

6. Design of IIR Filters

- *Reference:* Sections 7.1 of Text
- A digital filter, $H(e^{j\omega})$, with infinite impulse response (IIR), can be designed by first transforming it into a prototype analog filter $H_c(j\Omega)$ and then design this analog filter using a standard procedure. Once the analog filter is properly designed, it is then mapped back to the discrete-time domain to obtain a digital filter that meets the specifications.

The commonly used analog filters are

1. Butterworth filters – no ripples at all,
2. Chebychev filters - ripples in the passband OR in the stopband, and
3. Elliptical filters - ripples in BOTH the pass and stop bands.

The design of these filters are well documented in the literature.

A disadvantage of IIR filters is that they usually have nonlinear phase. Some minor signal distortion is a result.

- There are two main techniques used to design IIR filters:
 1. The Impulse Invariant method, and
 2. The Bilinear transformation method.

6.1 The Impulse Invariant Method

- In the impulse invariant method, the impulse response of the digital filter, $h[n]$, is made (approximately) equal to the impulse response of an analog filter, $h_c(t)$, evaluated at $t = nT_d$, where T_d is an (arbitrary) sampling period. Specifically

$$h[n] = T_d h_c(nT_d)$$

- From our discussion in Chapter 2,

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} H_c\left(j\frac{\omega}{T_d} + j\frac{2\pi k}{T_d}\right)$$

and aliasing would occur if $H_c(j\Omega)$ is not bandlimited to π/T_d (in rad/s).

If $H_c(j\Omega)$ is bandlimited to π/T_d , then

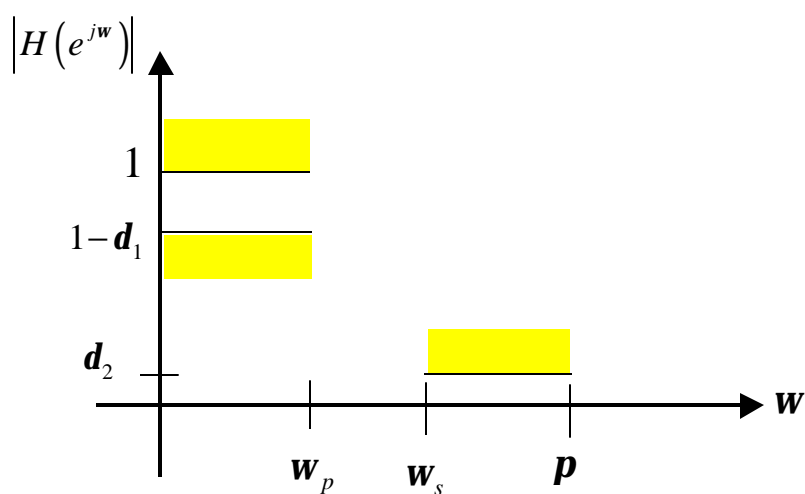
$$H(e^{j\omega}) = H_c(j\omega/T_d).$$

In this case, it is straight forward to specify the prototype analog filter.

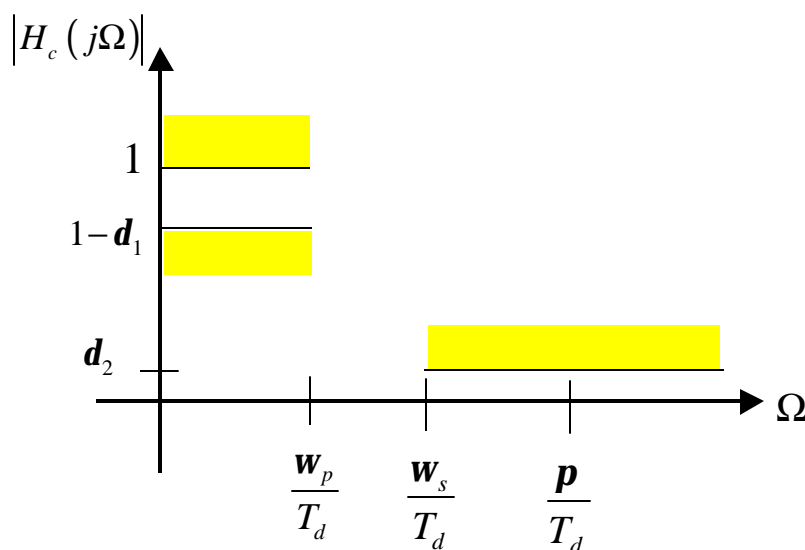
However, all the commonly used prototype analog filters used in the impulse invariant design method are indeed non-bandlimited. So there is aliasing. However, the aliasing can be minimized if we over-design the analog filter (especially in the stop band).

- The picture below illustrates the design procedure. We first specify the digital filter as shown in the first diagram. Then we map the digital

frequency ω onto the analog frequency $\Omega = \omega/T_d$ and make $H_c(j\Omega) = H(e^{j\omega T_d})$. Notice from the diagram that we can only control the magnitude of the responses because of the nature of the analog filters used.



Digital filter specifications



Prototype analog filter

- Let $H_c(s)$ be the Laplace transform of $h_c(t)$, where the complex number

$$s = \mathbf{s} + j\Omega$$

is the Laplace domain variable. Assuming $H_c(s)$ has only single-order poles s_k , then

$$H_c(s) = \sum_{k=1}^N \frac{A_k}{s - s_k}$$

This implies

$$h_c(t) = \begin{cases} \sum_{k=1}^N A_k e^{s_k t} & t \geq 0 \\ 0 & t < 0 \end{cases},$$

$$\begin{aligned} h[n] &= T_d h_c(nT_d) \\ &= T_d \sum_{k=1}^N A_k e^{s_k n T_d} u[n] \\ &= \sum_{k=1}^N A_k T_d \left(e^{s_k T_d} \right)^n u[n] \end{aligned},$$

and

$$\begin{aligned} H(z) &= \sum_{n=-\infty}^{\infty} h[n] z^{-n} \\ &= \sum_{n=0}^{\infty} \sum_{k=1}^N A_k T_d \left(e^{s_k T_d} \right)^n z^{-n} \\ &= \sum_{k=1}^N A_k T_d \sum_{n=0}^{\infty} \left(z^{-1} e^{s_k T_d} \right)^n \\ &= \sum_{k=1}^N \frac{A_k T_d}{1 - z^{-1} e^{s_k T_d}} \end{aligned}$$

- It is observed that the pole $s_k = \mathbf{s}_k + j\Omega_k$ in the s -plane is mapped into the pole

$$\begin{aligned} p_k &= e^{s_k T_d} \\ &= e^{\mathbf{s}_k T_d} e^{j\Omega_k T_d} \end{aligned}$$

in the z -plane. If all the s_k 's are on the left half of the s -plane, i.e. $\mathbf{s}_k < 0, k=1,2,\dots, N$, then all the p_k 's are within the unit-circle in the z -plane (i.e. $|p_k| < 1$). This means that a stable analog filter always yields a stable digital filter with the impulse invariant method.

- It should be emphasized that

$$z \neq e^{sT_d}$$

in general.

- **Example:** Design a digital low pass IIR filter with the following specifications:

$$\begin{aligned} 0.89125 \leq |H(e^{j\omega})| \leq 1, & \quad 0 \leq \omega \leq 0.2\mathbf{p} \\ |H(e^{j\omega})| \leq 0.17783, & \quad 0.3\mathbf{p} \leq \omega \leq \mathbf{p} \end{aligned}$$

Use the impulse invariant technique and an analog Butterworth filter. Assume $T_d = 1$ and that there is minimal aliasing.

Solution

- with $T_d = 1$, this means $\Omega = \omega$, and $H_c(j\Omega) = H(e^{j\Omega})$. Consequently, the specifications of the analog prototype filter are

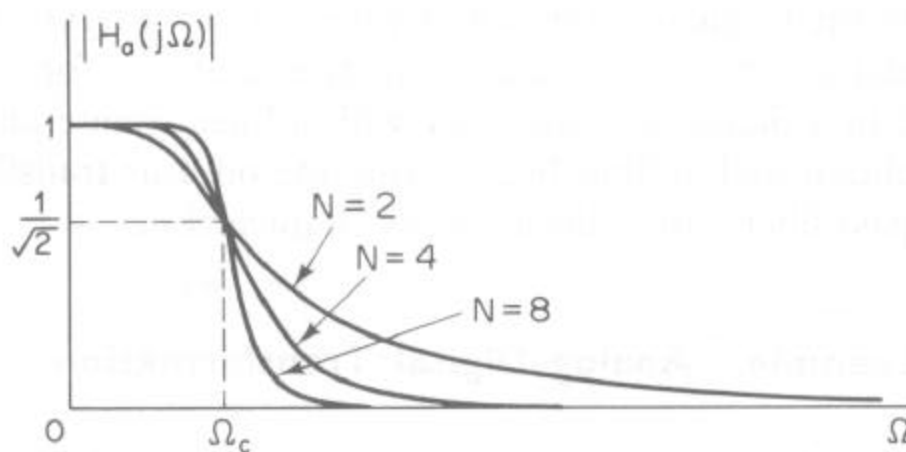
$$\begin{aligned} 0.89125 \leq |H_c(j\Omega)| \leq 1, & \quad 0 \leq |\Omega| \leq 0.2p \\ |H_c(j\Omega)| \leq 0.17783, & \quad 0.3p \leq \Omega \end{aligned}$$

Notice that we did not specify an upper limit for the stopband frequency.

- The magnitude square response of a Butterworth filter of order N is

$$|H_c(j\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}}$$

where Ω_c is the 3-dB frequency of the filter. It is observed that the magnitude square response decreases monotonically with frequency. The larger N is, the closer the Butterworth filter is to an ideal low pass filter.



The term $H_o(j\Omega)$ in this figure is the same as our $H_c(j\Omega)$

- The 3-dB frequency and the filter order N are solutions of the two simultaneous equations:

$$1 + \left(\frac{0.2\mathbf{p}}{\Omega_c} \right)^{2N} = \left(\frac{1}{.89125} \right)^2$$

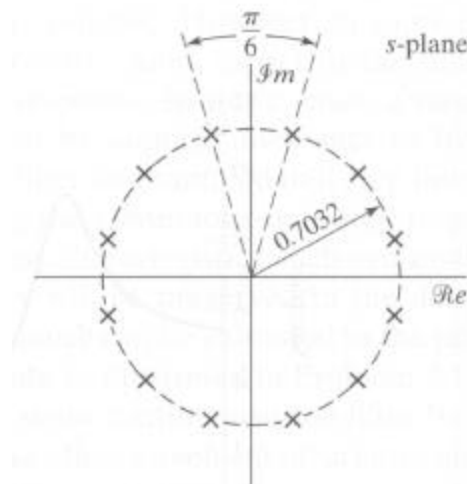
$$1 + \left(\frac{0.3\mathbf{p}}{\Omega_c} \right)^{2N} = \left(\frac{1}{.17783} \right)^2$$

The exact results are $N = 5.8858$ and $\Omega_c = .70474$. After rounding, we have $N = 6$ and $\Omega_c = .7032$. This latter set of results means the passband requirement is met exactly at $\mathbf{w}_p = 0.2\mathbf{p}$ and met with margin at $\mathbf{w}_s = 0.3\mathbf{p}$. Specifically, $|H_c(j0.3\mathbf{p})| = 0.1700 < 0.17783$.

- The next step is to find the poles of the Butterworth filter. Note that

$$H_c(s)H_c(-s) = \frac{1}{\left(1 + \left[\frac{s}{j\Omega_c} \right]^{2N} \right)}$$

has $2N$ poles whose locations in the s -plane are depicted in the diagram below.



Mathematically, these poles are

$$r_k = 0.7032 \exp \left\{ j \left(\frac{\mathbf{P}}{12} + \frac{\mathbf{P}}{2} + (k-1) \frac{2\mathbf{P}}{12} \right) \right\}; \quad k = 1, 2, \dots, 2N$$

Half of these poles are the poles of the Butterworth filter. Specifically we choose those r_k 's on the left-half s -plane to be the poles of $H_c(s)$. Consequently the poles of $H_c(s)$ are

$$\begin{aligned} & -0.182 \pm 0.679 j \\ & -0.497 \pm 0.497 j \\ & -0.679 \pm 0.182 j \end{aligned}$$

The poles of the corresponding digital filter are

$$\begin{aligned} & 0.649 \pm 0.524 j \\ & 0.535 \pm 0.290 j \\ & 0.499 \pm 0.092 j \end{aligned}$$

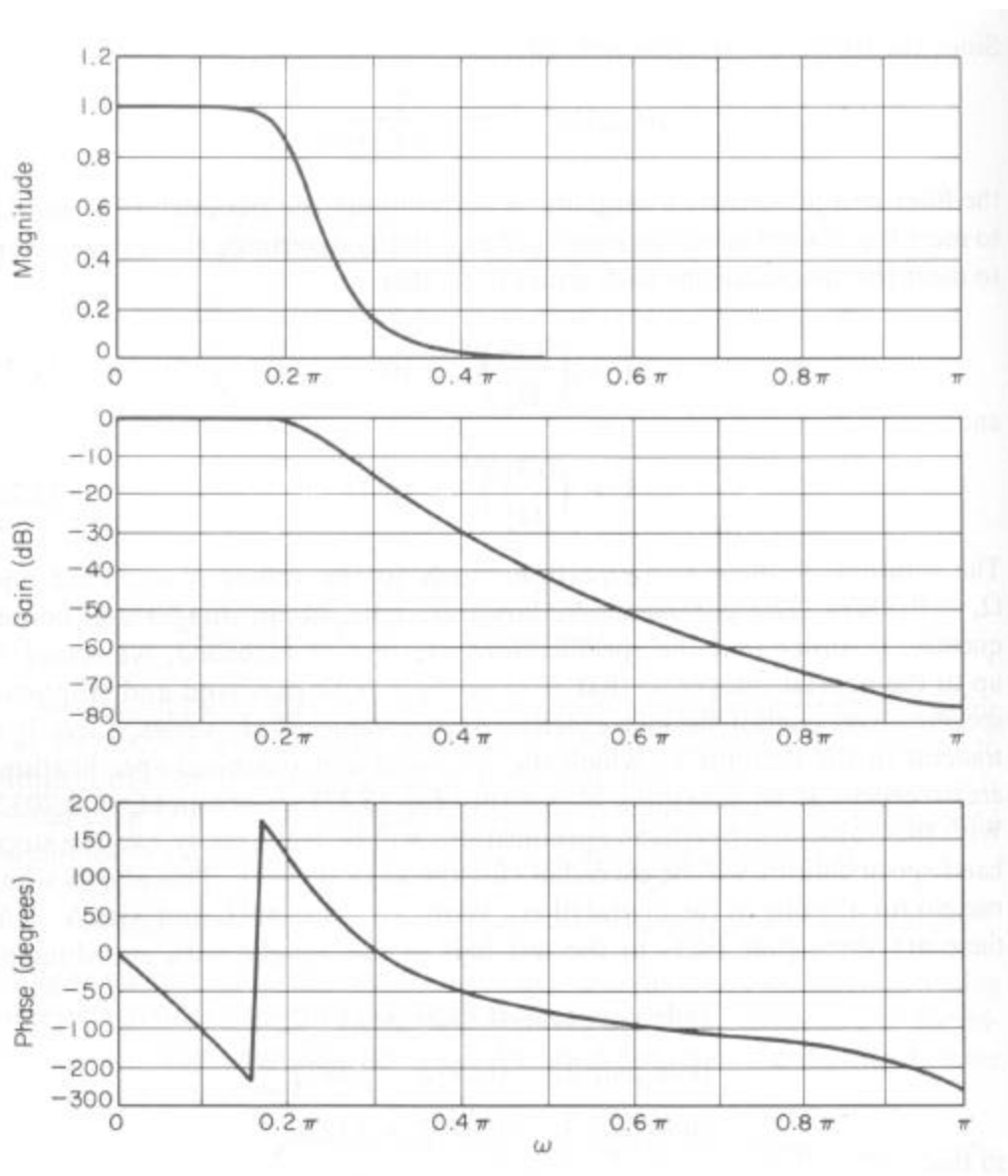
- The transfer function of the Butterworth filter is

$$\begin{aligned} H_c(s) &= \prod_{k=1}^6 \frac{1}{\left(1 - \frac{s}{s_k}\right)} = \frac{\Omega_c^6}{\prod_{k=1}^6 (s - s_k)} \\ &= \frac{0.12093}{(s^2 + 0.3640s + 0.4945)(s^2 + 0.9945s + 0.4945)(s^2 + 1.3585s + 0.4945)} \\ &= \sum_{k=1}^6 \frac{A_k}{s - s_k} \end{aligned}$$

where the A_k s are the partial fraction expansion coefficients.

- The transfer function of the digital filter $H(z)$ is (remember $T_d = 1$)

$$\begin{aligned}
 H(z) &= \sum_{k=1}^N \frac{A_k}{1 - z^{-1} e^{s_k}} \\
 &= \frac{0.2871 - 0.4466z^{-1}}{1 - 1.2971z^{-1} + 0.6949z^{-2}} + \frac{-2.1428 + 1.1455z^{-1}}{1 - 1.0691z^{-1} + 0.3699z^{-2}} \\
 &\quad + \frac{1.8557 - 0.6303z^{-1}}{1 - 0.9972z^{-1} + 0.2570z^{-2}}
 \end{aligned}$$



- If Matlab is used to design the above prototype analog filter, first set

$$W_p = 0.2p = 0.62832$$

$$W_s = 0.3p = 0.94248$$

$$R_p = 20 \log_{10}(0.89125) = -1$$

$$R_s = 20 \log_{10}(0.17783) = -15$$

Then issue the Matlab command

$$[N, W_n] = \text{buttord}(W_p, W_s, R_p, R_s, 's')$$

Matlab will return the filter order in N and its 3db frequency Ω_c in the variable W_n . The results I got is $N=6$, $W_p=0.70866$. The latter result is slightly different from $\Omega_c = 0.7032$.

The poles of the corresponding Butterworth filter can be obtained by issuing the Matlab command

$$[Z, P, K] = \text{butter}(N, W_n, 's')$$

where the array Z contains the zeros, the array P contains all the poles, and the variable K is the gain. The poles I obtained for this example are

$$\begin{aligned} & -6.8451e-001 + 1.8341e-001i \\ & -6.8451e-001 - 1.8341e-001i \\ & -5.0109e-001 + 5.0109e-001i \\ & -5.0109e-001 - 5.0109e-001i \\ & -1.8341e-001 + 6.8451e-001i \\ & -1.8341e-001 - 6.8451e-001i \end{aligned}$$

which are slightly different from those in the text.

Finally use the matlab command ‘impinvar’ to convert the analog prototype filter into a corresponding digital filter.

- **Example:** Repeat the last example using a Chebyshev filter.

- The magnitude square response of a N -th order Chebyshev filter with a ripple parameter of ϵ is

$$|H_c(j\Omega)|^2 = \frac{1}{1 + \epsilon^2 V_N^2\left(\frac{\Omega}{\Omega_c}\right)},$$

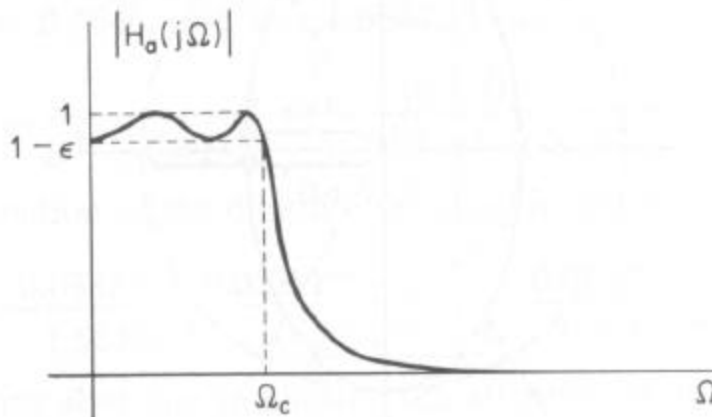
where $V_N(x)$ is the N -th order Chebyshev polynomial, defined as

$$V_N(x) = \cos\left(N \cos^{-1} x\right)$$

For example, $V_0(x) = 1$, $V_1(x) = x$, $V_2(x) = 2x^2 - 1$. In general,

$$V_{N+1}(x) = 2xV_N(x) - V_{N-1}(x)$$

- It is observed that when x is between 0 and unity, $V_N(x)$ varies between 0 and unity. However, when x is greater than 1, $\cos^{-1} x$ is imaginary and so $V_N(x)$ behaves like a hyperbolic cosine and consequently increases monotonically for x greater than unity. Consequently, the magnitude response of a Chebyshev filter looks like the following (**correction:** replace $1 - \epsilon$ by $1/\sqrt{1 + \epsilon^2}$)



- We will design a Chebyshev filter such that the passband requirement is met exactly at $\Omega = 0.2\mathbf{p}$. Consequently, $\Omega_c = 0.2\mathbf{p}$ and

$$0.89125^2 = \frac{1}{1+\mathbf{e}^2}, \quad \text{or} \quad \mathbf{e} = 0.50885.$$

- To determine the filter order N , we calculate $|H_c(0.3\mathbf{p})|^2$ for different values of N and pick the smallest N that exceeds the specification. Note that the requirement is $|H_c(0.3\mathbf{p})| \leq 0.17783$ or $|H_c(0.3\mathbf{p})|^2 \leq 0.031624$. It can be shown that

$$N = 3, \quad |H_c(0.3\mathbf{p})|^2 = 0.04551$$

$$N = 4, \quad |H_c(0.3\mathbf{p})|^2 = 0.00694$$

Consequently the filter order is $N=4$.

- The poles of the Chebyshev filter lie on an ellipse in the s -plane with minor axis $a\Omega_c$ and major axis $b\Omega_c$, where

$$a = \frac{1}{2}(\mathbf{a}^{1/N} - \mathbf{a}^{-1/N}) = 0.3646,$$

$$b = \frac{1}{2}(\mathbf{a}^{1/N} + \mathbf{a}^{-1/N}) = 1.0644,$$

and

$$\mathbf{a} = \mathbf{e}^{-1} + \sqrt{1 + \mathbf{e}^{-2}} = 4.1702.$$

The equation of this ellipse is given by

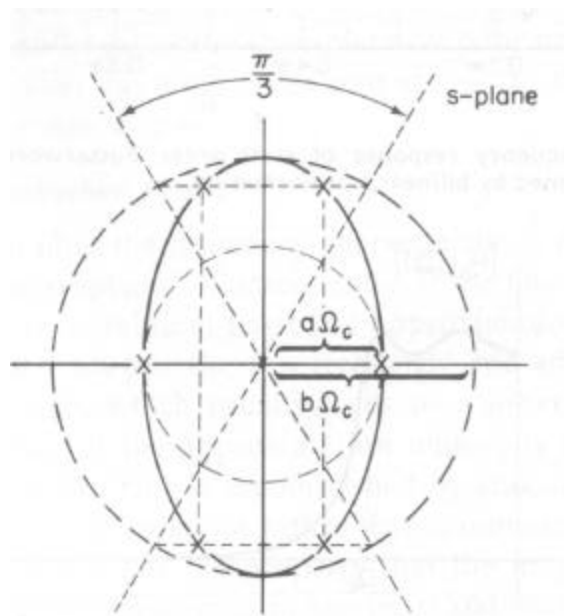
$$\frac{\mathbf{s}^2}{a^2\Omega_c^2} + \frac{\Omega^2}{b^2\Omega_c^2} = 1$$

where $s = \sigma + j\Omega$ is the complex s variable. In addition to the above ellipse, we defined the major circle as the circle centered at $s=0$ and with a radius $b\Omega_c$. Similarly, the minor circle is the circle centered at $s=0$ and with a radius $a\Omega_c$.

To locate the poles, we first identify the points on the major and minor circles that are equally spaced in angle with a spacing of π/N and arranged in such a way that the points are symmetrically located with respect to the imaginary axis (but never fall on the imaginary axis) and a point occurs on the real axis for N odd but not for N even.

The poles of a Chebyshev filter fall on the above ellipse with the ordinate specified by the points identified on the major circle and the abscissa specified by the points identified on the minor circle.

The diagram below illustrates how the poles can be located for the case $N=3$.



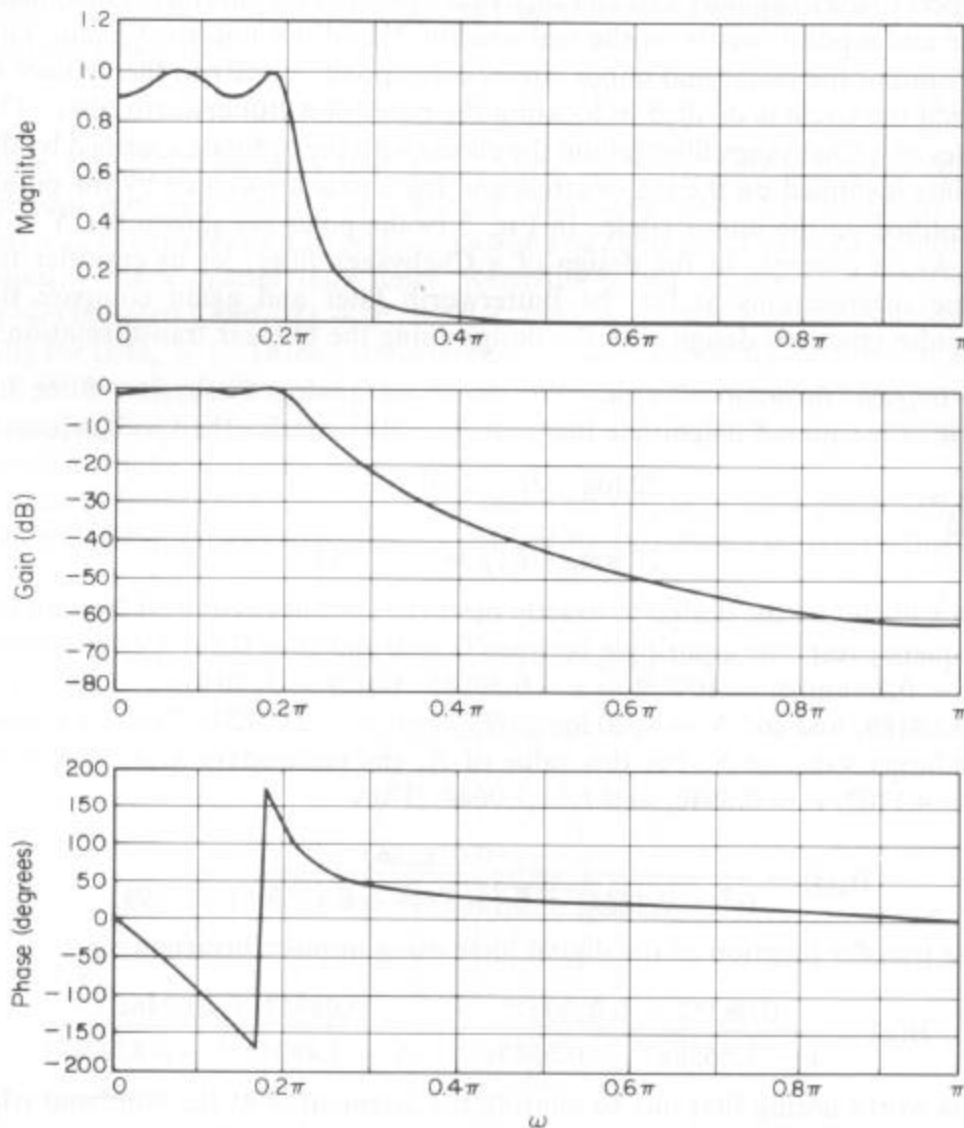
An example for finding the poles of a Chebyshev filter with $N=3$.

It should be emphasized that only poles on the left-half plane are used in the transfer function of the filter.

- Once all the 4 poles, s_1, s_2, s_3, s_4 , are found, the transfer function of the Chebyshev filter can be written as (please verify)

$$H_c(s) = \frac{1}{\sqrt{1+e^2}} \frac{\prod_{k=1}^4 s_k}{\prod_{k=1}^4 (s-s_k)} = \frac{0.038286}{(s^2 + 0.4233s + 0.1103)(s^2 + 0.1753s + 0.3894)}$$

The magnitude and phase of this transfer function are shown below



- If you use Matlab to determine this filter, first set

$$W_p = 0.2\pi = 0.62832$$

$$W_s = 0.3\pi = 0.94248$$

$$R_p = 20 \log_{10}(0.89125) = -1$$

$$R_s = 20 \log_{10}(0.17783) = -15$$

and then issue the Matlab command

$$[N, W_n] = \text{cheb1ord}(W_p, W_s, R_p, R_s, 's')$$

Finally complete the filter design by issuing the command

$$[Z, P, K] = \text{cheby1}(N, -R_p, W_n, 's')$$

The zeros of the filter will be returned in the array Z, the poles stored in P, and the gain stored in K. What I found for the poles were

$$-2.1166e-001 + 2.5593e-001i$$

$$-2.1166e-001 - 2.5593e-001i$$

$$-8.7673e-002 + 6.1788e-001i$$

$$-8.7673e-002 - 6.1788e-001i$$

and the gain was $K = 0.038286$. There are no zeros. These results agree with those obtained through analysis.

- The transfer function of the corresponding digital IIR filter is

$$H(z) = \frac{0.08327 + 0.0239z^{-1}}{1 - 1.5658z^{-1} + 0.6549z^{-2}} + \frac{0.08327 - 0.0246z^{-1}}{1 - 1.4934z^{-1} + 0.8392z^{-2}}$$

- It is interesting to point out the Chebyshev filter has a lower order than the Butterworth filter in the last example.

6.2 The Bilinear Transformation Method

- In the impulse invariant method, aliasing occurs when the prototype analog filter is transformed back into the digital filter. To reduce the distortion introduced by aliasing, we start off by tightening the specifications on the digital filter. This is somewhat cumbersome and may lead to several iterations before the “optimal” filter is found.
- Aliasing occurs because points in the Ω axis separated by $2\mathbf{p}/T_d$ are mapped into the same digital frequency \mathbf{W} . In the Bilinear transformation method, there is a one-to-one correspondence between Ω and \mathbf{W} . So aliasing is avoided in transforming the prototype analog filter back into the digital filter.

Since \mathbf{W} is limited to $[-\mathbf{p}, \mathbf{p}]$ but Ω varies from $-\infty$ to $+\infty$, it becomes clear that Ω must be compressed when it is mapped to \mathbf{W} . In other word, Bilinear transformation is non-linear in nature.

- Let $H_c(s)$ be the transfer function of the prototype analog filter. The transfer function $H(z)$ of the digital filter is obtained by substituting

$$s = \frac{2}{T_d} \frac{1 - z^{-1}}{1 + z^{-1}}$$

into the expression of $H_c(s)$. In other word

$$H(z) = H_c \left(\frac{2}{T_d} \frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

- The Bilinear transformation can be written alternatively as:

$$\begin{aligned}
 (1 + z^{-1}) sT_d &= 2(1 - z^{-1}), \text{ or} \\
 sT_d + z^{-1} sT_d &= 2 - 2z^{-1}, \text{ or} \\
 (2 + sT_d) z^{-1} &= 2 - sT_d, \text{ or} \\
 z &= \frac{2 + sT_d}{2 - sT_d}, \text{ or} \\
 &= \frac{1 + \frac{sT_d}{2}}{1 - \frac{sT_d}{2}}
 \end{aligned}$$

- If $s = \mathbf{s} + j\Omega$, where Ω is the analog frequency, then

$$\begin{aligned}
 z &= \frac{\left(1 + \frac{sT_d}{2}\right) + j \frac{\Omega T_d}{2}}{\left(1 - \frac{sT_d}{2}\right) - j \frac{\Omega T_d}{2}} \\
 &= \frac{\sqrt{\left(1 + \frac{sT_d}{2}\right)^2 + \left(\frac{\Omega T_d}{2}\right)^2} \exp\left\{j \arctan\left[\left(\frac{\Omega T_d}{2}\right) / \left(1 + \frac{sT_d}{2}\right)\right]\right\}}{\sqrt{\left(1 - \frac{sT_d}{2}\right)^2 + \left(\frac{\Omega T_d}{2}\right)^2} \exp\left\{-j \arctan\left[\left(\frac{\Omega T_d}{2}\right) / \left(1 - \frac{sT_d}{2}\right)\right]\right\}} \\
 &= |z| \exp\{j\mathbf{q}\}
 \end{aligned}$$

where

$$|z| = \frac{\sqrt{\left(1 + \frac{sT_d}{2}\right)^2 + \left(\frac{\Omega T_d}{2}\right)^2}}{\sqrt{\left(1 - \frac{sT_d}{2}\right)^2 + \left(\frac{\Omega T_d}{2}\right)^2}}$$

and

$$\mathbf{q} = \arctan\left[\left(\frac{\Omega T_d}{2}\right) / \left(1 + \frac{\mathbf{s} T_d}{2}\right)\right] + \arctan\left[\left(\frac{\Omega T_d}{2}\right) / \left(1 - \frac{\mathbf{s} T_d}{2}\right)\right]$$

are respectively the magnitude and phase of z .

Observations:

1. When $\mathbf{s} < 0$, $|z| < 1$. So left-plane poles of $H_c(s)$ will be mapped into poles within the unit circle in the z -plane. In other word, a stable prototype analog filter will lead to a stable digital filter.
2. When $\mathbf{s} = 0$, $|z| = 1$. In other word, z lies on the unit circle and can be written as

$$z = e^{j\mathbf{w}} \quad (\text{when } \mathbf{s} = 0)$$

where

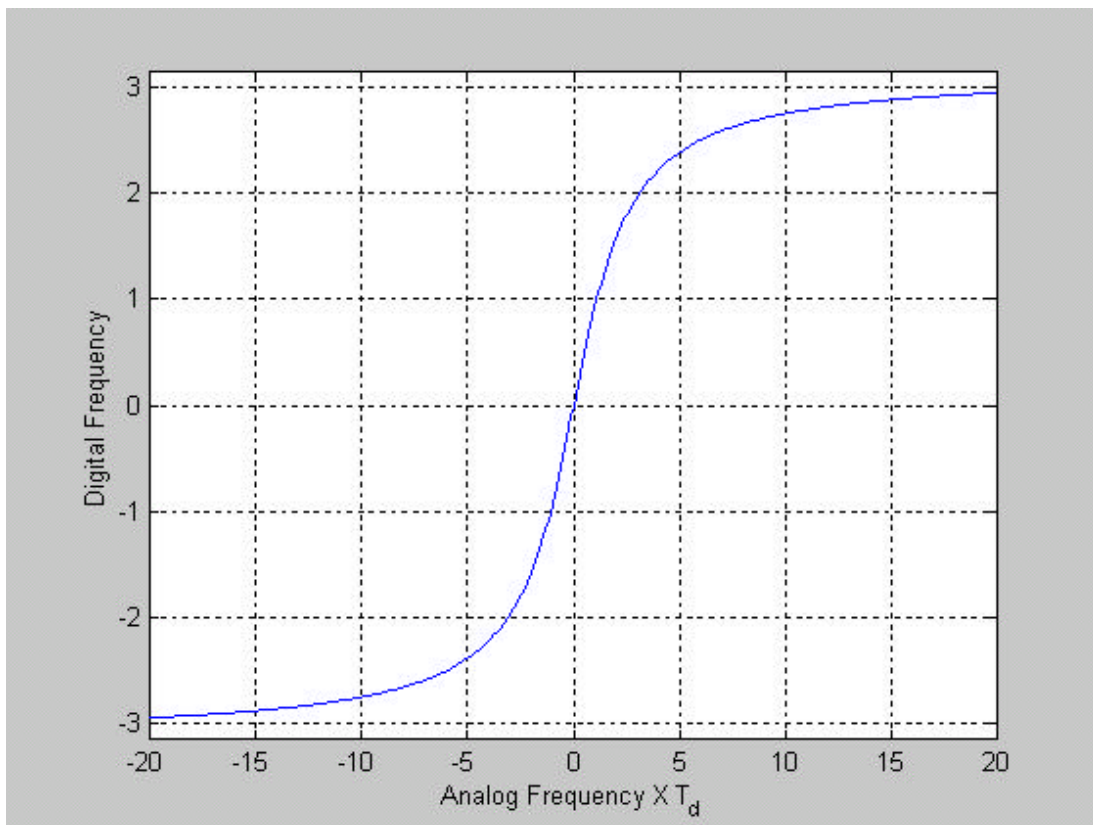
$$\mathbf{w} \equiv \mathbf{q} = 2\arctan\left\{\frac{\Omega T_d}{2}\right\}$$

represents the digital frequency. Alternatively, we can express the analog frequency Ω in terms of the digital frequency \mathbf{w} as

$$\Omega = \frac{2}{T_d} \tan\left(\frac{\mathbf{w}}{2}\right)$$

The figure below illustrates the relationship between Ω and \mathbf{w} . It is clear that compression occurs in the mapping process.

These results indicate that the entire left-half s -plane is mapped into a disc of radius 1 in the z -plane.



- **Example:** Design a digital low pass filter with the following specifications

$$0.89125 \leq |H(e^{j\omega})| \leq 1, \quad 0 \leq |\omega| \leq 0.2\pi$$

$$|H(e^{j\omega})| \leq 0.17783, \quad 0.3\pi \leq |\omega| \leq \pi$$

using the Bilinear transformation method and a Butterworth prototype filter. Compare the results with those obtained through the impulse invariant method.

Solution:

For simplicity, set $T_d = 1$. So $\Omega = 2 \tan(\omega/2)$. This means the digital passband and stopband frequencies,

$$\mathbf{w}_p = 0.2\mathbf{p}, \quad \mathbf{w}_s = 0.3\mathbf{p}$$

are mapped into the analog passband and stopband frequencies

$$\Omega_p = 2\tan(\mathbf{w}_p / 2) = 0.64984,$$

$$\Omega_s = 2\tan(\mathbf{w}_s / 2) = 1.0191$$

The specifications of the prototype analog filter become

$$\begin{aligned} 0.89125 \leq |H_c(j\Omega)| \leq 1, & \quad 0 \leq |\Omega| \leq 0.64984 \\ |H_c(j\Omega)| \leq 0.17783, & \quad 1.0191 \leq |\Omega| \leq \infty \end{aligned}$$

Next we define the following Matlab variables

$$Rp = 20\log_{10}(0.89125) = -1$$

$$Rs = 20\log_{10}(0.17783) = -15$$

$$Wp = 0.64984$$

$$Ws = 1.0191$$

and issue the command `[N,Wn]=buttord(Wp,Ws,Rp,Rs,'s')`. Matlab will return the order of the Butterworth filter in the variable N and the 3-db frequency in the variable Wn. It was found that

$$N = 6$$

$$Wn = 0.76627 \equiv \Omega_c$$

To complete the design of the prototype Butterworth filter, we next issue the Matlab command `[Z,P,K]=butter(N,Wn,'s')`. The zeros of $H_c(s)$ will then be returned in the array Z, the poles in the array P, and the filter gain (same as Ω_c^N) in the variable K. It was found that there are no zeros, the poles are

$$\begin{aligned}
& -7.4016e-001 + 1.9832e-001i \\
& -7.4016e-001 - 1.9832e-001i \\
& -5.4183e-001 + 5.4183e-001i \\
& -5.4183e-001 - 5.4183e-001i \\
& -1.9832e-001 + 7.4016e-001i \\
& -1.9832e-001 - 7.4016e-001i
\end{aligned}$$

and the gain is 0.20243. Denoting the poles as s_1, s_2, \dots, s_6 , the transfer function of the Butterworth filter is

$$\begin{aligned}
H_c(s) &= \frac{K}{\prod_{k=1}^6 (s - s_k)} \\
&= \frac{0.20243}{(s^2 + 0.39665s + 0.58716)(s^2 + 1.0837s + 0.58716)(s^2 + 1.4803s + 0.58716)}
\end{aligned}$$

This result is very close to that in Example 7.3 of the Text. Compared to the result obtained under the impulse invariant method, we observe a noticeable difference in $H_c(s)$.

Finally, the last step in the design exercise is to map the above transfer function into a digital filter using Bilinear transformation, i.e.

$$H(z) = H_c \left(2 \frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

This can be done by using the Matlab command

$$[Zd, Pd, Kd] = \text{bilinear}(Z, P, K, 1)$$

where Z , P , and K are same as before, 1 corresponds to the sampling frequency $1/T_d$, Zd is an array containing the zeros of $H(z)$, Pd is the array

storing the poles of $H(z)$, and K_d is the gain of $H(z)$. The results I obtained are:

$$K_d = 0.00073798$$

$Z_d =$

$$\begin{aligned} & -1 \\ & -1 \\ & -1 \\ & -1 \\ & -1 \\ & -1 \end{aligned}$$

$P_d =$

$$\begin{aligned} & 4.5216e-001 + 1.0510e-001i \\ & 4.5216e-001 - 1.0510e-001i \\ & 5.0527e-001 + 3.2087e-001i \\ & 5.0527e-001 - 3.2087e-001i \\ & 6.3430e-001 + 5.5026e-001i \\ & 6.3430e-001 - 5.5026e-001i \end{aligned}$$

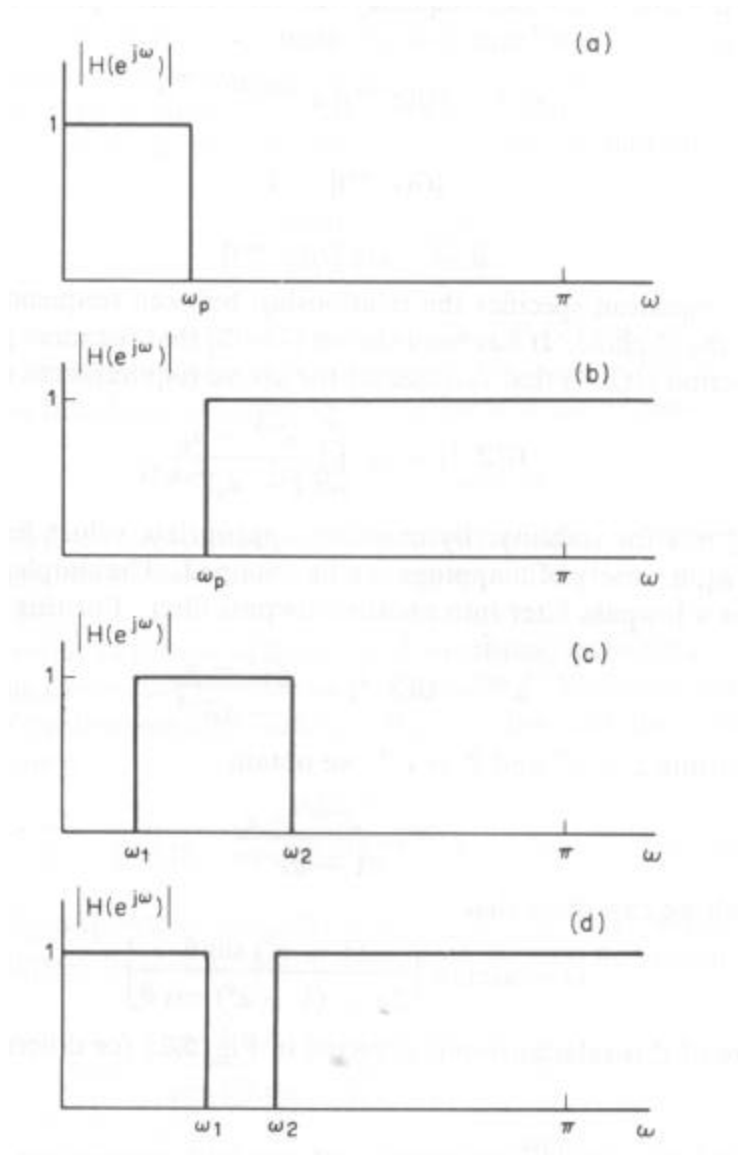
Calling the 6 zeros as z_1, z_2, \dots, z_6 and the 6 poles as p_1, p_2, \dots, p_6 . Then the transfer function of the digital filter $H(z)$ can be written as

$$\begin{aligned} H(z) &= K_d \frac{\prod_{k=1}^6 (1 - z_k z^{-1})}{\prod_{k=1}^6 (1 - p_k z^{-1})} \\ &= \frac{0.00073798 (1 + z^{-1})^6}{(1 - 1.2686z^{-1} + 0.70512z^{-2})(1 - 1.0105z^{-1} + 0.35826z^{-2})} \times \\ & \quad \frac{1}{(1 - 0.90433z^{-1} + 0.21550z^{-2})} \end{aligned}$$

The result is once again close to that in Example 7-3 of the text.

6.3 Frequency Transformation of Lowpass IIR Filters

- So far, we focus our discussion on the design of lowpass IIR filters. How about highpass, bandpass, and bandstop filters?



Frequency responses of lowpass, highpass, bandpass, and bandstop filters.

- The impulse invariant method will not be suitable for the highpass and bandstop filters because of heavy aliasing.

- It is possible to get around the problem by first designing a digital prototype lowpass filter and then perform an algebraic transformation on the digital lowpass filter to obtain the desired frequency selective filter.
- The transfer function of the digital prototype low pass filter is denoted by $H_{lp}(p)$ where p plays the same role as z , the conventional z -transform variable.

The transfer function of the desired frequency selective filter is denoted by $H(z)$.

We want to find a relationship between p and z , denoted by

$$p^{-1} = G(z^{-1})$$

such that

$$H(z) = H_{lp}(p) \Big|_{p^{-1}=G(z^{-1})}$$

It should be emphasized that the transformation relates p^{-1} to z^{-1} , not p to z . This is due to the fact that $H_{lp}(p)$ is usually expressed as a function in p^{-1} instead of p .

- We want a stable, rational $H_{lp}(p)$ be mapped into a stable rational $H(z)$. Consequently, the requirements on the mapping function $p^{-1} = G(z^{-1})$ are:
 1. $G(z^{-1})$ must be a rational function of z^{-1} .
 2. The inside of the unit circle in the p -plane is mapped into the inside of the unit circle in the z -plane.

3. The unit circle in the p -plane is mapped into the unit circle in the z -plane. So if \mathbf{q} and \mathbf{W} are the frequency variables in the p -plane and the z -plane, then

$$e^{-j\mathbf{q}} = G(e^{-j\mathbf{w}})$$

Consequently

$$\left|G(e^{-j\mathbf{w}})\right| = 1$$

and

$$\mathbf{q} = -\angle G(e^{-j\mathbf{w}}).$$

- The most general form of the function $G(z^{-1})$ that satisfies all the above requirements is:

$$p^{-1} = G(z^{-1}) = \pm \prod_{k=1}^N \frac{z^{-1} - \mathbf{a}_k}{1 - \mathbf{a}_k z^{-1}}; \quad |\mathbf{a}_k| < 1$$

By choosing appropriate values for N and the \mathbf{a}_k 's, a variety of mappings can be obtained.

- **Low-pass to Low-pass transformation:**

$$p^{-1} = \frac{z^{-1} - \mathbf{a}}{1 - \mathbf{a}z^{-1}}$$

This means the relationship between \mathbf{q} and \mathbf{W} is

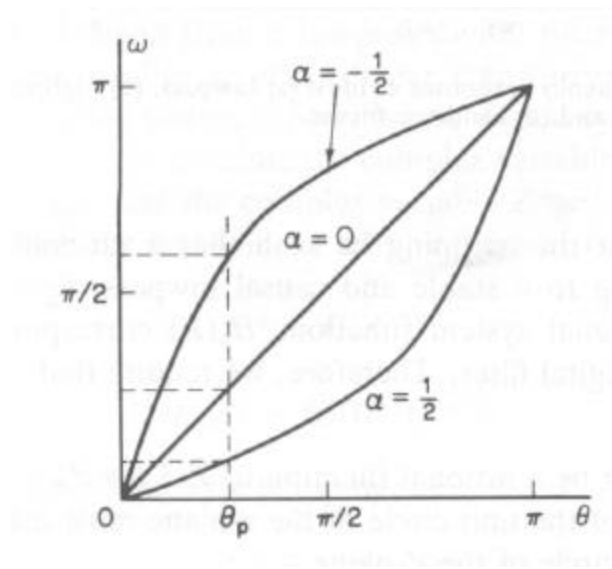
$$e^{-j\mathbf{q}} = \frac{e^{-j\mathbf{w}} - \mathbf{a}}{1 - \mathbf{a}e^{-j\mathbf{w}}}$$

or

$$\begin{aligned}
 e^{-jw} &= \frac{e^{-jq} + \mathbf{a}}{1 + \mathbf{a}e^{-jq}} \\
 &= \frac{(e^{-jq} + \mathbf{a})(1 + \mathbf{a}e^{jq})}{(1 + \mathbf{a}e^{-jq})(1 + \mathbf{a}e^{jq})} \\
 &= \frac{2\mathbf{a} + (1 + \mathbf{a}^2)\cos(\mathbf{q}) + j(\mathbf{a}^2 - 1)\sin(\mathbf{q})}{1 + \mathbf{a}^2 + 2\mathbf{a}\cos(\mathbf{q})}
 \end{aligned}$$

or

$$\mathbf{w} = \arctan \left[\frac{(1 - \mathbf{a}^2)\sin \mathbf{q}}{2\mathbf{a} + (1 + \mathbf{a}^2)\cos \mathbf{q}} \right]$$



For a given pair of \mathbf{q}_p and \mathbf{w}_p , the parameter \mathbf{a} is

$$\mathbf{a} = \frac{\sin \left[(\mathbf{q}_p - \mathbf{w}_p) / 2 \right]}{\sin \left[(\mathbf{q}_p + \mathbf{w}_p) / 2 \right]}$$

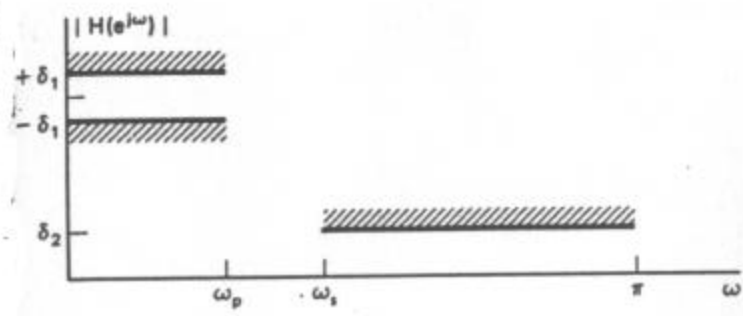
It can be shown that the absolute value of \mathbf{a} is less than 1.

- Transformations from LP to LP, HP, BP, and BS filters are shown below:

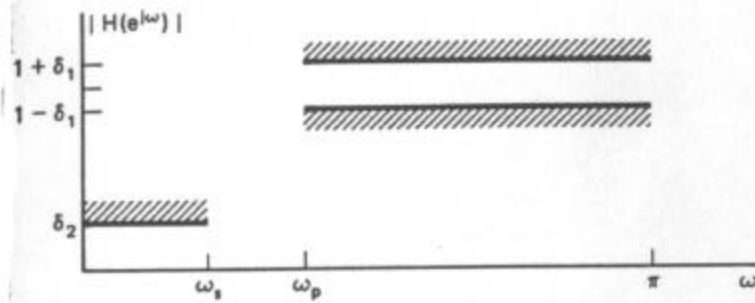
TABLE 7.1 TRANSFORMATIONS FROM A LOWPASS DIGITAL FILTER PROTOTYPE OF CUTOFF FREQUENCY θ_p

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

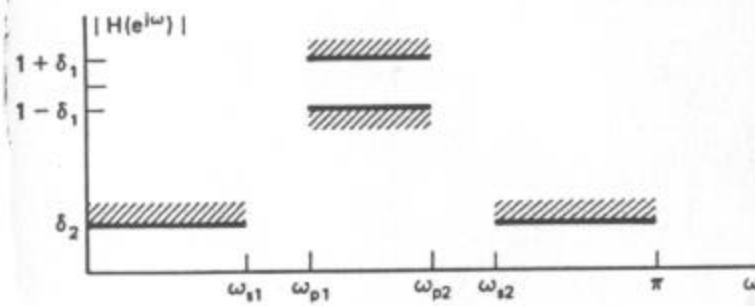
The corresponding definitions of filter cutoff frequencies are shown in the next page.



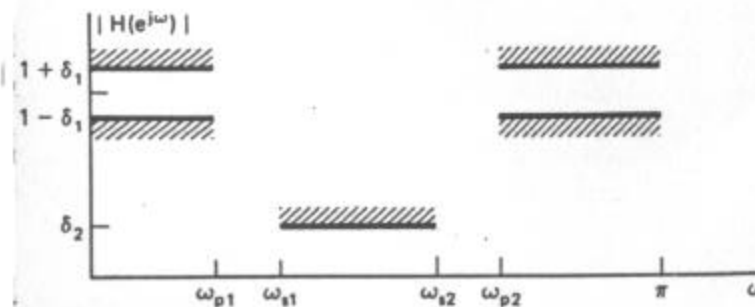
(a)



(b)



(c)



(d)

(a) low pass, (b) high pass, (c) bandpass, (d) bandstop.

- **Example:** The prototype low pass filter

$$H_{lp}(p) = \frac{0.001836(1+p^{-1})^4}{(1-1.5548p^{-1}+0.6493p^{-2})(1-1.4996p^{-1}+0.8482p^{-2})}$$

satisfies

$$\begin{aligned} 0.89125 \leq |H(e^{jq})| \leq 1, & \quad 0 \leq |q| \leq 0.2\mathbf{p} \\ |H(e^{jq})| \leq 0.17783, & \quad 0.3\mathbf{p} \leq |q| \leq \mathbf{p} \end{aligned}$$

Determine the corresponding high pass filter that satisfies the requirements

$$0.89125 \leq |H(e^{jw})| \leq 1; \quad 0.6\mathbf{p} \leq |w| \leq \mathbf{p}$$

Solution:

The passband frequency of the high pass filter is $w_p = 0.6\mathbf{p}$ and the passband frequency of the digital prototype filter is $q_p = 0.2\mathbf{p}$. So according to the design formula,

$$\mathbf{a} = \frac{\cos\left(\frac{q_p + w_p}{2}\right)}{\cos\left(\frac{q_p - w_p}{2}\right)} = -0.38197$$

This means

$$p^{-1} = -\frac{z^{-1} + \mathbf{a}}{1 + \mathbf{a}z^{-1}} = \frac{-z^{-1} + 0.38197}{1 - 0.38197z^{-1}}$$

Consequently,

$$H(z) = H_{lp}(p) \Big|_{p^{-1} = (-z^{-1} + 0.38197)/(1 - 0.38197z^{-1})}$$

$$= \frac{0.02426(1 - z^{-1})^4}{(1 - 1.0416z^{-1} + 0.4019z^{-2})(1 - 0.5661z^{-1} + 0.7647z^{-2})}$$

It should be pointed out that in actual design, the stopband frequency of the digital prototype filter will be determined by the stopband frequency of the high pass (or desired) filter. We assume in this example that the stop band frequency of the high pass filter is mapped into $\mathbf{q}_s = 0.3\mathbf{p}$ based on $\mathbf{a} = -0.38197$.

6.4 Implementation Structures for IIR Filters

- *Reference:* Section 6.3 of Text
- The transfer function of an IIR filter can always be expressed as a ratio of two polynomials, i.e.

$$H(z) = \frac{B(z)}{A(z)},$$

where $A(z)$ and $B(z)$ are respectively polynomials of orders N and M in z^{-1} .

It is the denominator polynomial that makes the impulse response of the filter infinitely long.

- Different expressions for $A(z)$, $B(z)$, and $H(z)$ lead to different implementation structures. For example, if these two polynomials are written as

$$A(z) = 1 - \sum_{k=1}^N a_k z^{-k}$$

and

$$B(z) = \sum_{k=0}^M b_k z^{-k},$$

then

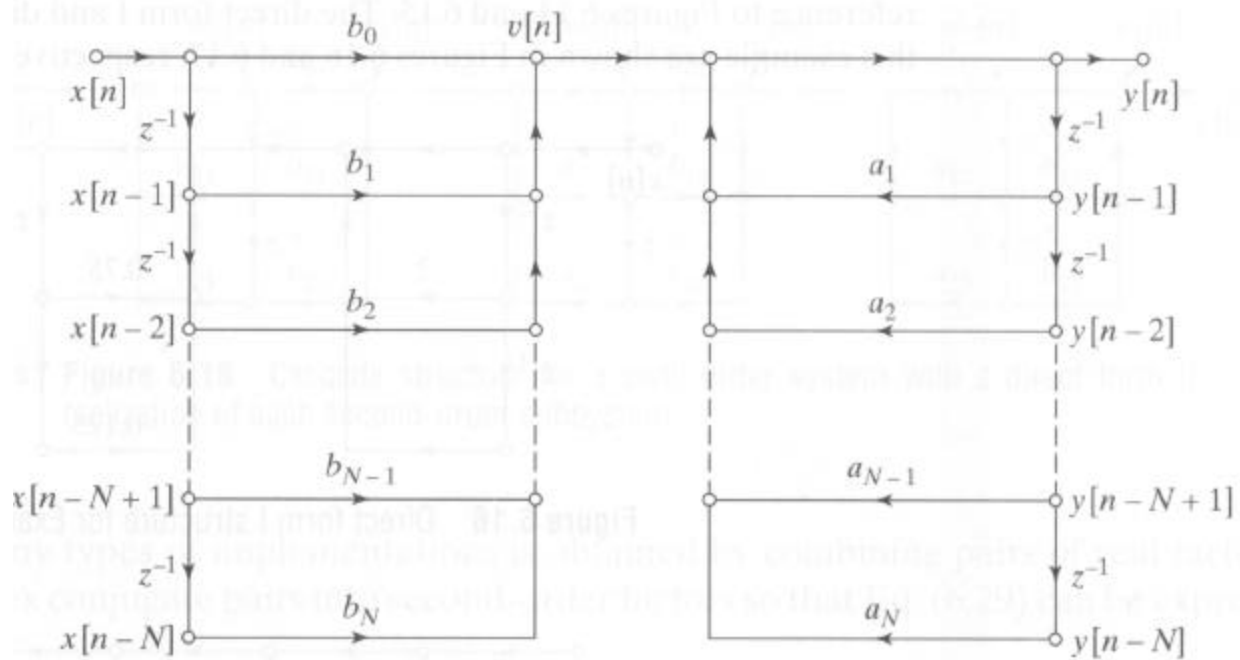
$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}}.$$

This means in the time domain, the input $x[n]$ and the output $y[n]$ of the filter satisfies

$$y[n] - \sum_{k=1}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k],$$

which is simply a linear constant coefficient different equation.

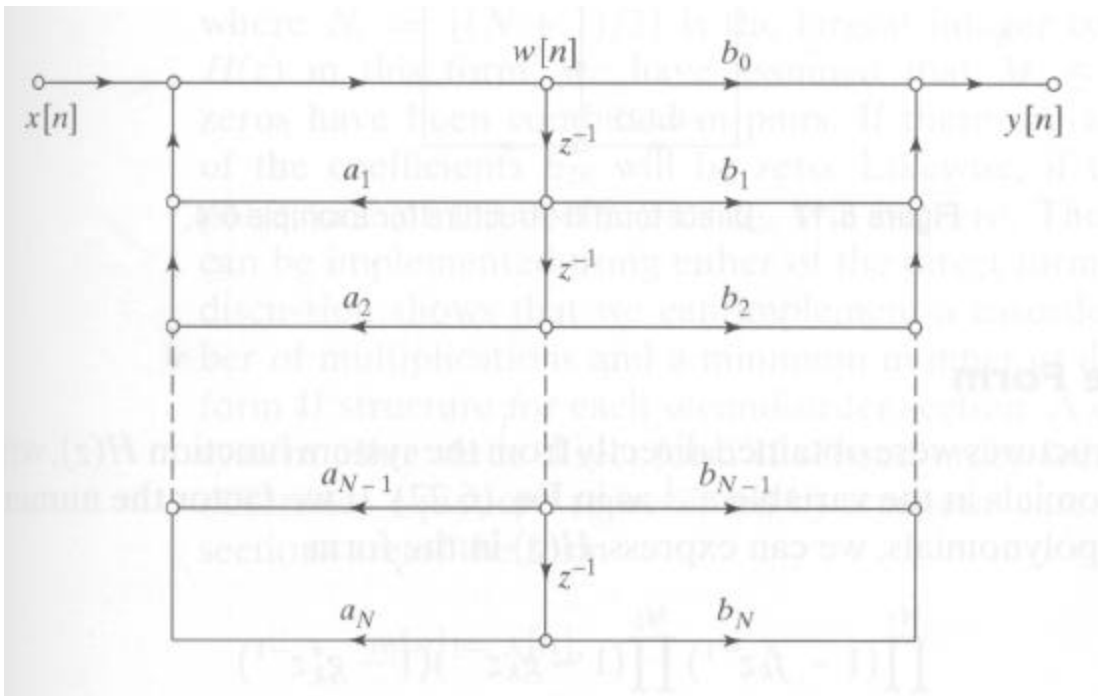
A possible computational structure for $y[n]$ is shown in the signal flow graph in the next page. There, a branch with a transmittance of z^{-1} is equivalent to a delay of 1 sample, and a branch with a transmittance a_i (or b_j) implies a scaling of the signal at the originating node of that branch by the constant a_i (or b_j). As in any signal flow graph, the signal at any node, i.e. the node value, is the sum of products of the signal at an originating node and the corresponding branch transmittance.



The above computational structure is called the *Direct Form I* implementation of an IIR filter. Note that for simplicity, we assume that the order of the numerator and denominator polynomials are identical, i.e. $N = M$.

It is observed that the Direct Form I implementation structure requires $2N + 1$ multiplications, $2N$ additions, and $2N$ delay elements.

- The number of delay elements can be reduced to N if we interchange the two sections in the Direct Form I structure. This leads us to the *Direct Form II* structure shown below.



- It is also possible to express the transfer function of an IIR filter in product form as

$$H(z) = C \frac{\prod_{k=1}^{M_1} (1 - f_k z^{-1}) \prod_{k=1}^{M_2} (1 - g_k z^{-1})(1 - g_k^* z^{-1})}{\prod_{k=1}^{N_1} (1 - c_k z^{-1}) \prod_{k=1}^{N_2} (1 - d_k z^{-1})(1 - d_k^* z^{-1})},$$

where C is a constant, the f_k 's are the real zeros of the transfer function, the g_k 's are the complex zeros (which always exist in conjugate pairs), the c_k 's are the real poles, and the d_k 's are the complex poles (which always exist in conjugate pairs).

Note that the products $(1 - g_k z^{-1})(1 - g_k^* z^{-1})$ and $(1 - d_k z^{-1})(1 - d_k^* z^{-1})$ are 2nd order polynomials in z^{-1} with real coefficients.

For simplicity, we assume that $N_1 = M_1 = 2K$ (K an integer) and $N_2 = M_2$. This means we can combine pairs of real poles and combine pairs of real zeros and rewrite $H(z)$ as

$$H(z) = \prod_{k=1}^{N_s} \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$

$$= \prod_{k=1}^{N_s} H_k(z),$$

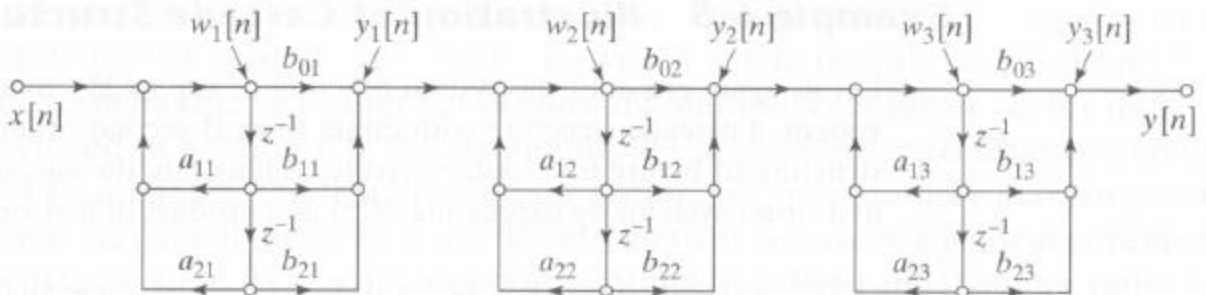
where

$$H_k(z) = \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$

and

$$N_s = K + N_2.$$

This expression for the transfer function enables us to visualize the IIR filter as the serial concatenation of N_s subsystems. Subsequently it leads to the *Cascade implementation structure*; see for example the case of $N_s = 3$ below.



- **Exercise:** Draw the Cascade form of the high pass filter in Section 6-3.

- With the Cascade form implementation, the locations of the poles and zeros will not change dramatically when the a_{ij} 's and the b_{ij} 's are quantized (as in fixed point implementation). This is in contrast to the Direct form implementation where the poles and zeros can change substantially because of quantization. This issue will be discussed in details in Chapter 7 of the lecture notes.
- If we express the transfer function as a partial fraction expansion, then

$$\begin{aligned}
 H(z) &= \sum_{k=1}^{N_s} \frac{e_{0k} + e_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}} \\
 &= \sum_{k=1}^{N_s} G_k(s),
 \end{aligned}$$

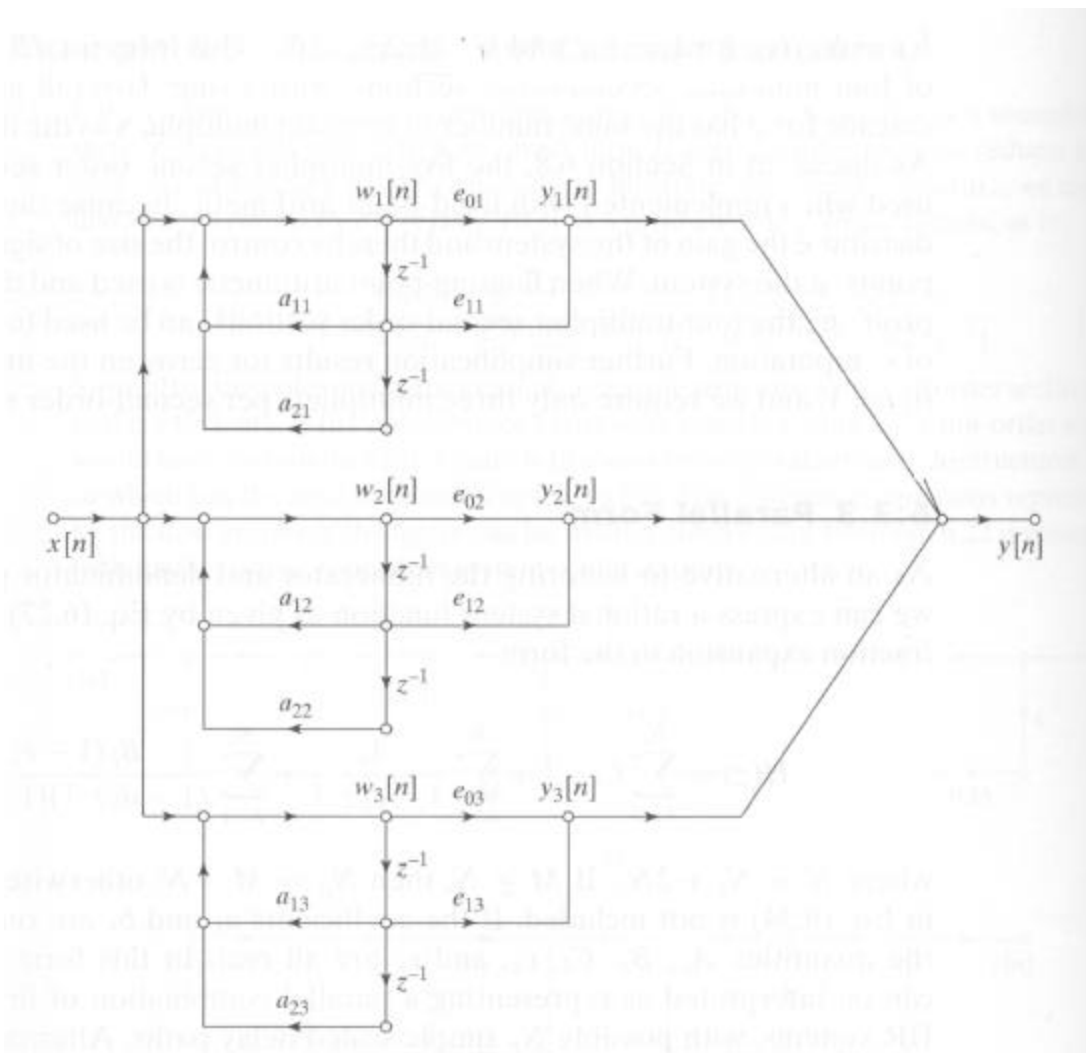
where

$$G_k(s) = \frac{e_{0k} + e_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

and all the polynomial coefficients are real.

The transfer function suggests that the IIR filter can be viewed as a “large” system comprising of N_s parallel subsystems. The input to all the subsystems is $x[n]$ and the output of the IIR filter, $y[n]$, is the sum of the $y_k[n]$'s , the individual outputs of the subsystems.

The figure below shows this *parallel form* of the IIR filter.



- The parallel form offers robustness against quantization error for the poles. The zeros, however, will be affected by the shifting of the poles in the individual subsystems.
- **Exercise:** While it is straight forward to incorporate any given initial condition of an IIR filter in the Direct Form I structure, it is not clear how this can be done in the other structures. Is this a matter of great concern? If so, suggest how a given initial condition can be incorporated into the Direct Form II, the Cascade form, and the Parallel form.