer function expressed as
  - partial fraction expansion of second order terms

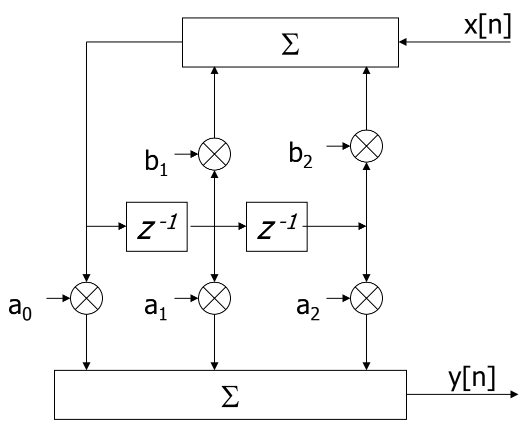


$$H(z) = C + \sum_{n=1}^{N} H_n(z)$$

Least sensitive to coefficient errors, i.e., when limited No. bits to represent real ($\Re$) coefficient

Bi-quadratic Digital Filter

- Canonic form of Second order system
- 2nd order, system 'building block'



Difference equation:

$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + b_1 y[n-1] + b_2 y[n-2]$$

IIR Filter Design Methods

- Normally based on analogue prototypes
  - Butterworth, Chebyshev, Elliptic etc
- Then transform $H(s) \rightarrow H(z)$
- Three popular methods:
- Impulse invariant
  - produces H(z) whose impulse response is a sampled version of h(t) (also step invariant)
- Matched z – transform
  - poles/zeros H(s) directly mapped to poles/zeros H(z)
- Bilinear z – transform
  - left hand s – plane mapped to unit circle in z - plane

Impulse Invariant

- Simplest approach, proceeds as follows,
- Select prototype analogue filter
- Determine H(s) for desired wc and ws
- Inverse Laplace,
  - i.e., calculate impulse response, h(t)
- Sample impulse response $h(t)|_{t=n\Delta td}$
  - $h[n] = \Delta td\ h(n\Delta td)$
- Take z - transform of $h[n] \Rightarrow H(z)$
  - poles, p1 map to $\exp(p1\Delta td)$ (maintains stability)
  - zeros have no simple mapping

Impulse Invariant

- Useful approach when
  - Impulse (or step) invariance is required, or
    - e.g., control applications
  - Designing Lowpass or Bandpass filters
- Has problems when
  - H(w) does not $\rightarrow 0$ as w $\rightarrow \infty$
  - i.e., if H(w) is not bandlimited, aliasing occurs
  - e.g., highpass or bandstop filters

Matched z - transform

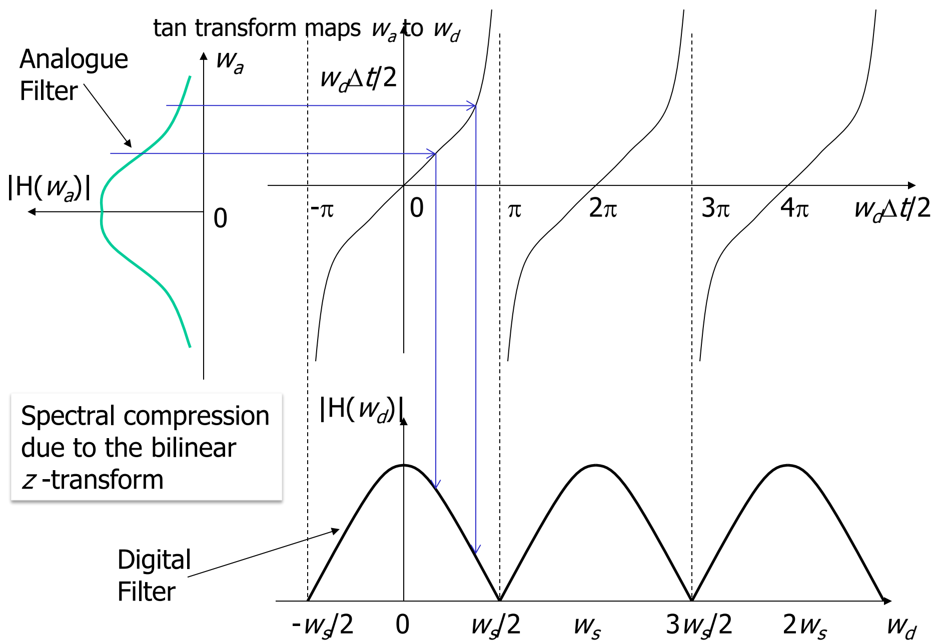
- Maps poles/zeros in s – plane directly
  - to poles/zeros in z – plane
- No great virtues/problems
- Fairly old method
  - not commonly used
  - so we won't consider it further

Bilinear z - transform

- Maps complete imaginary s –plane (±∞)
  - to unit circle in z -plane
- i.e., maps analogue frequency wa to
  - discrete frequency wd
- uses continuous transform,

$$w_a = \frac{2}{\Delta t}\tan\left(\frac{w_d \Delta t}{2}\right)$$

This compresses (warps) w_a to have finite extent $\pm w_s/2$
i.e., this removes possibility of any aliasing ☺



Note, H(w_d) periodic, due to sampling

Bilinear Transform

$$\omega_a = \frac{2}{\Delta t}\tan\left(\frac{\omega_d \Delta t}{2}\right)$$

The bilinear transform

$$s = \frac{2}{\Delta t}\frac{j\sin\left(\frac{\omega_d \Delta t}{2}\right)}{\cos\left(\frac{\omega_d \Delta t}{2}\right)}$$

Transforming to s-domain
Remember: $s = j\omega_a$
and $\tan\theta = \sin\theta/\cos\theta$
Where $\theta = \omega_d \Delta t/2$

$$s = \frac{2}{\Delta t}\frac{\frac{1}{2}\left(\exp\left(\frac{j\omega_d \Delta t}{2}\right) - \exp\left(\frac{-j\omega_d \Delta t}{2}\right)\right)}{\frac{1}{2}\left(\exp\left(\frac{j\omega_d \Delta t}{2}\right) + \exp\left(\frac{-j\omega_d \Delta t}{2}\right)\right)}$$

Using Euler's relation
This becomes…
(note: j terms cancel)

$$s = \frac{2}{\Delta t}\frac{(1 - \exp(-j\omega_d \Delta t))}{(1 + \exp(-j\omega_d \Delta t))}$$

Multiply by $\exp(-j\theta)/\exp(-j\theta)$

$$s = \frac{2(1 - z^{-1})}{\Delta t(1 + z^{-1})}$$

As $z = \exp(s_d \Delta t) = \exp(j\omega_d \Delta t)$

Bilinear Transform

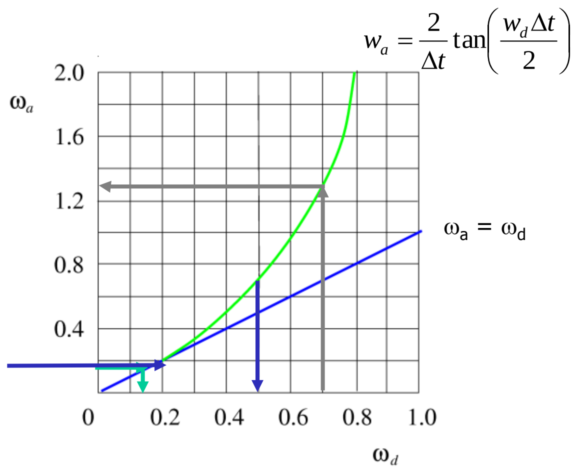
- Convert H(s) ⇒ H(z) by substituting,

$$s = \frac{2\left(1 - z^{-1}\right)}{\Delta t\left(1 + z^{-1}\right)}$$

- However, this transformation compresses the frequency response, which means
  - digital cut off frequency will be lower than the analogue prototype
- Therefore, analogue filter must be "pre-warped" prior to transforming H(s) ⇒ H(z)

Note: this comes directly from **tan** transform

Bilinear Pre-warping



Bilinear Transform: Example

- Design digital Butterworth lowpass filter
  - order, n = 2, cut off frequency wd = 628 rad/s
  - sampling frequency ws = 5024 rad/s (800Hz)
- pre-warp to find wa that gives desired wd

$$w_a = \left(\frac{2}{1/800}\right)\tan\left(\frac{628}{2\times 800}\right) = 663\,\text{rad/s}$$

Note: $w_d < w_a$ due to compression

- Butterworth prototype (unity cut off) is,

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

Bilinear Transform: Example

- De-normalised analogue prototype (s' = s/ ω_c)
  - $\omega_c = 663\ rad/s$ (required ω_a to give desired)

$$H(s_d) = \frac{1}{\left(\frac{s}{663}\right)^2 + \frac{\sqrt{2}s}{663} + 1}$$

  - Convert H(s) ⇒ H(z) by substituting

$$s = \frac{2\left(1-z^{-1}\right)}{\Delta t\left(1+z^{-1}\right)}$$

$$H(z) = \frac{1}{\left(\frac{2\times 800(1-z^{-1})}{663(1+z^{-1})}\right)^2 + \sqrt{2}\left(\frac{2\times 800(1-z^{-1})}{663(1+z^{-1})}\right) + 1}$$

$$H(z) = \frac{0.098z^2 + 0.195z + 0.098}{z^2 - 0.942z + 0.333}$$

Note: H(z) has both poles and zeros H(s) was all-pole

Bilinear Transform: Example

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.098z^2 + 0.195z + 0.098}{z^2 - 0.942z + 0.333}$$

- Multiply out and make causal:

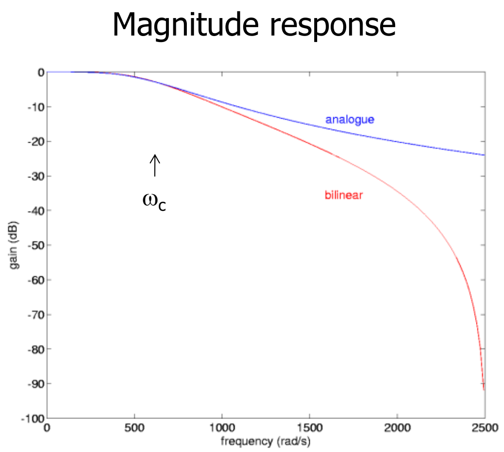
$$Y(z)(z^2 - 0.942z + 0.333) = X(z)(0.098z^2 + 0.195z + 0.098)$$

$$Y(z)(1 - 0.942z^{-1} + 0.333z^{-2}) = X(z)(0.098 + 0.195z^{-1} + 0.098z^{-2})$$

Finally, apply inverse z-transform to yield the difference equation:

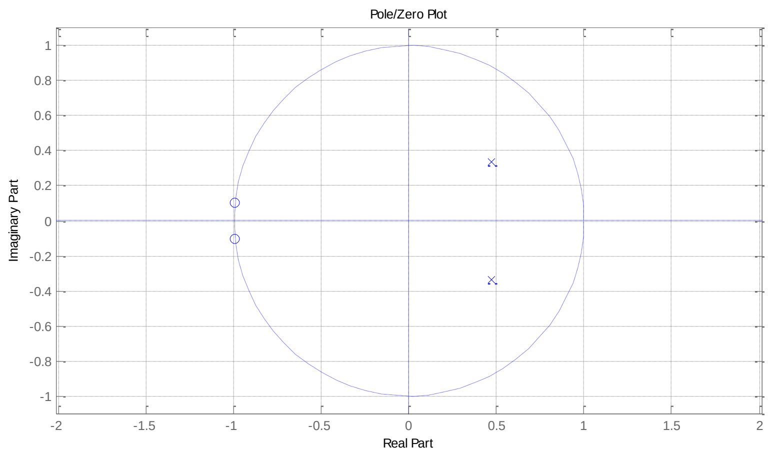
$$y[n] = 0.098x[n] + 0.195x[n-1] + 0.098x[n-2] \\ + 0.942y[n-1] - 0.333y[n-2]$$

Bilinear Transform: Example



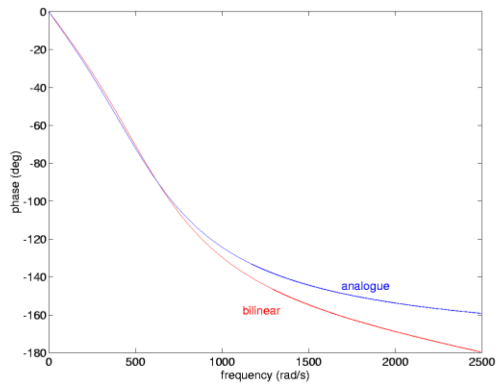
1. same cut off frequency,
2. increased roll off and attenuation in stopband
3. ∞ attenuation at $w_s/2$

Bilinear Transform: Example



Bilinear Transform: Example

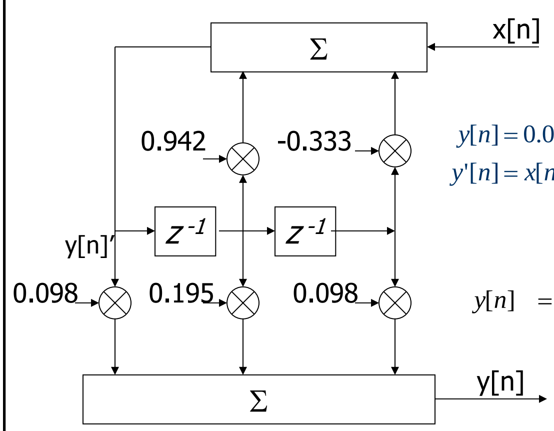
Phase response



Increased phase delay

Bilinear transform has effectively increased digital filter order (by adding zeros)

Bilinear Transform: Example



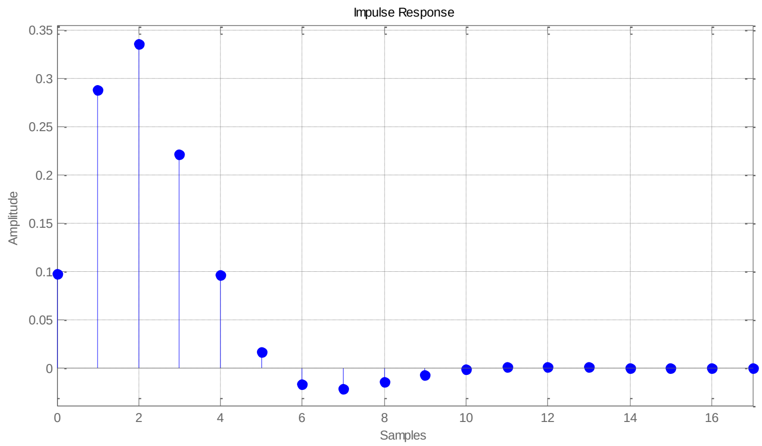
Canonical Implementation

$$y[n] = 0.098y'[n] + 0.195y'[n-1] + 0.098y'[n-2]$$
$$y'[n] = x[n] + 0.942y'[n-1] - 0.333y'[n-2]$$

of the difference equation

$$\begin{aligned} y[n] &= 0.098x[n] + 0.195x[n-1] + 0.098x[n-2] \\ &+ 0.942y[n-1] - 0.333y[n-2] \end{aligned}$$

Bilinear Transform: Example



Bilinear Design Summary

- Calculate pre-warping analogue cutoff frequency
- De-normalise filter transfer function using pre-warping cut-off
- Apply bilinear transform and simplify
- Use inverse z-transform to obtain difference equation

Direct Synthesis

- Not based on analogue prototype
  - But direct placement of poles/zeros
- Useful for
  - First order lowpass or highpass
    - simple smoothers
  - Resonators and equalisers
    - single frequency amplification/removal
  - Comb and notch filters
    - Multiple frequency amplification/removal

First Order Filter: Example

- General first order transfer function
  - Gain, G, zero at –b, pole at a (a, b both < 1)

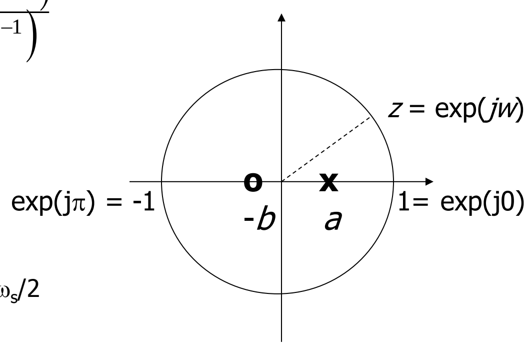
Remember: $H(w) = H(z)|_{z = \exp(jw\Delta t)}$

$$H(z) = \frac{G\left(1 + bz^{-1}\right)}{\left(1 - az^{-1}\right)}$$

with a +ve & b –ve
this is a lowpass filter
i.e.,

$$H(0) = \frac{G(1+b)}{(1-a)}$$

$$H(\pi) = \frac{G(1-b)}{(1+a)} \qquad \omega_s/2$$



First Order Filter: Example

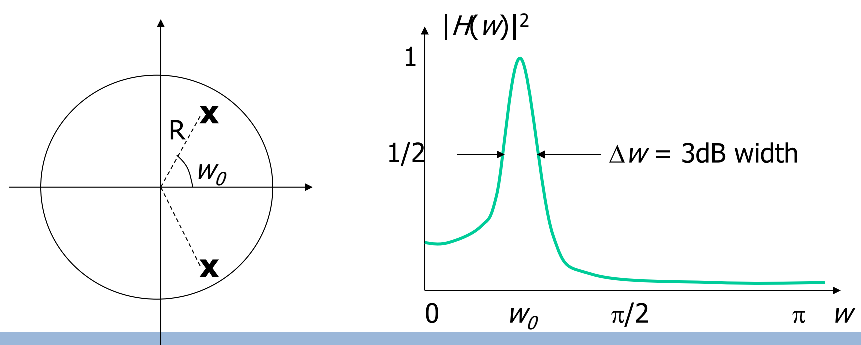
- Possible design criteria
  - cut-off frequency, wc
    - 3dB = 20 log(|H (wc)|)
    - e.g., at wc = π/2, (1+b)/(1+a) = √2
  - stopband attenuation
    - assume wstop = π (Nyquist frequency)
    - e.g., δ2 = H (π)/H (0) = 1/21 i.e.,

$$\frac{H(\pi)}{H(0)} = \frac{(1-b)(1-a)}{(1+b)(1+a)} = \frac{1}{21}$$

two unknowns (*a*,*b*) two (simultaneous) design equations.

Digital Resonator

- Second order ‘resonator’
  - single narrow peak frequency response
  - i.e., peak at resonant frequency, w0



Quality factor (Q-factor)

- Dimensionless parameter that compares
  - Time constant for oscillator decay/bandwidth (Δω) to
  - Oscillation (resonant) period/frequency (ω0)
    - High Q = less energy dissipated per cycle

$$Q = \frac{\omega_0}{\Delta\omega} = \frac{f_0}{\Delta f}$$

- Alternative to damping factor (ζ) as

$$Q = \frac{1}{2\zeta} \qquad H(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} = \frac{\omega_0^2}{s^2 + \frac{\omega_0}{Q} s + \omega_0^2}$$

- Note: Q < ½ overdamped (not an oscillator)

Digital Resonator Design

- To make a peak at w0 place pole
  - Inside unit circle (for stability)
  - At angle w0 distance R from origin
    - i.e., at location p = R exp(jw0)
      - R controls Δw
        - » Closer to unit circle → sharper peak
    - plus complex conj pole at p* = R exp(-jw0)

$$\begin{aligned} H(z) &= \frac{1}{\left(1 - R\cdot\exp(jw_0)z^{-1}\right)\left(1 - R\cdot\exp(-jw_0)z^{-1}\right)} \\ &= \frac{1}{1 - R\left(\exp(jw_0) + \exp(-jw_0)\right)z^{-1} + R^2 z^{-2}} \\ &= \frac{G}{1 + a_1 z^{-1} + a_2 z^{-2}} \end{aligned}$$

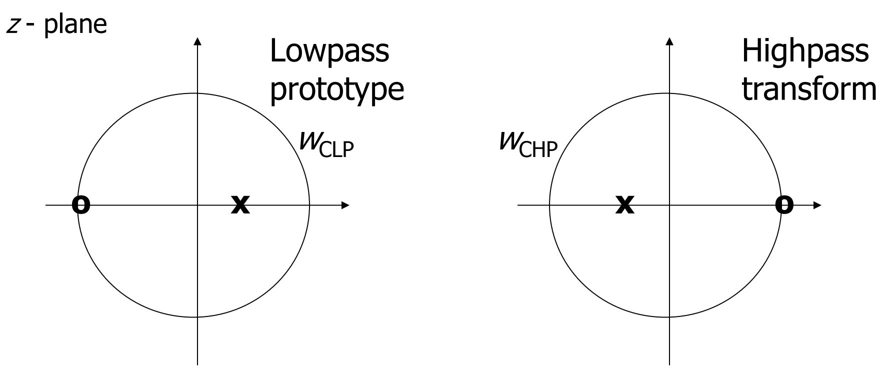
Where (via Euler's relation)

$$a_1 = -2R\cos(w_0) \text{ and } a_2 = R^2$$

Discrete Filter Transformations

- By convention, design Lowpass filters
  - transform to HP, BP, BS, etc
- Simplest transformation
  - Lowpass H(z') → highpass H(z)
  - HHP(z) = HLP(z)|z' → -z
    - reflection about imaginary axis (ws/4)
    - changing signs of poles and zeros
- LP cutoff frequency, wCLP becomes
    - HP cut-in frequency, wCHP = ½ - wCLP

Lowpass → highpass (z' = -z)



p_L = ¼, z_L = -1 p_H = -¼, z_H = 1

Poles/zeros reflected in imaginary axis: w_{CHP} = ½ - w_{CLP}

Same gain @ $w_s/4$ (i.e., $\pi/4$)

$|H(w_{HP})| = |H(\pi/2 - w_{LP})|$

Discrete Filter Transformations

- Lowpass H(z') $\rightarrow$ highpass H(z)
  - Cut-off (3dB) frequency = wc (remains same)
- Lowpass H(z') $\rightarrow$ Bandpass H(z)
  - Centre frequency = w0 & 3dB bandwidth = wc

$$z' = \frac{\cos(w_c \Delta t) - z}{1 - \cos(w_c \Delta t) z}$$

$$z' = \frac{\alpha z - z^2}{-\alpha z + 1} \qquad \alpha = \frac{\cos(w_0 \Delta t)}{\cos(w_c \Delta t)}$$

Note: these are not the only possible BP and BS transformations!

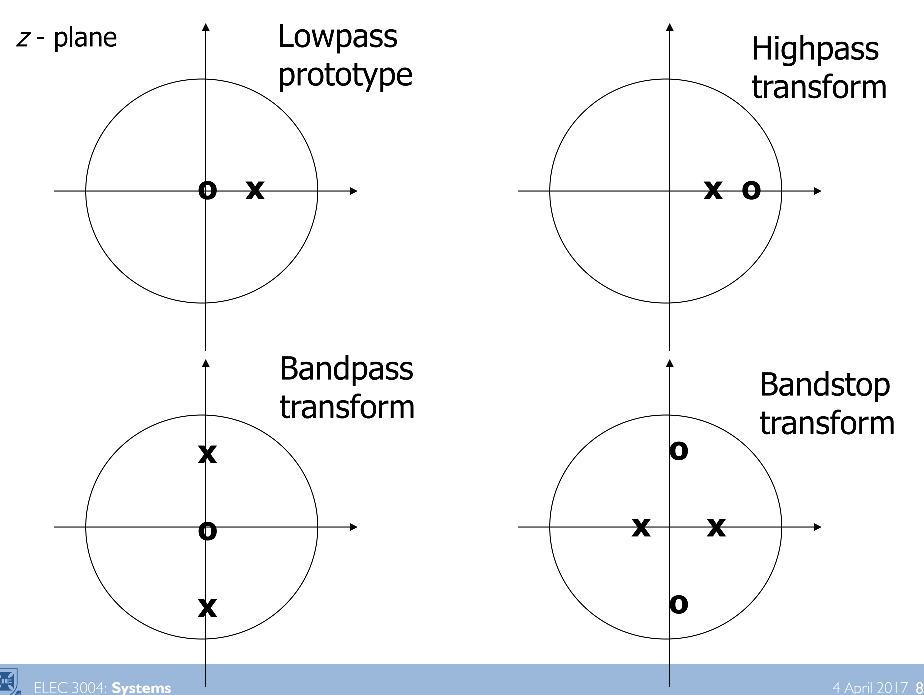
Discrete Filter Transformations

- Lowpass H(z') $\rightarrow$ Bandstop H(z)
  - Centre frequency = w0 3dB bandwidth = wc

$$z' = \frac{z^2 - \left(2\alpha/(k+1)\right)z + (1-k)/(1+k)}{1 + \left(2\alpha/(k+1)\right)z + \left((1-k)/(1+k)\right)z^2}$$

$$\alpha = \frac{\cos(w_0 \Delta t)}{\cos(w_c \Delta t)} \qquad k = \tan^2(w_c \Delta t)$$

Note: order doubles for bandpass/bandstop transformations



Summary

- Digital Filter Structures
  - Direct form (simplest)
  - Canonical form (minimum memory)
- IIR filters
  - Feedback and/or feedforward sections
- FIR filters
  - Feedforward only
- Filter design
  - Bilinear transform (LP, HP, BP, BS filters)
  - Direct form (resonators and notch filters)
  - Filter transformations (LP $\rightarrow$ HP, BP, or BS)
- Stability & Precision improved
  - Using cascade of 1st/2nd order sections

Next Time…

- **Digital Filters**

- Review:
  - Chapter 10 of Lathi

- A signal has many signals ☺
  [Unless it's bandlimited. Then there is the one ω]