Now we can map $H_L(s)$ from a known CT filter to DT

- We will consider three main types of filters
  - Butterworth, Chebyshev (type I/II), Elliptic

Ex: Butterworth (order n): $H_L(s)$ is obtained by taking the left-plane poles of $\dfrac{1}{1+(-s^2)^n}$

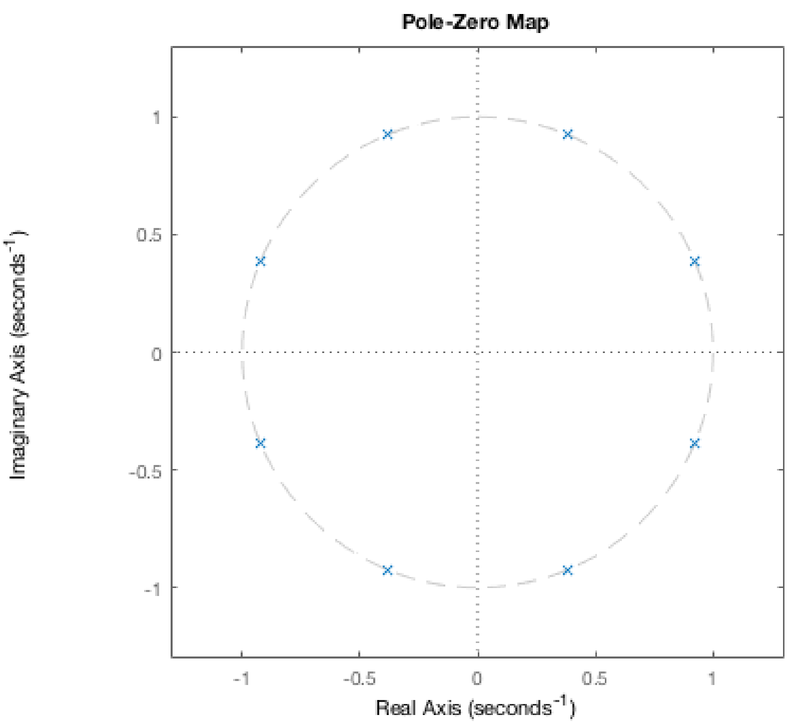
For Butterworth, $|H_L(j\Omega)|_{\Omega=1} = \dfrac{1}{\sqrt{2}}$ (-3 dB)

3 dB cutoff: $\Omega_c = 1 \Rightarrow \Omega_c = 1 = \alpha \tan\left(\dfrac{\omega_c}{2}\right)$

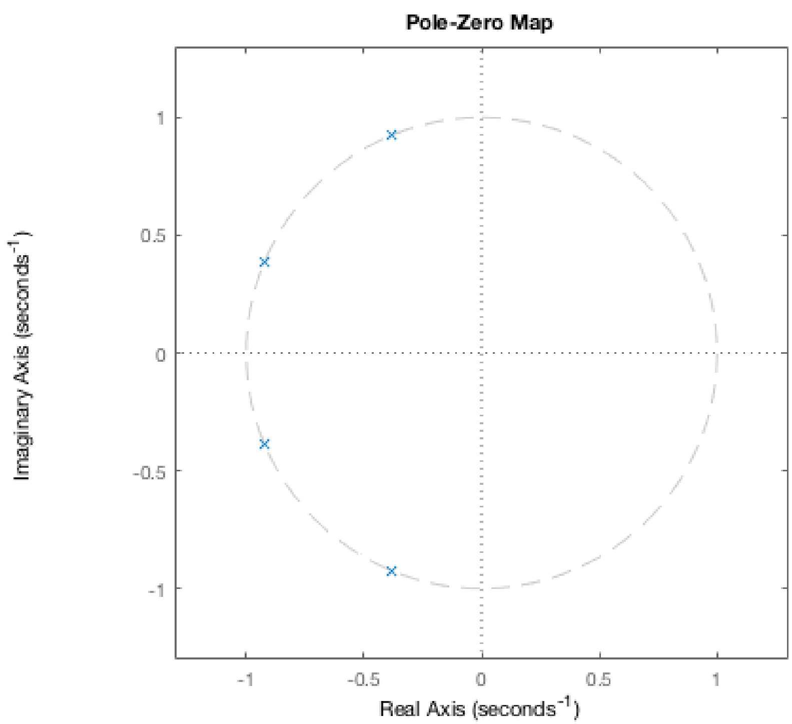
$$\Rightarrow \alpha = \frac{1}{\tan\left(\frac{\omega_c}{2}\right)}$$

```
% To design a digital Butterworth filter , we start with an analog
% Butterworth filter and apply the bilinear transformation
```

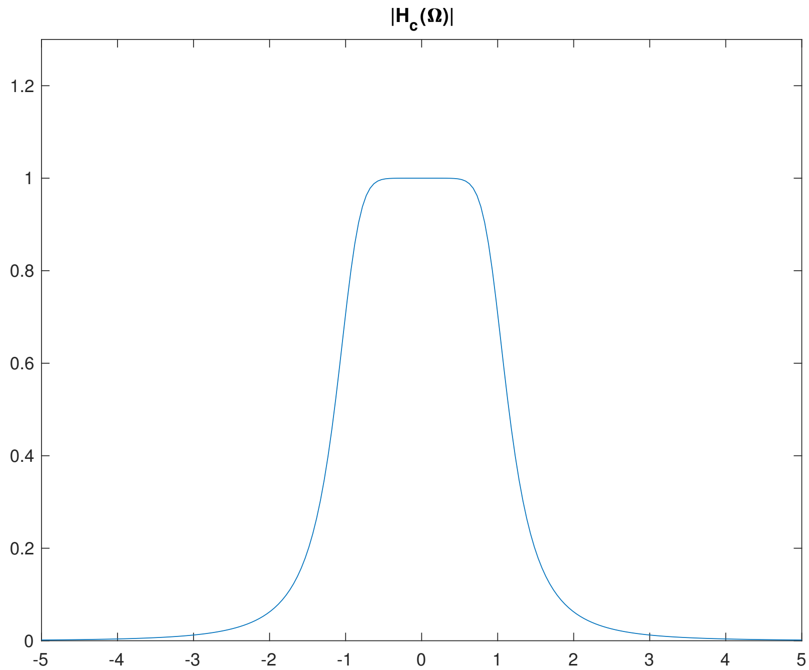
```
% To design an analog Butterworth filter of order n, we first look at the
% poles of 1/(1+(-s^2)^n)
% For n=2, this gives 1/(1+s^4), for n=3, 1/(1-s^6), and so on

% for order n = 4:
M_L = tf(1,[1,0,0,0,0,0,0,0,1]);

fig1 = figure(1); hold on; t = linspace(0,2*pi); ...
    plot(sin(t),cos(t),'--','LineWidth', 0.3,'Color', [.8,.8,.8]);
pzmap(M_L);
xlim([-1.3,1.3]); ylim([-1.3,1.3]); pbaspect([1,1,1]); hold off;
```



```
% Next, we take only the poles on the left half-plane (for stability)
[p,z] = pzmap(M_L);

% left-plane poles:
p_left = p(1:length(p)/2);

[b_CT,a_CT] = zp2tf([],p_left,1);

fig2 = figure(2); hold on; t = linspace(0,2*pi);
plot(sin(t),cos(t),'--','LineWidth', 0.3,'Color', [.8,.8,.8]);
pzmap(b_CT,a_CT);
xlim([-1.3,1.3]); ylim([-1.3,1.3]); pbaspect([1,1,1]); hold off;
```



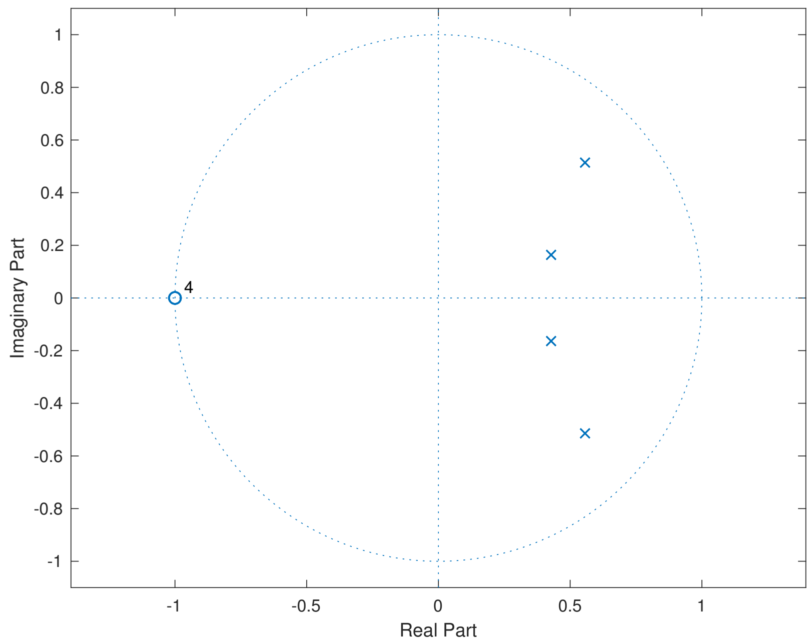
```
% Let's look at frequency response of CT Butterworth filter:
Omega = linspace(-5, 5, 200);

% H_L(s) = B_L(s)/A_L(s)
% H_c(s) = H(j*Omega)

H_c = polyval(b_CT,j*Omega)./polyval(a_CT,j*Omega);

plot(Omega,abs(H_c)); ylim([0,1.3]); title('|H_c(\Omega)|')
```




```
% Magnitude response at Omega = 1 is 1/sqrt(2) (i.e., -3dB from passband)

abs(polyval(b_CT,j*1)./polyval(a_CT,j*1))
```

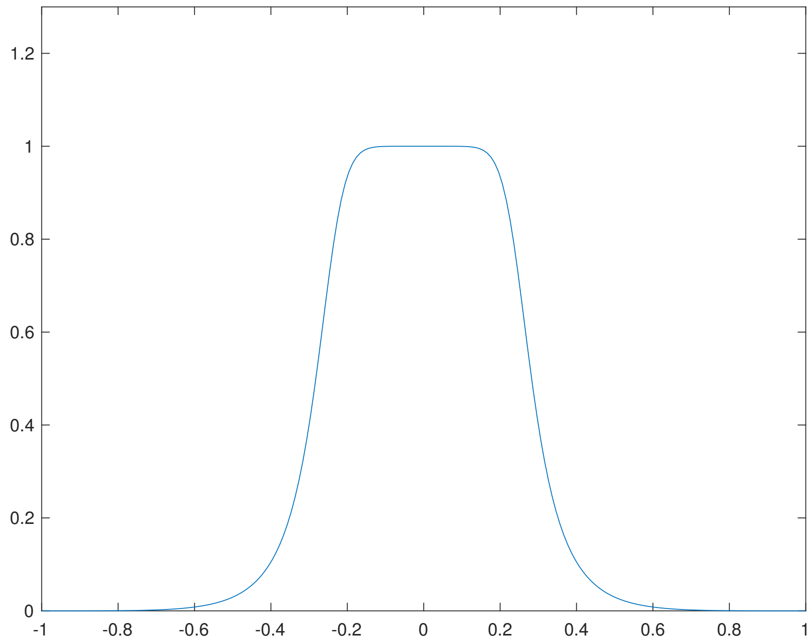
```
ans = 0.7071
```

```
% bilinear transformation

w_c = 0.25*pi;
alpha = 1/tan(w_c/2);

[zd,pd,kd] = bilinear(z,p_left,1,alpha/2);
[b,a] = zp2tf(zd,pd,kd);
zplane(b,a);
```

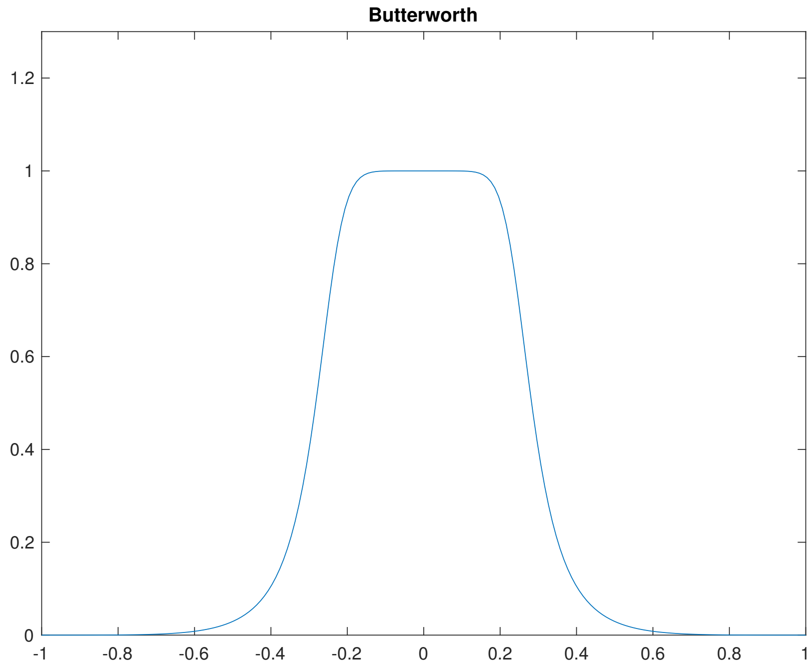


Notice how the zeros end up inside unit circle after BLT

```
% finally, we obtain the frequency response of the digital filter:

w = linspace(-pi, pi, 200);
h = freqz(b,a,w);
plot(w/pi,abs(h)); ylim([0,1.3]);
```



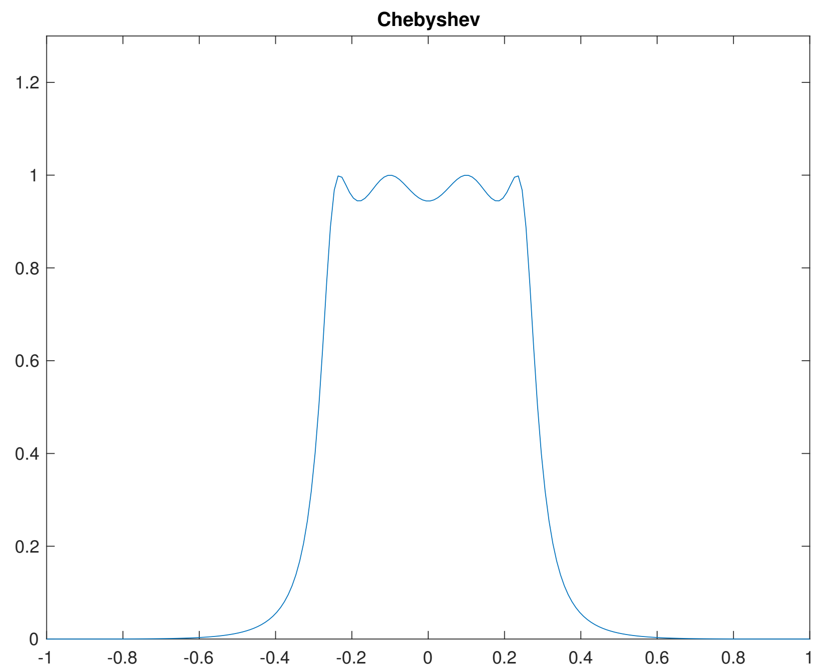

```
% In practice, we can do all this with Matlab's function butter
[bB,aB] = butter(4,0.25);
hB = freqz(bB,aB,w);

fig4 = figure(4);
plot(w/pi,abs(hB)); xlim([-1,1]); ylim([0,1.3]); title('Butterworth');
```



- Response is monotonic for omega >= 0 (no ripples!)
- Response in the passband is very flat!
- Transition band is not very narrow

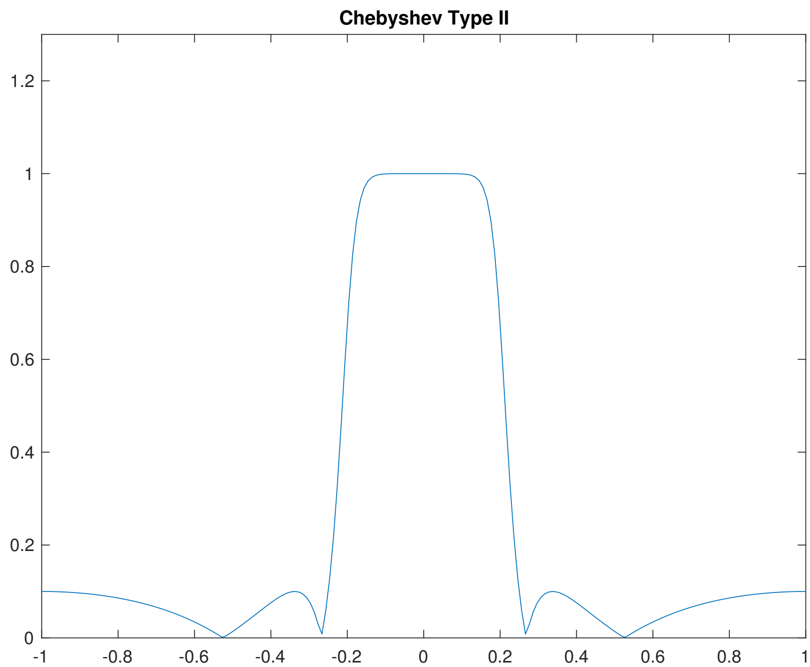
Now let's look at other IIR digital filters, Chebyshev and Elliptic

```
% Chebyshev (Type I)

[bC,aC] = cheby1(4,0.5,1/4);
hC = freqz(bC,aC,w);

fig6 = figure(6);
plot(w/pi,abs(hC)); xlim([-1,1]); ylim([0,1.3]); title('Chebyshev Type I');
```



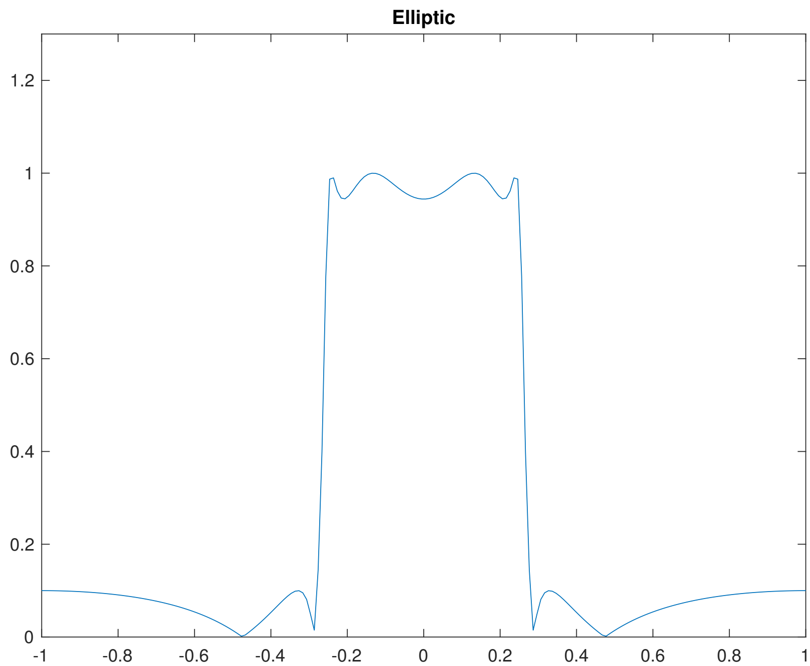
- transition band is narrower than for Butterworth filter
- equiripple in the passband
- in the stopband, Chebyshev is monotonic (no ripples)

```
% Chebyshev (Type II)

[bC2,aC2] = cheby2(4,20,1/4);
hC2 = freqz(bC2,aC2,w);

fig6 = figure(6);
plot(w/pi,abs(hC2)); xlim([-1,1]); ylim([0,1.3]); title('Chebyshev Type ...
    II');
```



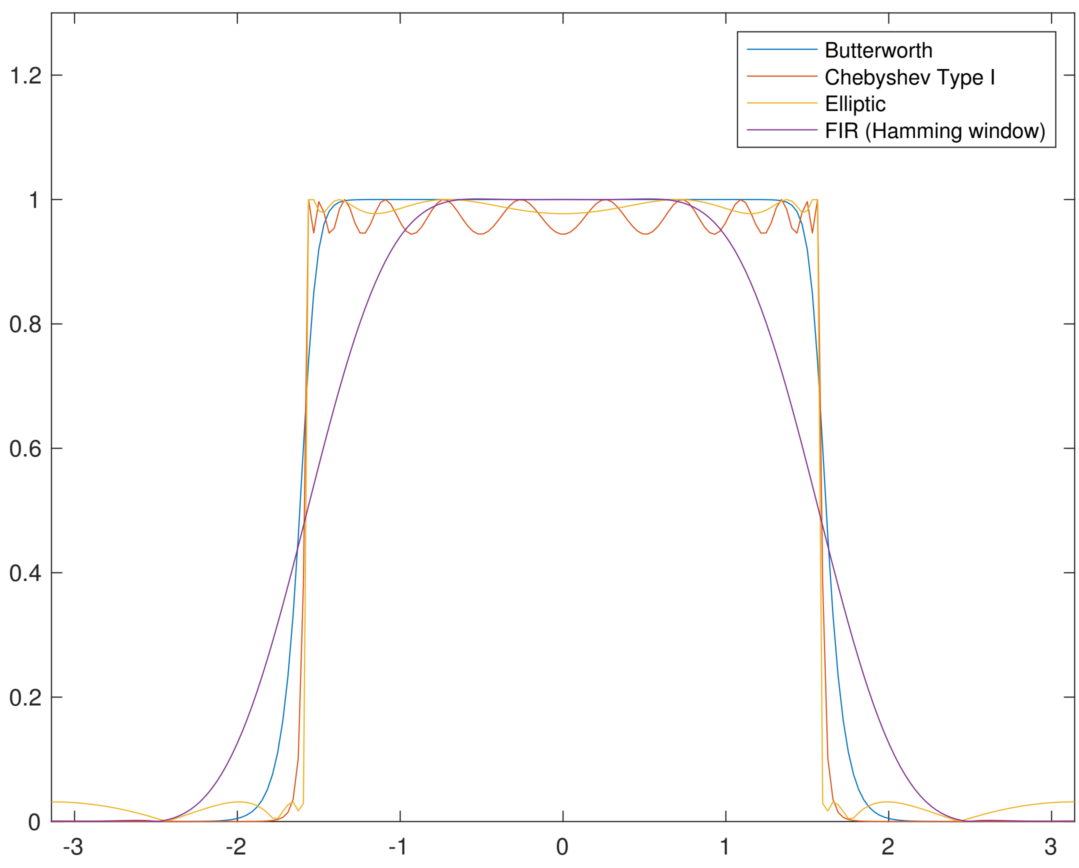
- equiripple in the stopband

```
% Elliptic filter

[bE,aE] = ellip(4,0.5,20,1/4);
hE = freqz(bE,aE,w);

fig7 = figure(7);
plot(w/pi,abs(hE)); xlim([-1,1]); ylim([0,1.3]); title('Elliptic');
```



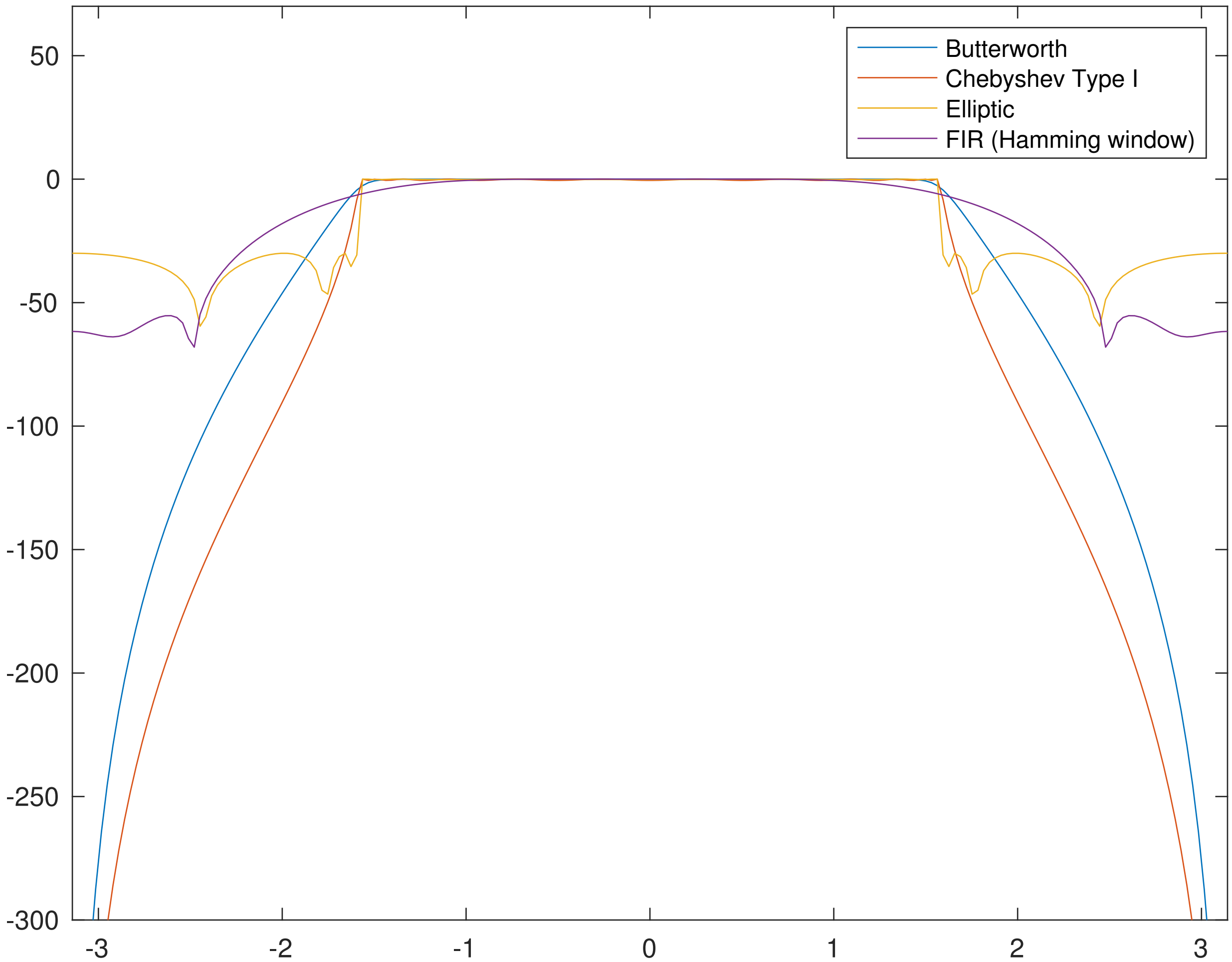
- equiripple both in stopband and in passband
- it has the narrowest transition band

```
n = 12;

[bB,aB] = butter(n,1/2);
hB = freqz(bB,aB,w);

[bC,aC] = cheby1(n,0.5,1/2);
hC = freqz(bC,aC,w);

[bE,aE] = ellip(n,0.2,30,1/2);
hE = freqz(bE,aE,w);

[bF,aF] = fir1(n,1/2,'low',hamming(n+1));
hF = freqz(bF,aF,w);

fig8 = figure(8);
plot(w,abs(hB),w,abs(hC),w,abs(hE),w,abs(hF)); ylim([0,1.3]); ...
    xlim([-pi,pi]);
legend('Butterworth','Chebyshev Type I','Elliptic','FIR (Hamming window)');
```

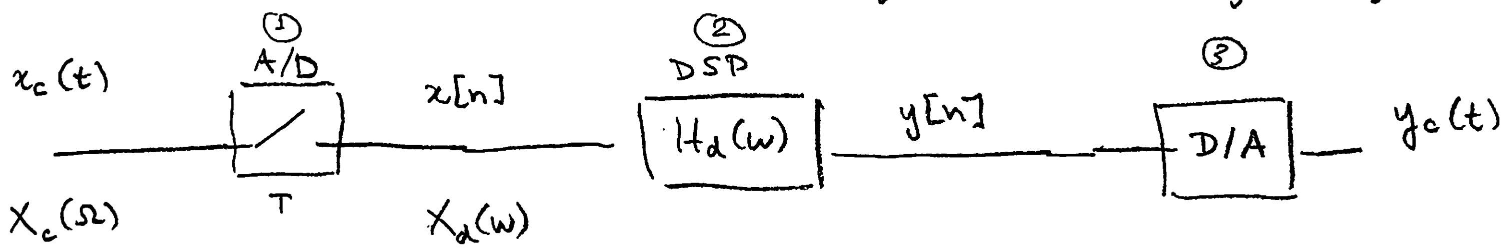


We notice that IIR filters have a much narrower transition band than an FIR filter of the same order

```
fig9 = figure(9);

plot(w,mag2db(abs(hB)),w,mag2db(abs(hC)),w,mag2db(abs(hE)),w,mag2db(abs(hF))); ...
    ylim([-300,70]); xlim([-pi,pi]);
legend('Butterworth','Chebyshev Type I','Elliptic','FIR (Hamming window)');
```



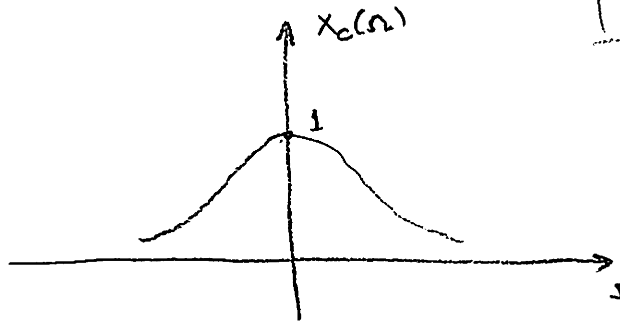
Lecture 23 - Discrete-time processing of Analog signals

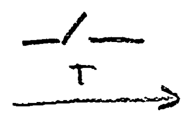


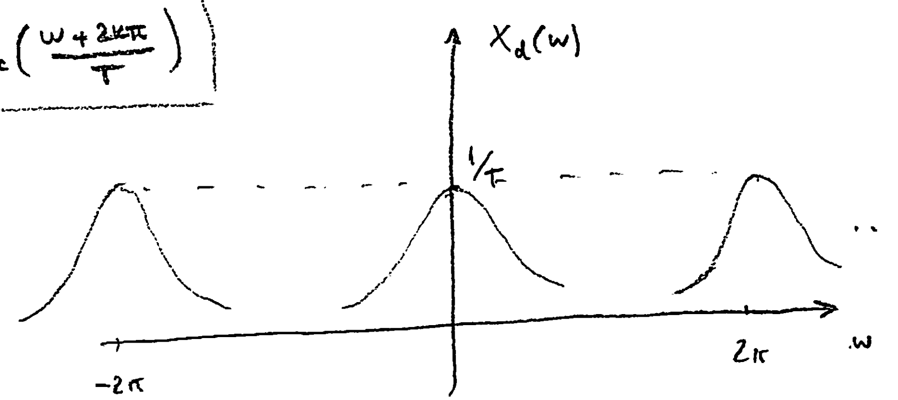
How does the analog response (i.e. $H_c(\Omega) = \frac{Y_c(\Omega)}{X_c(\Omega)}$) depend on $H_d(\omega)$?

① Recall (lecture 14):

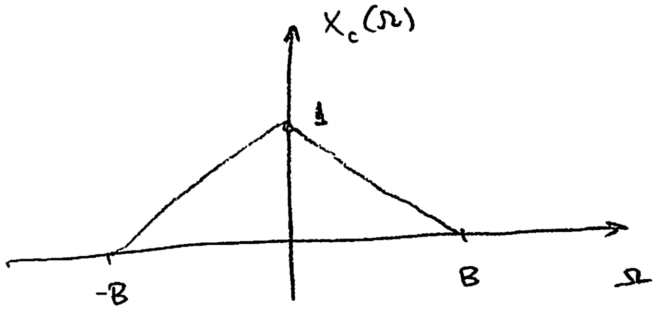
$$\boxed{X_d(\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega + 2k\pi}{T}\right)}$$

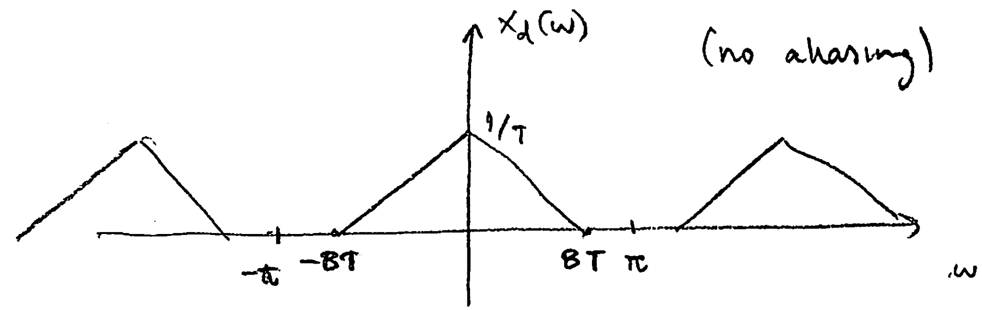






In particular, if $x_c(t)$ is bandlimited and $T < \frac{\pi}{B}$ $\left(F_s > 2 \cdot \frac{B}{2\pi}\right)$ (Nyquist condition)

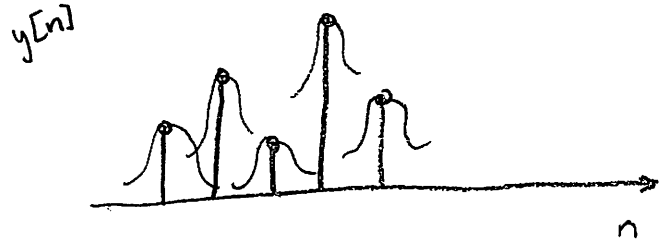




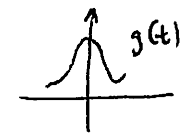
② After filtering with $H_d(\omega)$:

$$Y_d(\omega) = X_d(\omega) \cdot H_d(\omega) = H_d(\omega) \cdot \frac{1}{T} \sum_k X_c\left(\frac{\omega + 2k\pi}{T}\right)$$

③ Final step: D/A
(interpolation)



interpolating function



$$y_c(t) = \sum_{n=-\infty}^{\infty} y[n]\, g(t - nT)$$

We showed that $\boxed{Y_c(\Omega) = G(\Omega) \cdot Y_d(\Omega T)}$ (lecture 14)

Finally: $\boxed{Y_c(\Omega) = G(\Omega) \cdot H_d(\Omega T) \cdot \frac{1}{T} \sum_k X_c\left(\Omega + \frac{2k\pi}{T}\right)}$

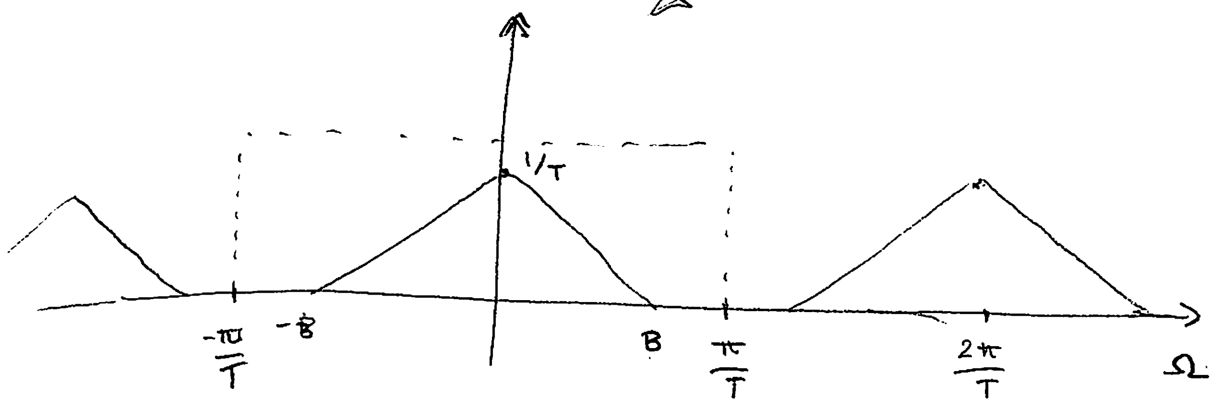
In general, solving for $\dfrac{Y_c(\Omega)}{X_c(\Omega)}$ is hard.

But it simplifies if we assume:

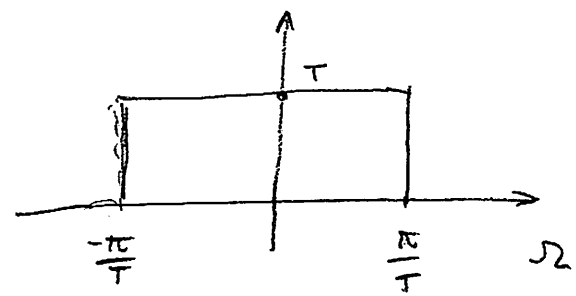
- $x_c(t)$ is band limited
- Nyquist sampling $(T < \pi/B)$
- Ideal D/A

If $T < \pi/B$: (no aliasing)

$$Y_c(\Omega) = H_d(\Omega T) \cdot G(\Omega) \boxed{\frac{1}{T}\sum_k X_c\left(\Omega + \frac{2k\pi}{T}\right)}$$



Ideal D/A



$$\Rightarrow \quad Y_c(\Omega) = H_d(\Omega T) \cdot X_c(\Omega)$$

} connection between analog and digital filtering

$$H_c^{(\text{effective})}(\Omega) = H_d(\Omega T) \quad \text{for } |\Omega| \le \frac{\pi}{T}$$

$\omega = \Omega T$

$$H_c^{(\text{effective})}\left(\frac{\omega}{T}\right) = H_d(\omega)$$

Ex: Differentiator $\quad y_c(t) = \frac{d}{dt} x_c(t)$

$$\Longleftrightarrow \quad Y_c(\Omega) = \underbrace{(j\Omega)}_{H_c(\Omega)} X_c(\Omega)$$

So we want an analog filter with $H_c(\Omega) = j\Omega$

$\Rightarrow$ Digital design $\quad H_d(\omega) = H_c\left(\frac{\omega}{T}\right) = j\frac{\omega}{T}$