1 Infinite-impulse response filter design

These notes describe some basic techniques for designing real causal infinite impulse response (IIR) filters. The goal for this filter design problem is to find a linear, constant-coefficient difference equation which approximates a desired frequency response magnitude $|H_{id}(e^{j\Omega})|$. Because these techniques involve transforming continuous-time (CT) filters to discrete-time (DT) filters, we will use Ω for the DT frequency and ω for the CT frequency, just as in Chapter 7 on sampling. This should hopefully reduce the confusion between the two frequency variables.

Just as in FIR filter design, it is not possible to obtain an ideal frequency response from a finite-order difference equation, so we will have to approximate our desired frequency response magnitude, as shown in Figure 1 for a DT lowpass filter. The same terminology is used to describe these specifications as in the FIR case earlier this semester. The passband is the region $\Omega < \Omega_p$, the stopband is the region $\Omega > \Omega_s$. The transition band is still the region between the passband and stopband, $\Omega_p \leq \Omega \leq \Omega_s$, with width $\Delta\Omega = \Omega_s - \Omega_p$. Within the passband, we accept some ripple given by δ_1, and similarly the stopband ripple is given by stopband (δ_2). For the IIR designs discussed in these notes, both the passband and stopband ripples will often be given in dB of attenuation, which is computed as $-20\log_{10}\delta$. The IIR filters discussed here can have different ripples in the passband and stopband, which is an important change from the FIR filter designs discussed earlier in the semester.

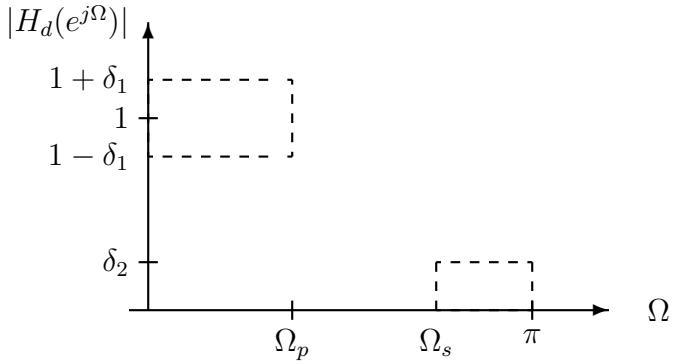

Figure 1: Specifications for the frequency response magnitude of a discrete-time lowpass filter

Section 2.4.2 of the textbook described how recursive linear, constant-coefficient difference equations with delayed $y[n-k]$ terms produce infinite-impulse response (IIR) filters. To simplify things, we will assume that all of the coefficients are real, so that $h[n]$ is also real and $|H_d(e^{j\Omega})|$ is even symmetric. The task is then to find the order $\max(N, M)$ and coefficients a_k and b_ℓ in

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{\ell=0}^{M} b_\ell x[n-\ell] \tag{1}$$

so that the frequency response

$$H_d(e^{j\Omega}) = \frac{\sum_{\ell=0}^{M} b_\ell e^{-j\Omega\ell}}{\sum_{k=0}^{N} a_k e^{-j\Omega k}} \tag{2}$$

satisfies the specification given in Figure 1.

1.1 Bilinear transform

The most common method for designing IIR filters is called the bilinear transform. This technique begins with a CT system function $H_c(s)$, and makes a change of variables for s to get a DT system function $H_d(z)$. The simplest change of variables to do this is

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}. \tag{3}$$

Before we get into details of how the DT filter design procedures works, let's look at an example of using this change of variables to transform a CT filter into a DT filter.

Example 1 Consider the CT lowpass filter with the system function

$$H_c(s) = \frac{0.1925}{s^3 + 1.1547s^2 + 0.6667s + 0.1925}. \tag{4}$$

The CT frequency response magnitude for this filter is shown in Figure 2. This third-order lowpass filter was designed so that the -3 dB point in the passband was at $\omega = 1/\sqrt{3} \approx 0.58$.

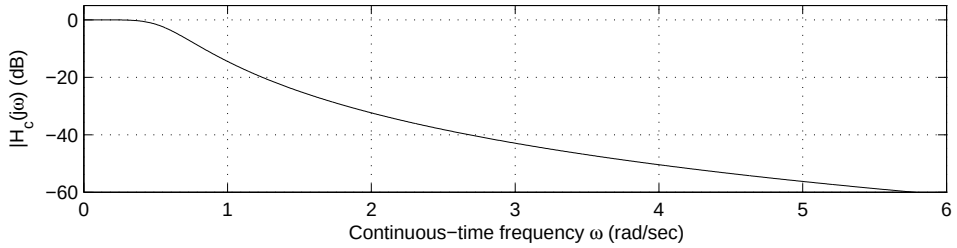

Figure 2: Continuous-time frequency response magnitude $|H_c(j\omega)|$ for Example 1.

The discrete-time system function is obtained by replacing each s in Eq. 4 by $(1 - z^{-1})/(1 + z^{-1})$, then solving for new polynomials in z^{-1}. When the dust clears from all this algebra, the result is

$$H_d(z) = \frac{0.0639 + 0.1916z^{-1} + 0.1916z^{-2} + 0.0639z^{-3}}{1 - 0.9658z^{-1} + 0.5826z^{-2} - 0.1060z^{-3}}. \tag{5}$$

Substituting $z = e^{j\Omega}$ into $H_d(z)$ gives the frequency response shown in Figure 3. The new DT filter is still a lowpass filter, just like the CT filter used as the prototype. The -3 dB

point for this new filter falls at $\pi/3$. There is straightforward relationship between the CT and DT frequencies for this change of variables which we'll derive shortly. Using our razor-sharp z-transform skills, we can find that the linear, constant-coefficient difference equation for this system is

$$\begin{aligned} 0.0639y[n] + 0.1916y[n-1] + 0.1916y[n-2] + 0.0639y[n-3] = \\ x[n] - 0.9658x[n-1] + 0.5826x[n-2] - 0.1060x[n-3]. \end{aligned} \tag{6}$$

This filter can be implemented using a block diagram with the coefficients from Eq. 6.

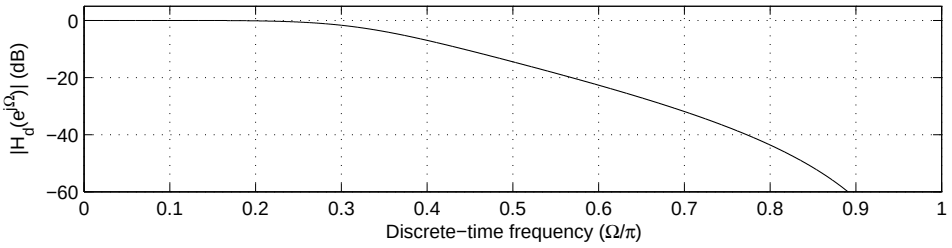

Figure 3: Discrete-time frequency response magnitude $|H_d(e^{j\Omega})|$ for Example 1.

This example illustrates some important properties of the bilinear transfer. First, it transforms causal, stable CT filters into causal, stable DT filters. Second, it preserves the type of frequency selective filters like lowpass or highpass filters. However, it raises an important question. In this example, we began with a CT filter and changed it into a DT filter. How can we choose the right CT filter so that the DT filter which results from the change of variables satisfies our specification? To answer this question, let's look more closely at the change of variables.

The change of variables $s = (1 - z^{-1})/(1 + z^{-1})$ has two important properties. Poles in the left-half s-plane map to poles inside the unit circle of the z-plane (You'll get to demonstrate this on Problem Set 10.) As mentioned above, this means that casual stable CT filters map to causal stable DT filters. The second important property is that the CT frequency axis $s = j\omega$ maps onto the DT frequency axis $z = e^{j\Omega}$. (Again, more fun awaiting you on Problem Set 10.) Because the frequency response is determined on the $s = j\omega$ contour for CT systems, and on the $z = e^{j\Omega}$ contour for DT systems, the mapping between the two frequencies is obtained by making these substitutions into Eq. (3).

$$\begin{aligned} s\big|_{s=j\omega} &= \left.\frac{1-z^{-1}}{1+z^{-1}}\right|_{z=e^{j\Omega}} \\ j\omega &= \frac{1-e^{-j\Omega}}{1+e^{-j\Omega}} \\ &= \left(\frac{e^{-j\Omega/2}}{e^{-j\Omega/2}}\right)\left(\frac{e^{j\Omega/2}-e^{-j\Omega/2}}{e^{j\Omega/2}+e^{-j\Omega/2}}\right) \\ &= \frac{2j\sin(\Omega/2)}{2\cos(\Omega/2)} \\ \omega &= \tan(\Omega/2) \qquad \text{or} \qquad (7) \\ \Omega &= 2\arctan(\omega) \qquad (8) \end{aligned}$$

As a result, we can relate the CT and DT frequency responses for the bilinear transform according to

$$H_d(e^{j\Omega}) = H_c(j\omega)|_{\omega=\tan(\Omega/2)} \tag{9}$$

Figure 4 plots Eq. (8) over a wide range of ω. There are two important things to notice about this picture. First, the change of variables is not linear, so there is no simple scaling factor between ω and Ω as there is in unaliased sampling with $\Omega = \omega T$. Frequency selective filters such as lowpass, highpass and bandpass filters will preserve their type in the mapping, since their frequency responses are piecewise constant. However, frequency responses which are not piecewise constant, such as a DT differentiator, will not preserve their shape through this mapping. Second, the nonlinear mapping warps the ω frequency axis so that the infinitely long $s = j\omega$ axis fits onto one circumference of the unit circle. A consequence of this is that large values of ω will be compressed very close together in Ω. For example, compare how close $\omega = 15$ and $\omega = 20$ end up in Ω compared with $\omega = 5$ and $\omega = 10$.

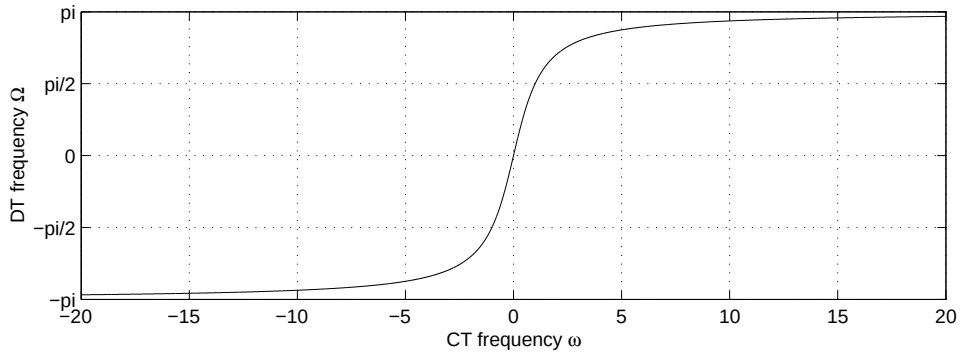

Figure 4: Mapping between CT frequency ω and DT frequency Ω from Eq. (8)

Figure 4 and Eqs. (7) – (9) contain the key to our design problem. With this knowledge of how the frequencies map between CT and DT for the bilinear transform, we can work backward from our desired passband edge Ω_p and stopband edge Ω_s to find the appropriate band edges ω_p and ω_s for our CT prototype filter. If we design the CT filter to meet these specifications, the DT filter that results when we make the change of variables from Eq. (3) will "automatically" satisfy our DT specifications.

Example 2 Suppose that we want to design a DT lowpass filter with $\Omega_p = \pi/2$ and $\Omega_s = 3\pi/4$. From Eq. (7), we get the CT band edge frequencies as

$$\begin{aligned} \omega_p &= \tan(\Omega_p/2) = \tan(\pi/4) = 1 \\ \omega_s &= \tan(\Omega_s/2) = \tan(3\pi/8) \approx 2.41 \end{aligned}$$

Note that because the bilinear transform only warps the frequency axis, and not the vertical axis, the passband and stopband ripples of δ_1 and δ_2 will not change between DT and CT. There are many common CT filter design techniques and programs to find a system function $H_c(s)$ satisfying these specifications. Once you obtain $H_c(s)$, making

the change of variables from Eq. (3) and a lot of algebra later, you would have $H_d(z)$ satisfying the specification given.

Pulling together all of these ideas results in a four step procedure for designing a DT frequency selective filter using the bilinear transform.

1. Write down the DT specification clearly in terms of the band edges and ripples.
2. Pre-warp the band edge frequencies using Eq. (7) to find the appropriate CT filter specification. The ripples in each band remain unchanged.
3. Find the system function $H_c(s)$ using a common CT filter design technique.
4. Substitute Eq. (3) into $H_c(s)$ to find $H_d(z)$ for a DT filter satisfying the specifications from Step 1.

The CT filter design in Step 3 is usually one of the four common CT filters: Butterworth, Chebyshev Type I, Chebyshev Type II, and Elliptic (or Cauer) filters. We will discuss these in the following section.

1.2 Designing IIR filters in Matlab

Matlab provides helpful functions which greatly simplify the design of IIR filters. These functions will compute the filter order needed to satisfy a DT specification, and then design the CT filter, perform the bilinear transform, and do all the algebra to return $H_d(z)$ for you. In this section, we will introduce the functions used to define each of the four common filter types. To do this, we will use the same DT specifications for all four filters, and compare the results we obtain. The DT filter we will design is a lowpass filter satisfying

$$\begin{array}{rcl|ll} 0.99 & \leq & |H_d(e^{j\Omega})| & \leq & 1.01, & |\Omega| \leq 0.2\pi, \\ & & |H_d(e^{j\Omega})| & \leq & 0.001, & 0.4\pi \leq |\Omega| \leq \pi. \end{array} \tag{10}$$

In the examples that follow, the Butterworth filter requires the largest order to satisfy this specification, the elliptic filter satisfies the filter with the smallest order, and the Chebyshev filters fall in the middle. This is generally the trend in filter order vs. filter type for any frequency selective filter designed with the bilinear transformation.

1.2.1 Butterworth filters

Butterworth filters have frequency responses which decrease monotonically from one in the passband to nearly zero in the stopband without ripples. Figure 5 plots the frequency response (in dB) of a 12th order Butterworth filter satisfying the specification from Eq. 10. As expected, the passband is nearly flat until 0.2π, then drops rapidly so that it is below $-20\log_{10} 0.001 = 60$ dB by the time it reaches 0.4π. Figure 6 plots just the passband and stopband on a linear frequency scale to confirm that they satisfy the DT specification given in Eq. (10).

The Matlab command `[N,Omegan] = buttord(Omegap,Omegas,Rp,Rs)` is used to find the order `N` and cutoff frequency `Omegan` necessary for a Butterworth lowpass filter to satisfy the specification given. `Omegap` is the passband edge normalized by π, i.e., Ω_p/π.

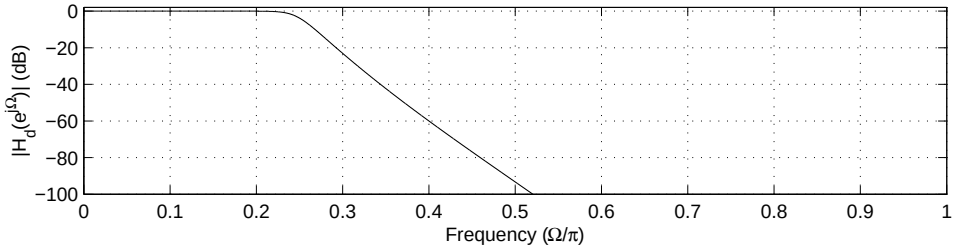

Figure 5: Frequency response magnitude $|H_d(e^{j\Omega})|$ for a 12th order Butterworth filter satisfying Eq. (10)

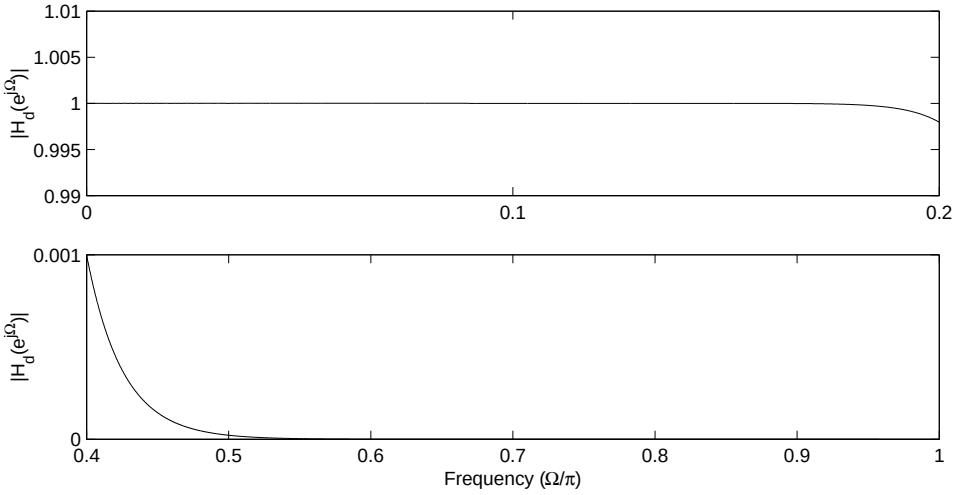

Figure 6: Zoom in on the passband and stopband for Butterworth filter $|H_d(e^{j\Omega})|$

Similarly, `Omegas` is the stopband edge normalized by π, or Ω_s/π. `Rp` is the minimum allowable passband amplitude in dB. In terms of the specification in Figure 1, this is `Rp` $= -20\log_{10}(1-\delta_1)$. The last parameter, `Rs`, is the maximum allowable stopband ripple in dB, or $-20\log_{10}(\delta_2)$. For the example here, the following commands solve for the order and cutoff frequency of the Butterworth filter in Figure 5.

```
Omegap = 0.2;
Omegas = 0.4;
delta1 = 0.01;
delta2 = 0.001;
Rp = -20*log10(1-delta1);
Rs = -20*log10(delta2);
[N,Omegan] = buttord(Omegap,Omegas,Rp,Rs);
```

Once you obtain the values of `N` and `Omegan`, the Matlab function `butter` computes the numerator and denominator coefficients for $H_d(z)$ as shown below

```
[bbutter,abutter] = butter(N,Omegan);
```

The vectors `bbutter` and `abutter` above are the linear, constant-coefficient difference equation coefficients for the filter. Using z-transform properties, these are also the coefficients for the numerator and denominator polynomials in z^{-1} in $H_d(z)$. For this 12th order filter, these vectors are each 13 elements long, and therefore not very practical to write out for $H_d(z)$ or the linear, constant-coefficient difference equation . In order to find these values, Matlab has done all of the algebra required by the change of variables from Eq. (3).

DT Butterworth filters generally have a distinctive pole-zero pattern. Figure 7 plots the pole-zero response for the 12th order filter whose frequency response is shown in Figure 5. The poles are in a radius around $\Omega = 0$ to raise the passband frequency response, while the zeros are clustered around $\Omega = \pi$ to bring down the stopband.

There are other input options to `buttord` and `butter` to produce other common filters such as highpass and bandpass filters. You are encouraged to read the `help` on these commands to learn how to do this.

1.2.2 Chebyshev Type I filters

Chebyshev Type I filters have ripples in the passband but a smooth monotonic stopband. Figure 8 plots the frequency response of a 7th order Chebyshev Type I filter satisfying Eq. (10). Because it is plotted on a dB scale, the passband ripples are not obvious. It is clear from this plot that the stopband is more than 60 dB down by $\Omega_s = 0.4\pi$. Figure 9 enlarges the passband and stopband. The passband ripples are clearly visible in the top panel, but the ripples stay within the range specified in Eq. (10). The bottom panel demonstrates that the stopband actually exceeds the specification.

The Matlab commands to make these filters are similar to the Butterworth filter commands. `cheb1ord` uses the same input arguments as `buttord` to find the order `N` and break frequency `Omegan`.

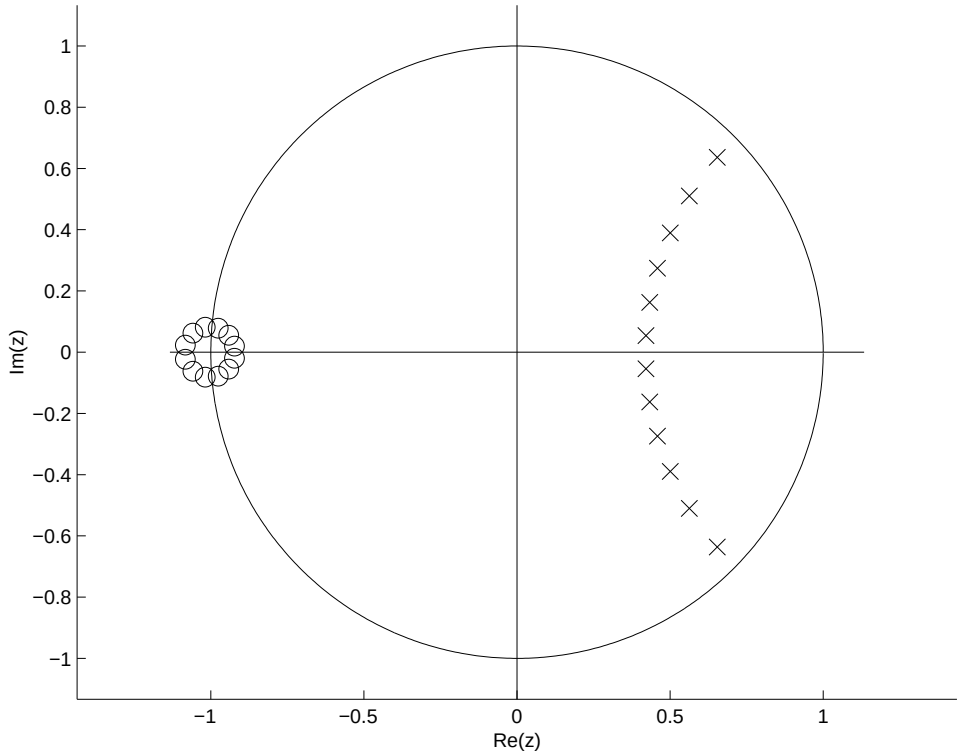

Figure 7: Pole-zero plot for the Butterworth filter satisfying Eq. (10)

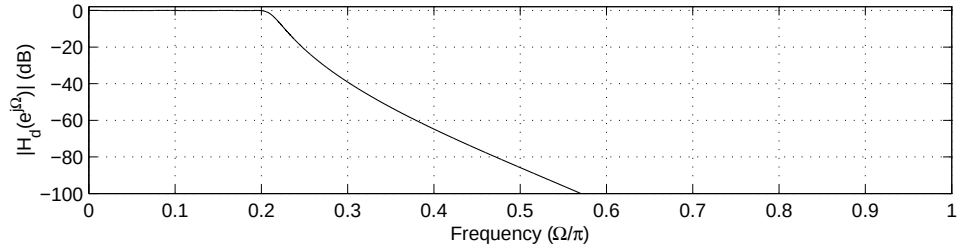

Figure 8: Frequency response magnitude $|H_d(e^{j\Omega})|$ for a 7th order Chebyshev Type I filter satisfying Eq. (10)

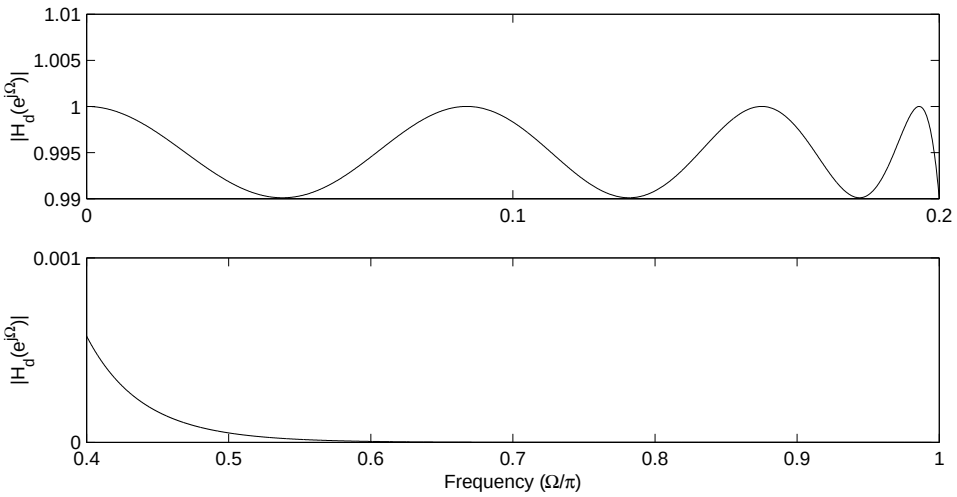

Figure 9: Zoom in on the passband and stopband for Chebyshev Type I filter $|H_d(e^{j\Omega})|$

```
Omegap = 0.2;
Omegas = 0.4;
delta1 = 0.01;
delta2 = 0.001;
Rp = -20*log10(1-delta1);
Rs = -20*log10(delta2);
Rpb = min(abs([20*log10(1+delta1),20*log10(1-delta1)]));
[N,Omegan] = cheb1ord(Omegap,Omegas,Rp,Rs);
```

The Matlab command `cheby1` returns the linear, constant-coefficient difference equation coefficients for the filter. `cheby1` takes both `N` and `Omegan` as input arguments. It also requires an additional argument specifying the passband ripple limits in dB. This is the smaller of the absolute values of $20\log_{10}(1+\delta_1)$ and $20\log_{10}(1-\delta_1)$.

```
Rpb = min(abs([20*log10(1+delta1),20*log10(1-delta1)]));
[bcheby1,acheby1] = cheby1(N,Rpb,Omegan);
```

Again, the output vectors `bcheby1` and `acheby1` contain the linear, constant-coefficient difference equation coefficients for the filter, or the numerator and denominator coefficients for $H_d(z)$. We can use these coefficients to find the pole-zero plot shown in Figure 10. The pattern of poles and zeros shown is typical of Chebyshev Type I filters. The collection of poles near $\Omega = 0$ raises the passband, but the nearly linear arrangement results in ripples shown in Figure 9. This change in the pole locations from the arc of the Butterworth filter in Figure 7 produces the passband ripples, in contrast to the smooth passband of the Butterworth filter. The collection of zeros at $\Omega = \pi$ brings the frequency response down for the stopband.

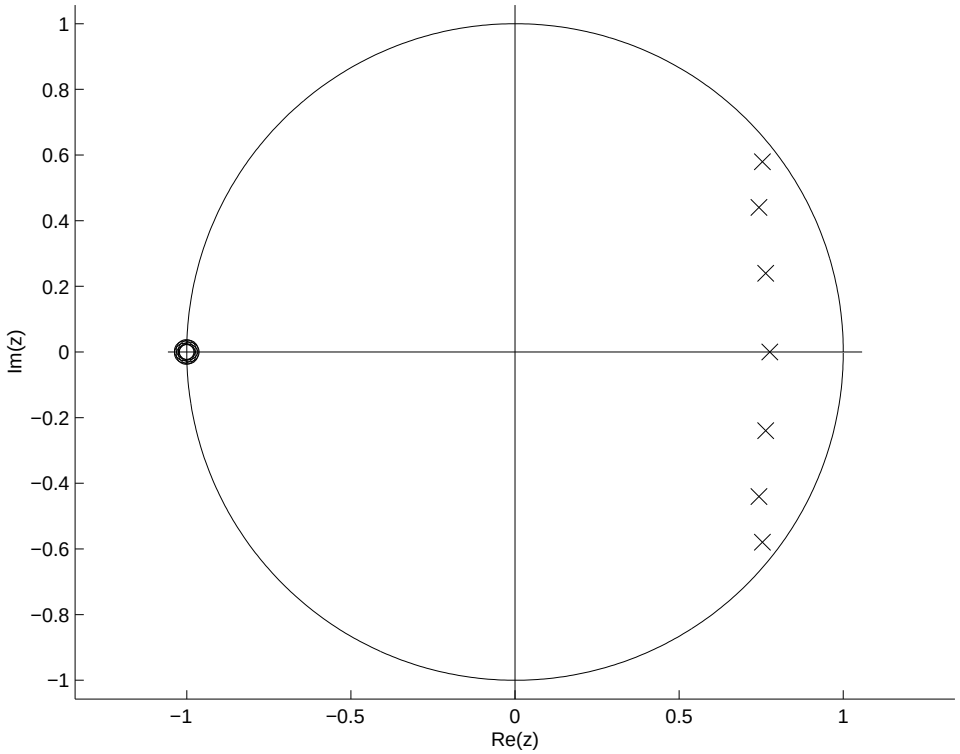

Figure 10: Pole-zero plot for the Chebyshev Type I filter satisfying Eq. (10)

1.2.3 Chebyshev Type II filters

Chebyshev Type II filters have a monotonic passband and ripples in the stopband. Figure 11 plots the frequency response of a 7th order Chebyshev Type II filter satisfying Eq. (10). The passband is basically flat, and the ripples in the stopband all fall below -60 dB. Figure 12 enlarges the passband and stopband. The passband just meets the specification at $\Omega_p = 0.2\pi$, while the stopband ripples go up to 0.001 but no higher.

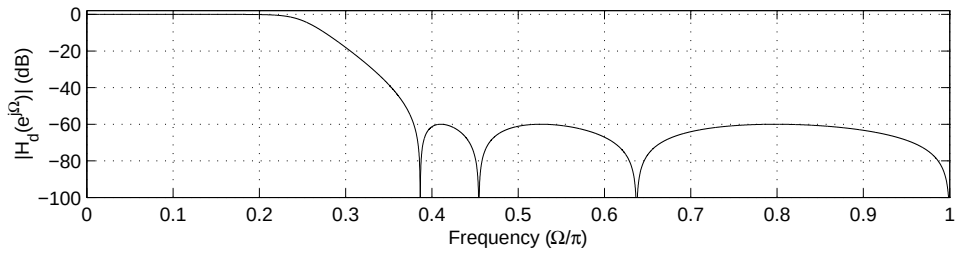

Figure 11: Frequency response magnitude $|H_d(e^{j\Omega})|$ for a 7th order Chebyshev Type II filter satisfying Eq. (10)

The Matlab command `cheb2ord` has the same inputs and outputs as `cheb1ord`. The linear, constant-coefficient difference equation coefficients are computed by `cheby2`, which is similar to `cheby1` in input arguments with one change. The second input argument to `cheby2` is the maximum stopband ripple in dB, which is just Rs. The following commands

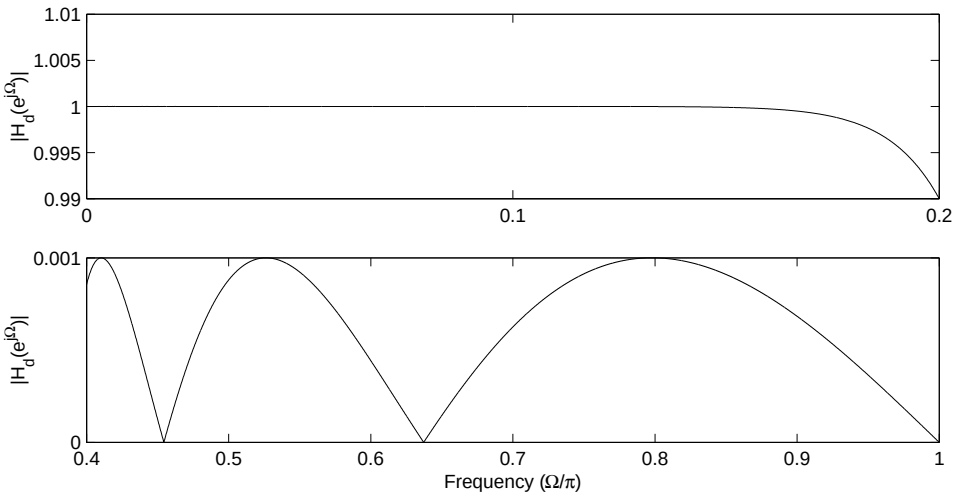

Figure 12: Zoom in on the passband and stopband for Chebyshev Type II filter $|H_d(e^{j\Omega})|$

define the filter coefficients for the filter whose frequency response appears in Figure 11

```
Omegap = 0.2;
Omegas = 0.4;
delta1 = 0.01;
delta2 = 0.001;
Rp = -20*log10(1-delta1);
Rs = -20*log10(delta2);
[N,Omegan] = cheb2ord(Omegap,Omegas,Rp,Rs);
[bcheby2,acheby2] = cheby2(N,Rs,Omegan);
```

Figure 13 plots the poles and zeros for this filter. This pattern is typical of Chebyshev Type II filters. The poles which produce the smooth passband are in an arc similar to that seen in the Butterworth filter. The zeros for the stop band are right on the unit circle, causing the deep notches seen in Figures 11 and 12. The angles of the zeros on the unit circle correspond to the frequencies Ω with $|H_d(e^{j\Omega})| = 0$ seen in the bottom panel of Figure 12.

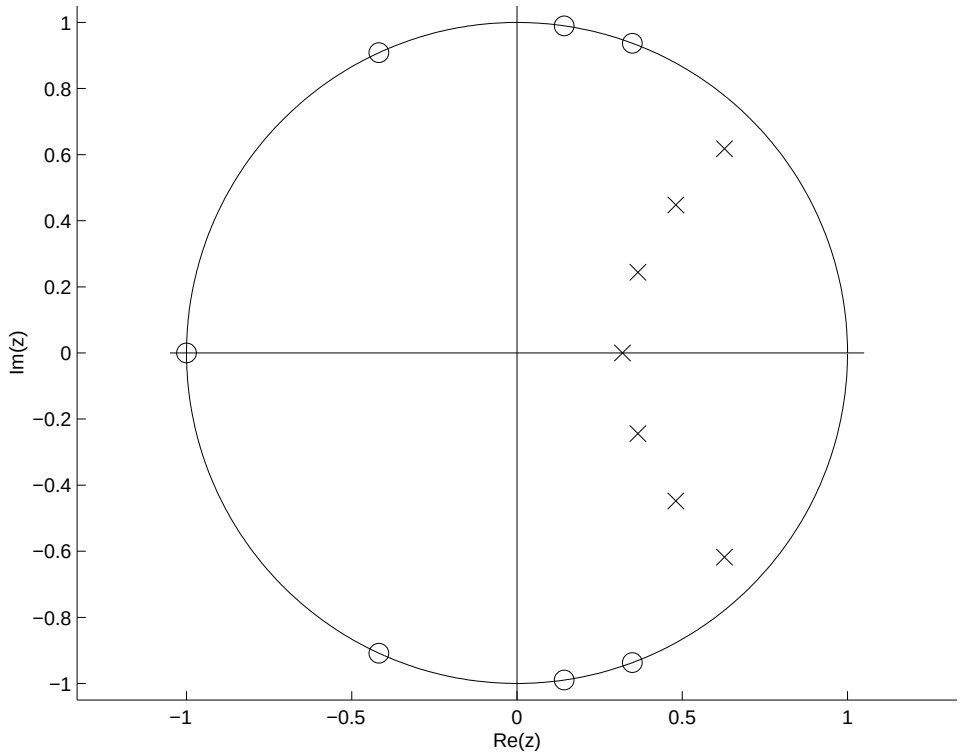

Figure 13: Pole-zero plot for the Chebyshev Type II filter satisfying Eq. (10)

1.2.4 Elliptic filters

Elliptic, or Cauer, filters have ripples in both the passband and stopband. Figure 14 plots the frequency response on a dB scale for a 5th order elliptic filter satisfying Eq. (10). On this scale, the passband ripples are not apparent, but they can been seen in the top panel of Figure 15. Figure 15 illustrates clearly that passband exactly meets the specification at Ω_p and that the maximum stopband ripples are just up to 0.001.

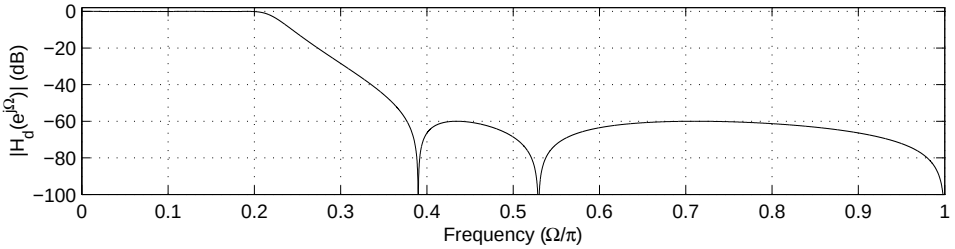

Figure 14: Frequency response magnitude $|H_d(e^{j\Omega})|$ for a 5th order Elliptic filter satisfying Eq. (10)

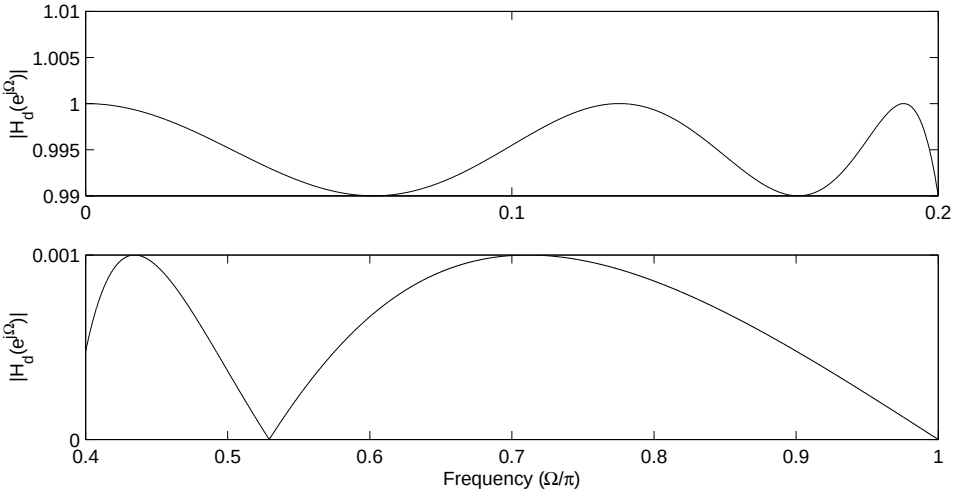

Figure 15: Zoom in on the passband and stopband for the Elliptic filter $|H_d(e^{j\Omega})|$

The Matlab commands for elliptic filters combine the features seen for the three previous filter types. The command `ellipord` returns the filter order `N` and break frequency `Omegan` needed to satisfy the specification given by the band edges Ω_p and Ω_s, and the associated ripples `Rp` and `Rs`. The Matlab command `ellip` computes the linear, constant-coefficient difference equation coefficients given `N` and `Omegan`, as well as both the passband ripple as defined for the Chebyshev Type I filter and the stopband ripple as defined for the Chebyshev Type II filter. The following commands produce the coefficients specifying the elliptic filter

```
Omegap = 0.2;
Omegas = 0.4;
delta1 = 0.01;
delta2 = 0.001;
Rp = -20*log10(1-delta1);
Rs = -20*log10(delta2);
Rpb = min(abs([20*log10(1+delta1),20*log10(1-delta1)]));
[N,Omegan] = ellipord(Omegap,Omegas,Rp,Rs);
[bellip,aellip] = ellip(N,Rp,Rs,Omegan);
```

Figure 16 plots the pole-zero plot for the elliptic filter. The pattern of poles and zeros combines features we've seen already in the Chebyshev filters. The poles generating the passband are arranged in a very shallow arc, almost in a line, similar to the arrange for the Chebyshev Type I filter which also had ripples in the passband. The zeros fall directly on the unit circle, as seen in the Chebyshev Type II filter. This fits with the stopband ripples seen in both the elliptic and Chebyshev Type II filter. Again, the angles of these poles match the frequencies of the zeros seen in the bottom panel of Figure 15.

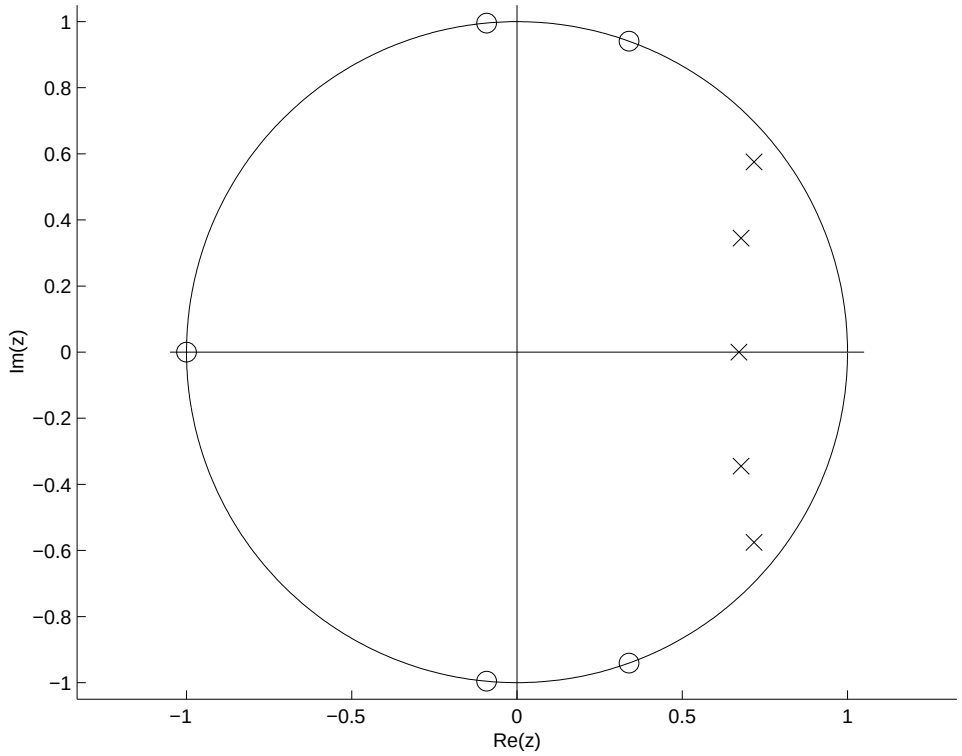

Figure 16: Pole-zero plot for the elliptic filter satisfying Eq. (10)

This should give you a good overview of the common types of IIR filter, and some basic idea of how the bilinear transform produces DT filters from CT filters.