Homework #2
By: Zachary Sweep
2/9/18

1. **Starting from the definition of the Chebyshev function**

$$\mathbf{C}_N(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) = \begin{cases} \boldsymbol{cos}(\boldsymbol{N}\ \mathbf{cos}^{-1}\boldsymbol{x}), & |\boldsymbol{x}| \le \mathbf{1} \\ \boldsymbol{cosh}(\boldsymbol{N}\ \mathbf{cosh}^{-1}\boldsymbol{x}), & |\boldsymbol{x}| > \mathbf{1} \end{cases}$$

Show that $\boldsymbol{C_N(x)}$ obeys the following recursion: $\boldsymbol{C_{N+1}(x) = 2xC_N(x) - C_{N-1}(x)}$. Consider the cases $|\mathbf{x}| \le \mathbf{1}$ and $|\mathbf{x}| > \mathbf{1}$ separately.

For $|x| \le 1$:

$$C_N(x) = \cos(N\cos^{-1}x)$$

$$\begin{aligned} C_{N+1}(x) = \cos\left((N+1)\cos^{-1}x\right) &= cos(Ncos^{-1}x + cos^{-1}x) \\ &= cos(Ncos^{-1}x)cos(cos^{-1}x) - sin(Ncos^{-1}x)sin(cos^{-1}x) \\ &= cos(Ncos^{-1}x)x - sin(Ncos^{-1}x)sin(cos^{-1}x) \end{aligned}$$

$$\begin{aligned} C_{N-1}(x) = \cos\left((N-1)\cos^{-1}x\right) &= cos(Ncos^{-1}x - cos^{-1}x) \\ &= cos(Ncos^{-1}x)cos(cos^{-1}x) + sin(Ncos^{-1}x)sin(cos^{-1}x) \\ &= cos(Ncos^{-1}x)x + sin(Ncos^{-1}x)sin(cos^{-1}x) \end{aligned}$$

$$\begin{aligned} C_{N+1}(x) + C_{N-1}(x) = {} & cos(Ncos^{-1}x)x - \cancel{sin(Ncos^{-1}x)sin(cos^{-1}x)} \\ & + cos(Ncos^{-1}x)x + \cancel{sin(Ncos^{-1}x)sin(cos^{-1}x)} \end{aligned}$$

$$C_{N+1}(x) + C_{N-1}(x) = 2x * cos(Ncos^{-1}x)$$

$$C_{N+1}(x) = 2x * cos(Ncos^{-1}x) - C_{N-1}(x) = 2xC_N(x) - C_{N-1}(x)$$

For $|x| > 1$:

$$C_N(x) = \cosh(N\cosh^{-1}x):$$

$$\begin{aligned} C_{N+1}(x) = \cosh\left((N+1)\cosh^{-1}x\right) &= cosh(Ncosh^{-1}x + cosh^{-1}x) \\ &= cosh(Ncosh^{-1}x)cosh(cosh^{-1}x) + sinh(Ncosh^{-1}x)sinh(cosh^{-1}x) \\ &= cosh(Ncosh^{-1}x)x + sinh(Ncosh^{-1}x)sinh(cosh^{-1}x) \end{aligned}$$

$$\begin{aligned} C_{N-1}(x) = \cosh\left((N-1)\cosh^{-1}x\right) &= cosh(Ncosh^{-1}x - cosh^{-1}x) \\ &= cosh(Ncosh^{-1}x)cosh(cosh^{-1}x) - sinh(Ncosh^{-1}x)sinh(cosh^{-1}x) \\ &= cosh(Ncosh^{-1}x)x - sinh(Ncosh^{-1}x)sinh(cosh^{-1}x) \end{aligned}$$

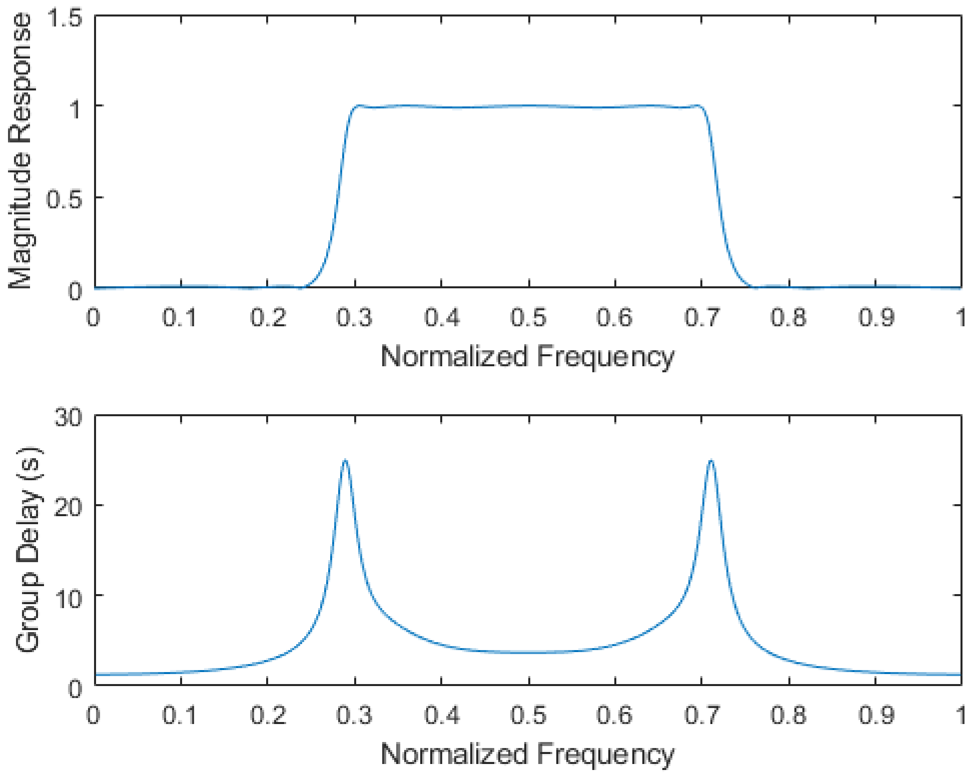
$$\begin{aligned} C_{N+1}(x) + C_{N-1}(x) & \\ &= cosh(Ncosh^{-1}x)x + \cancel{sinh(Ncosh^{-1}x)sinh(cosh^{-1}x)} \\ &+ cosh(Ncosh^{-1}x)x - \cancel{sinh(Ncosh^{-1}x)sinh(cosh^{-1}x)} \end{aligned}$$

$$C_{N+1}(x) + C_{N-1}(x) = 2x * cosh(Ncosh^{-1}x)$$

$$C_{N+1}(x) = 2x * cosh(Ncosh^{-1}x) - C_{N-1}(x) = 2xC_N(x) - C_{N-1}(x)$$

2. **Design a bandpass discrete-time Elliptic filter with the following specifications: $\omega_{s1} = 0.2\pi$, $\omega_{p1} = 0.3\pi$, $\omega_{p2} = 0.7\pi$, $\omega_{s2} = 0.8\pi$, $G_p = 0.99$, $G_s = 0.01$. Plot the Magnitude response and group delay for the resulting filter design. Realize the transfer function as a cascade of first-order and second-order sections with real-valued coefficients. List the coefficients for each section.**

```
ws1 = 0.2*pi;
wp1 = 0.3*pi;
wp2 = 0.7*pi;
ws2 = 0.8*pi;
Gp  = 0.99;
Gs  = 0.01;

Rp = -20*log10(Gp);
Rs = -20*log10(Gs);

% Calculating digital Elliptic filter order and new w using normalized pass
% band and stop band frequencies.
[N,wpnew] = ellipord([wp1/pi wp2/pi],[ws1/pi ws2/pi],Rp,Rs);

% Calculate Zeros, Poles, and Gain for digital Elliptic filter
[z,p,k]   = ellip(N,Rp,Rs,wpnew);

% Calculate Z domain numerator polynomial
Bz = poly(z);

% Calculate Z domain denominator polynomial
Az = poly(p);

% Adjust for gain using k computed in line 24
Bz = k*Bz;

% Calculating Magnitude Response
[H,wh]   = freqz(Bz,Az);

% Calculating Group Delay
[Gpd,wg] = grpdelay(Bz,Az);

% Plotting Magnitude Response and Group Delay
figure(1)
subplot(211)
plot(wh/pi,abs(H),'-')
xlabel('Normalized Frequency'); ylabel('Magnitude Response');
subplot(212)
plot(wg/pi,Gpd,'-')
xlabel('Normalized Frequency'); ylabel('Group Delay (s)');
```

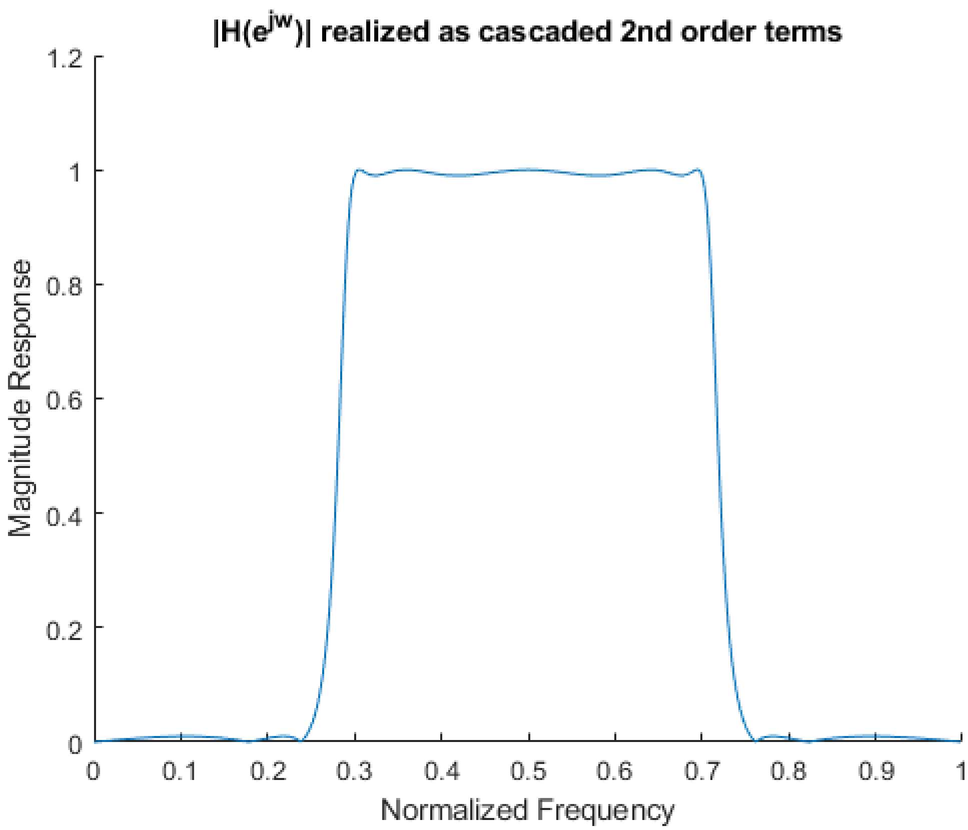


Realize transfer function as cascaded 2nd order terms

The 2nd Order terms were acquired by using MATLAB poly function on complex-conjugate pairs for variables z (zeros) and p (poles).

```
% 2nd Order Numerator coefficients indexed as follows:
% Z^2*p(1) + Z*p(2) + p(3)
z0 = [1 0 -1];
z1 = [1 -1.6954 1];
z2 = [1 1.6954 1];
z3 = [1 1.4663 1];
z4 = [1 -1.4663 1];

% 2nd Order Denominator coefficients indexed as follows:
% Z^2*p(1) + Z*p(2) + p(3)
p0 = [1 0 0.3335];
p1 = [1 -0.8406 0.6311];
p2 = [1 0.8406 0.6311];
p3 = [1 1.1701 0.9050];
p4 = [1 -1.1701 0.9050];

% Frequency response of each term
h0 = freqz(z0,p0);
h1 = freqz(z1,p1);
h2 = freqz(z2,p2);
h3 = freqz(z3,p3);
```

```
[h4,ww] = freqz(z4,p4);

% Cascading 2nd order Transfer functions
% Compensating for Gain
H_cascaded = k*(h0.*h1.*h2.*h3.*h4);

figure(2)
hold on
title('|H(e^{jw})| realized as cascaded 2nd order terms');
xlabel('Normalized Frequency'); ylabel('Magnitude Response');
plot(ww/pi, abs(H_cascaded),'-');
```



Published with MATLAB® R2017b

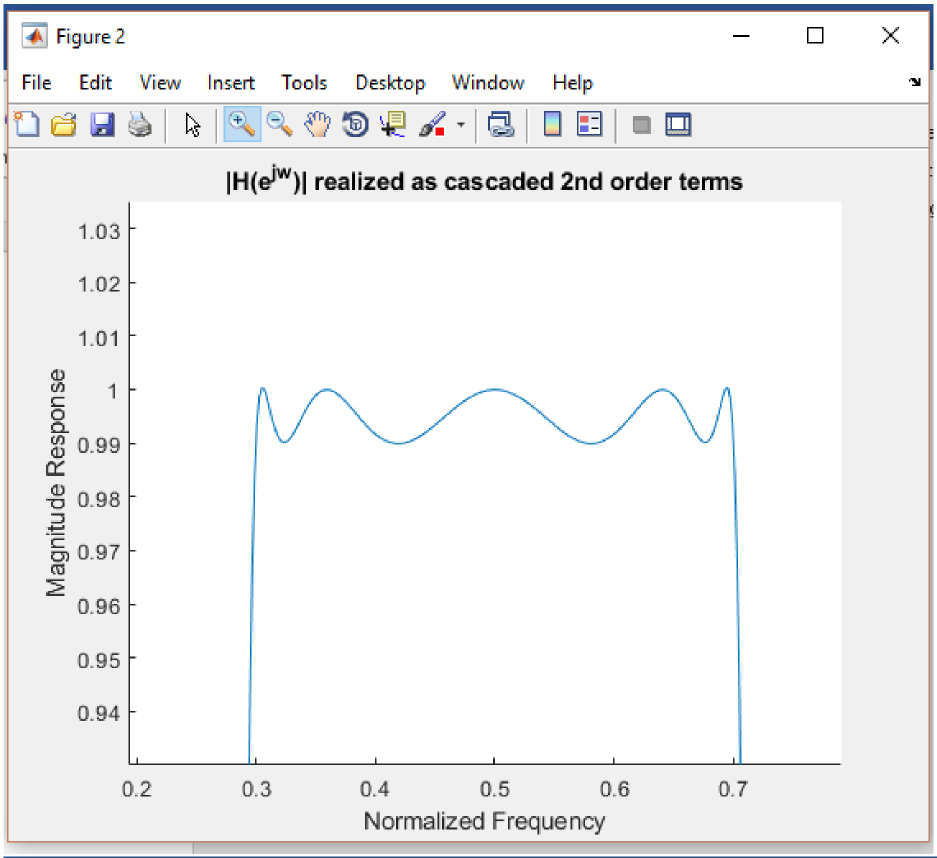


Figure 2
File Edit View Insert Tools Desktop Window Help
|H(e^jw)| realized as cascaded 2nd order terms
Magnitude Response
1.03
1.02
1.01
1
0.99
0.98
0.97
0.96
0.95
0.94
0.2 0.3 0.4 0.5 0.6 0.7
Normalized Frequency

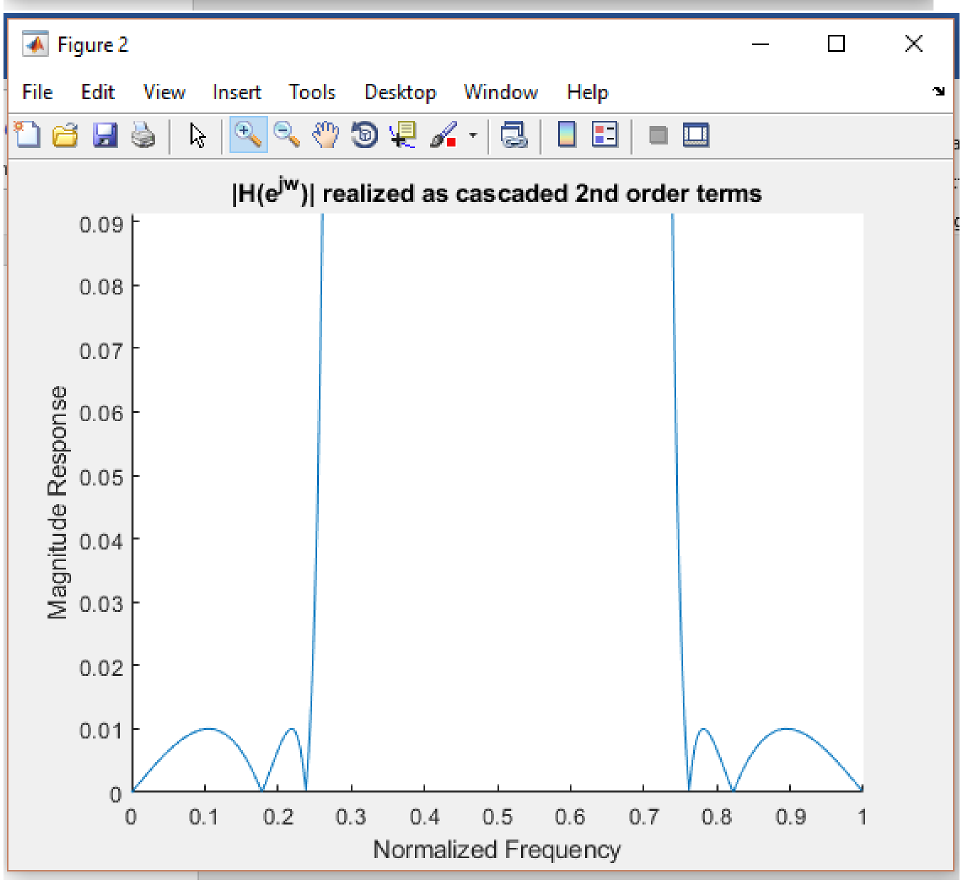


Figure 2
File Edit View Insert Tools Desktop Window Help
|H(e^jw)| realized as cascaded 2nd order terms
Magnitude Response
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency