2020-09-11/MS

Signal Processing, lecture 2

Suggested solutions to exercises:

2.1

A Chebyshev prototype filter is (usually) normalized to a ripple bandwidth of 1 rad/s.

a. Show that the 3 dB bandwidth can be found by:

$$\omega_{3dB} = \cosh\left(\frac{1}{n}\cosh^{-1}\frac{1}{\varepsilon}\right)$$

Hints:

$$\varepsilon^2 C_n^2(\omega_{3dB}) = 1 \quad and$$

$$C_n(\omega) = \cosh(n \cdot \cosh^{-1}\omega) \quad for \quad \omega > 1$$

$$|H(j\omega_{3dB})|^2 = \frac{1}{2} = \frac{1}{1+\varepsilon^2 C_n^2(\omega_{3dB})}$$

$$\varepsilon^2 C_n^2(\omega_{3dB}) = 1$$

$$C_n(\omega_{3dB}) = \frac{1}{\varepsilon}$$

$$C_n(\omega_{3dB}) = \cosh(n \cdot \cosh^{-1}\omega_{3dB}) = \frac{1}{\varepsilon}$$

$$\omega_{3dB} = \cosh\left(\frac{1}{n}\cosh^{-1}\frac{1}{\varepsilon}\right)$$

b. Find the 3-dB bandwidth for a 4th order filter with a 0.5 dB ripple bandwidth of 1 rad/s.

$$\varepsilon = \sqrt{10^{0.5/10}-1} = 0.35$$

$$\omega_{3dB} = \cosh\left(\frac{1}{4}\cosh^{-1}\frac{1}{0.35}\right) = 1.093$$

2.2

The requirements for a Chebyshev low-pass filter are:

- Passband ripple: 0.5 dB
- Ripple bandwidth: 20 kHz
- The attenuation at 190 kHz shall be at least 30 dB

a. Find the frequency scaling factor, k_f, and the necessary filter order, n

mish@es.aau.dk Ming Shen / Ole Kiel Jensen 9940 8677

$$k_f = \frac{2\pi 20e3}{1} = 125664 \qquad \omega_{norm} = \frac{190\ kHz}{20\ kHz} = 9.5$$

$$n \geq \frac{1}{\cosh^{-1}\omega_{norm}} \cosh^{-1}\sqrt{\frac{10^{\alpha_{S,dB}/10}-1}{10^{Ripple_{dB}/10}-1}}$$

$$\underline{n} \geq \frac{1}{\cosh^{-1} 9.5} \cosh^{-1}\sqrt{\frac{10^{30/10}-1}{10^{0.5/10}-1}} = 1.767 \rightarrow \underline{2}$$

b. Find (analytically) the actual attenuation at 190 kHz

$$\varepsilon = \sqrt{10^{Ripple_{dB}/10}-1} = \sqrt{10^{0.5/10}-1} = 0.3493$$

$$C_n(\omega_{norm}) = \cosh(n \cdot \cosh^{-1}\omega_{norm}) = \cosh(2\cdot\cosh^{-1} 9.5) = 179.5$$

$$H(j\omega_{norm})_{dB} = -10\log(1+\varepsilon^2 C_n^2(\omega)) = -10\log(1+0.3493^2 179.5^2) = 35.95\ dB$$

c. Compare the results with Exercise 1.2. *The attenuation of the Chebyshev filter is 6 dB higher than of the Butterworth filter.*

d. Find (analytically) the poles of the prototype filter.

$$s_k = \sin\frac{(2k-1)\pi}{2n}\cdot\sinh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\varepsilon}\right) + j\cos\left(\frac{(2k-1)\pi}{2n}\right)\cdot\cosh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\varepsilon}\right)$$

$k = n+1 \ldots 2n$

$$s_3 = \sin\frac{(6-1)\pi}{4}\cdot\sinh\left(\frac{1}{2}\sinh^{-1}\frac{1}{0.349}\right) + j\cos\left(\frac{(6-1)\pi}{4}\right)\cdot\cosh\left(\frac{1}{2}\sinh^{-1}\frac{1}{0.349}\right)$$

$$= -0.713 - j1.004$$

$$s_4 = \sin\frac{(8-1)\pi}{4}\cdot\sinh\left(\frac{1}{2}\sinh^{-1}\frac{1}{0.349}\right) + j\cos\left(\frac{(8-1)\pi}{4}\right)\cdot\cosh\left(\frac{1}{2}\sinh^{-1}\frac{1}{0.349}\right)$$

$$= -0.713 + j1.004$$

e. Check the results of d. with Matlab

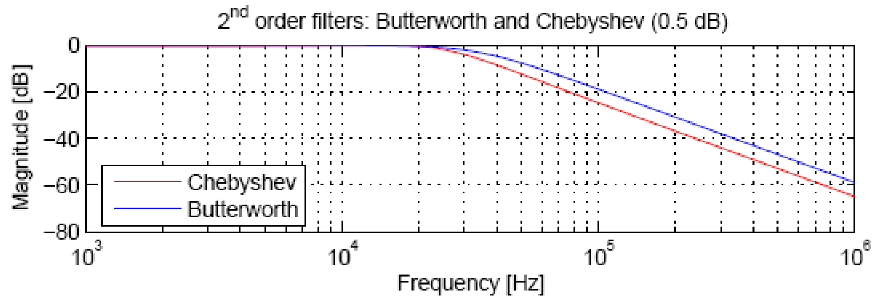
```
>> % Chebyshev poles:
Order = 2;
Ripple_dB = 0.5;
[dummy ChePoles K] = cheb1ap(Order,Ripple_dB);
ChePoles
ChePoles =
  -0.7128 + 1.0040i
  -0.7128 - 1.0040i
```

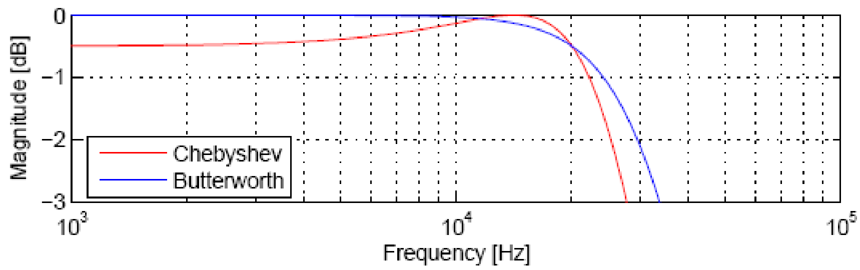
f. Use Matlab to plot the transfer function (1 kHz - 1 MHz) and the group delay together with the filter from Exercise 1.2. Note that the frequency scaling factors, k_f, are different for the two filters. Hints: There are many ways to do this. A possible way is shown in Exerc2_2_template.m where $H_{Scaled}(j2\pi f) = H_{Norm}(j\omega_{Norm})$ and $\omega_{Norm} = \dfrac{2\pi f}{k_f}$.

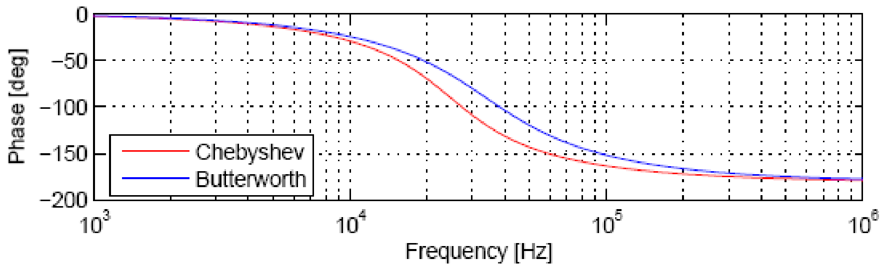
```
% Exerc2_2.m            070731/OKJ

clear;
```

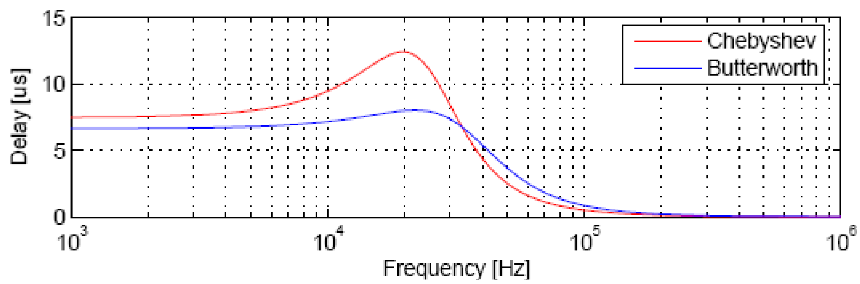
```
% Chebyshev poles:
Order = 2;
Ripple_dB = 0.5;
[dummy ChePoles K] = cheb1ap(Order,Ripple_dB);
ChePoles

% TRANSFER FUNCTIONS:

% Frequency definitions:
freq = logspace(3,6,3000);       % Freq. for plot
kfCheb = 2*pi*20e3/1;            % Freq. scaling factor
kfBut = 212600;                  % Freq. scaling factor from exercise 1.2
jomNcheb = j*2*pi*freq/kfCheb;   % j*normalized radian frequency
jomNbut = j*2*pi*freq/kfBut;

% Chebyshev:
Wcut = 1;                                        % Ripple bandwidth [rad/s] for
the normalized filter
[b a] = cheby1(Order,Ripple_dB,Wcut,'s');        % 's' indicates analog filter
Hcheb = polyval(b,jomNcheb)./polyval(a,jomNcheb);   % Insert j*omega in the trans-
fer function
Hcheb_dB = 20*log10(abs(Hcheb));
Hcheb_deg = 180/pi*angle(Hcheb);
Hcheb_del(2:1:length(freq)) = (angle(Hcheb(1:1:length(freq)-1))-
angle(Hcheb(2:1:length(freq))))/2/pi./(freq(2:1:length(freq))-
freq(1:1:(length(freq)-1)));
Hcheb_del(1) = Hcheb_del(2);                     % Cheating with the first
point

% Butterworth:
Wcut = 1;                                        % 3 dB bandwidth [rad/s] for
the normalized filter
[b a] = butter(Order,Wcut,'s')                   % 's' indicates analog filter
Hbut = polyval(b,jomNbut)./polyval(a,jomNbut);   % Insert j*omega in the trans-
fer function
Hbut_dB = 20*log10(abs(Hbut));
Hbut_deg = 180/pi*angle(Hbut);
Hbut_del(2:1:length(freq)) = (angle(Hbut(1:1:length(freq)-1))-
angle(Hbut(2:1:length(freq))))/2/pi./(freq(2:1:length(freq))-
freq(1:1:(length(freq)-1)));
Hbut_del(1) = Hbut_del(2);                       % Cheating with the first
point

figure(1);
subplot(4,1,1);
semilogx(freq,Hcheb_dB,'r',freq,Hbut_dB,'b');
grid;
xlabel('Frequency [Hz]');
ylabel('Magnitude [dB]');
legend('Chebyshev','Butterworth','Location','SouthWest');
title('2^n^d order filters: Butterworth and Chebyshev (0.5 dB)');

subplot(4,1,2);
semilogx(freq,Hcheb_dB,'r',freq,Hbut_dB,'b');
set(gca,'Ylim',[-3 0]);
grid;
xlabel('Frequency [Hz]');
ylabel('Magnitude [dB]');
legend('Chebyshev','Butterworth','Location','SouthWest');

subplot(4,1,3);
semilogx(freq,Hcheb_deg,'r',freq,Hbut_deg,'b');
grid;
xlabel('Frequency [Hz]');
ylabel('Phase [deg]');
legend('Chebyshev','Butterworth','Location','SouthWest');

subplot(4,1,4);
semilogx(freq,Hcheb_del*1e6,'r',freq,Hbut_del*1e6,'b');
grid;
xlabel('Frequency [Hz]');
ylabel('Delay [us]');
```

```
legend('Chebyshev','Butterworth','Location','NorthEast');

%  An easier way to make bode-plots (excl. delay):
% Order = 2;
% Ripple_dB = 0.5;
% Wcut = 2*pi*20e3;
% [NumPoly DenomPoly] = cheby1(Order,Ripple_dB,Wcut,'s');
% ChebSys = tf(NumPoly,DenomPoly);
% bode(ChebSys,{2*pi*1e3,2*pi*1e6});
% grid;
% % In the bode-plot window:
% %   Right-cleck on the axes, choose Properties > Units > Frequency in Hz
```









2.3

A low-pass prototype filter has the transfer function shown.

$$H_{LPP}(s) = \frac{0.423}{(s+0.446)(s^2+0.446s+0.949)}$$

a. Find the location of the poles.
b. Determine which types of filter it is (and explain your conclusion):
   - Butterworth?
   - Chebyshev?

a. *Poles:*

$$s + 0.446 = 0 \Leftrightarrow s = -0.446$$

$$s^2 + 0.446s + 0.949 = 0 \Leftrightarrow s = \frac{-0.446 \pm \sqrt{0.446^2 - 4 \cdot 1 \cdot 0.949}}{2} = \begin{cases} -0.223 + j0.948 \\ -0.223 - j0.948 \end{cases}$$

$$\underline{\underline{Poles = \begin{cases} -0.223 + j0.948 \\ -0.446 \\ -0.223 - j0.948 \end{cases}}}$$

b. *Type of filter:*
- *The poles of a Butterworth filter are located at the unit circle. This is not the case here, so Butterworth is excluded. (Ref. Slide 1:20 or p. 54-55)*
- *Only possibility: Chebyshev.*
- *PS: The poles of a Chebyshev filter are located on an ellipse. It can be shown, that in this case the poles are on an ellipse with semi-axes 0.446 and j1.124. The transfer function corresponds to a Chebyshev-filter with 1.3 dB ripple.*