% EE253
% DSP I
% --

Prof. Essam Marouf
Matlab Handout # 4

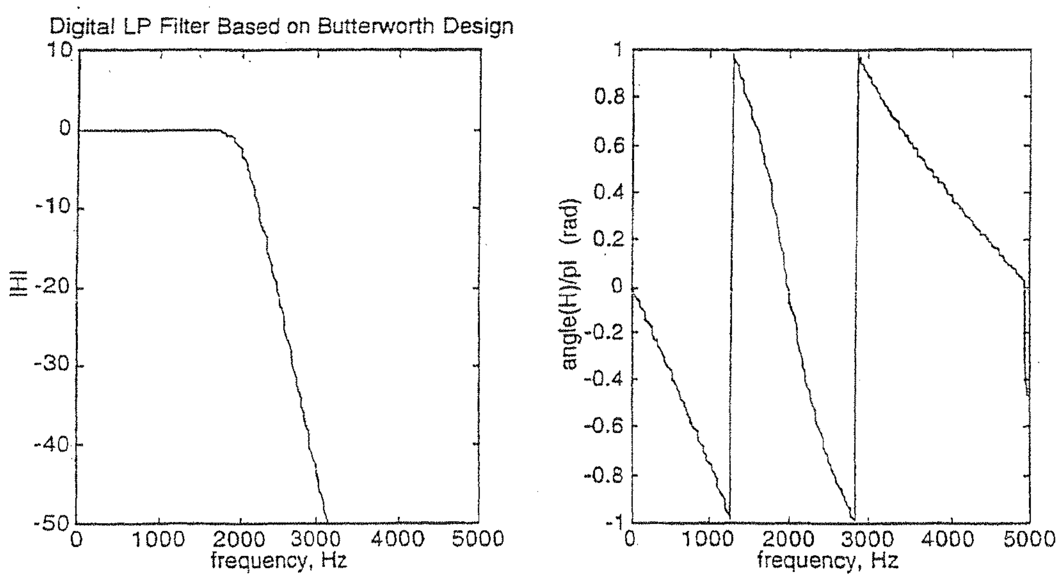
Design of a Digital LP Filers Based on Analog Prototypes

I. IIR Filter Design Based on Analog Butterworth Filters

A. Using Matlab as a Calculator

```
% Script File IIR_Analog_1.m
                                % Digital Filter Specificatons
f1= 2E3;                        % passband edge in Hz
f2= 3E3;                        % stopband edge in Hz
f_sam=10E3;                     % sampling frequency in Hz
del_1 =1-1/sqrt(2);             % peak ripple in passband
del_2=0.01;                     % peak ripple in the stopband

                                % Digital Frequencies
w1=2*pi*f1/f_sam                % w1 =  1.2566 rad
w2=2*pi*f2/f_sam                % w2 =  1.8850 rad

                                % Prewarp the w's to get analog filter cutoff freqs.
W1=2*tan(w1/2)                  % W1 =  1.4531 rad/s
W2=2*tan(w2/2)                  % W2 =  2.7528 rad/s
                                % Determine Analog filter Wc and N
Wc=W1;                          % W1 is specoified to be the 3 dB drop
A2=-20*log10(del_2);            % stopband attenuation in dB

                                % Filter order from lecture notes
N=ceil( log10(10^(A2/10) -1)/(2*log10(W2/Wc)))
                                % N = 8

                                % Determine Poles and Zeros
                                % Analog filter poles in the s-plane (see lecture notes)
k=[0:1:N-1];
v= Wc*exp(j*(pi/2 + pi/(2*N)+ pi*k/N))'
                                % v =
                                % -0.2835 - 1.4252i
                                % -0.8073 - 1.2082i
                                % -1.2082 - 0.8073i
                                % -1.4252 - 0.2835i
                                % -1.4252 + 0.2835i
                                % -1.2082 + 0.8073i
                                % -0.8073 + 1.2082i
                                % -0.2835 + 1.4252i

                                % Map the poles to the z-plane
d=(1+v/2)./(1-v/2)

                                % d =
                                % 0.2607 - 0.7868i
                                % 0.2022 - 0.5174i
                                % 0.1726 - 0.2951i
                                % 0.1599 - 0.0960i
                                %  0.1599 + 0.0960i
                                % 0.1726 + 0.2951i
                                % 0.2022 + 0.5174i
                                %  0.2607 + 0.7868i
```

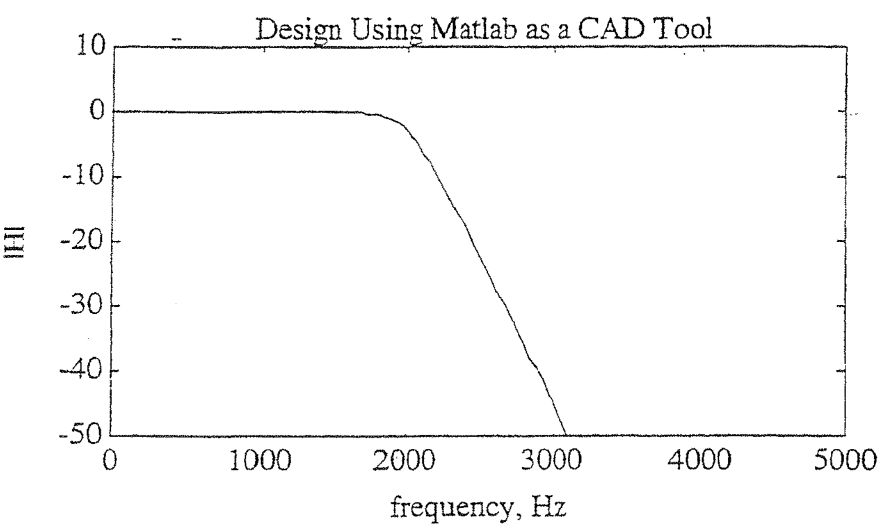
```
                                        % all zeros in the z-plane are at z=-1
c=-ones(size(k));
%
%                                       % Group poles and zeros in at atmost biquad sections

% Skipped here; see similar examples in lecture notes and homework

                                        % Check H(z)
b=poly(c)                               % b =
                                        % 1    8   28   56   70   56   28    8    1
a=real(poly(d))
                                        % a =
                                        % 1.0000  -1.5906   2.0838  -1.5326   0.8694  -0.3192
                                        % 0.0821  -0.0122   0.0009

                                        % Compute frquency response
[H,w]=freqz(b,a,256);
                                        % gain factor; Butterworth Hmax is at the origin
b0=1/abs(H(1))                          % b0 = 0.0023
H=b0*H;
                                        % plot the frequency response
plot(f_sam*w/(2*pi), 20*log10(abs(H)),'-r')
axis([0 5000 -50 10]);
xlabel('frequency, Hz')
ylabel('|H|')
title('Digital LP Filter Based on Butterworth Design'), pause
plot(f_sam*w/(2*pi), angle(H)/pi,'-r')
axis([0 5000 -1 1]);
xlabel('frequency, Hz')
ylabel('angle(H)/pi  (rad)')
```



B. Using Matlab as a CAD Tool

Remember, Matlab uses frequencies normalized by f_sam/2 (that is, 2f/f_sam)

```
% Script File: IIR_Analog_2.m
                                        % Digital Filter Specificatons
f1= 2E3;                                % passband edge in Hz
f2= 3E3;                                % stopband edge in Hz
f_sam=10E3;                             % sampling frequency in Hz
del_1 =1-1/sqrt(2);                     % peak ripple in passband
del_2=0.01;                             % peak ripple in the stopband

                                        % Estimate required filter order
A1=-20*log10(del_1);
A2=-20*log10(del_2);
N=buttord(w1/pi, w2/pi, A1, A2)         % N = 8

                                        % Determine Filter Coefficients
[b,a]= butter(N,w1/pi);
                                        % Plot Frequency Resposnse
[H,w]=freqz(b,a,128);
plot(f_sam*w/(2*pi), 20*log10(abs(H)),'-')
axis([0 5000 -50 10]);
xlabel('frequency, Hz')
ylabel('|H|')
title('Design Using Matlab as a CAD Tool')
axis;
```

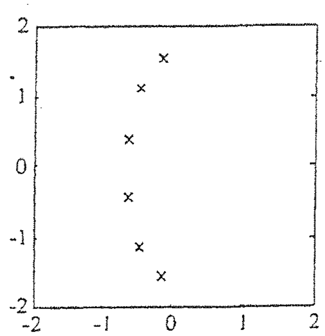


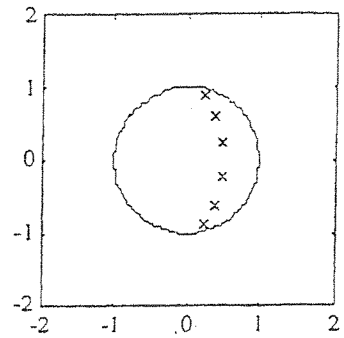
II. Design Based on Chebyshev I Analog Filter

A. Using Matlab as a Calculator

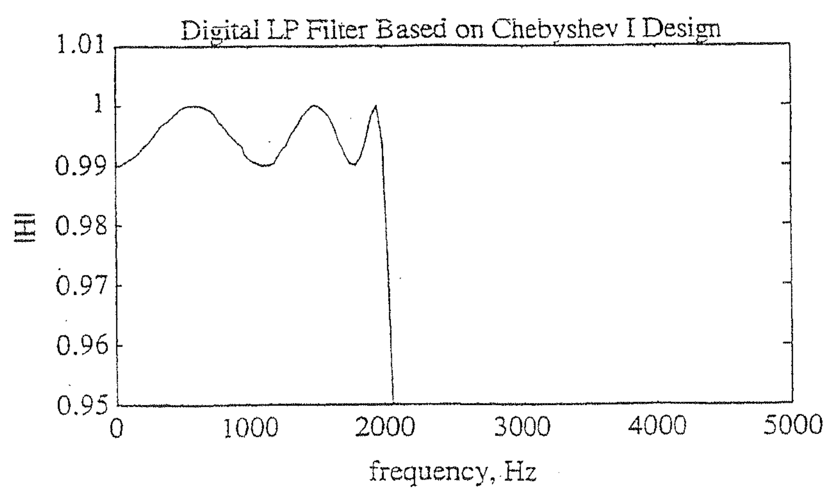
```
% Script File: IIR_Analog_3.m
%
j=sqrt(-1);
                                          % Digital Filter Specifications
f1=2E3;                                   % passband edge Hz
f2=3E3;                                   % stopband edge Hz
f_sam=10E3;                               % sanpling frequency Hz
del_1=0.01;                               % passband ripple
del_2=0.01;                               % stopband ripple

                                          % corresponding critical digital frequencies
w1=2*pi*f1/f_sam;
w2=2*pi*f2/f_sam;
                                          % Prewarp the w's
W1=2*tan(w1/2);
W2=2*tan(w2/2);
                                          % Determine required filter parameters
epsilon=sqrt(1/(1-del_1)^2 -1)            % see the lecture notes: epsilon = 0.1425

A1=-20*log10(1-del_1);                    % calculate the filter order
A2=-20*log10(del_2);
R=(10^(A2/10)-1)/(10^(A1/10)-1);          Note √R here is R in class notes
N=ceil(acosh(sqrt(R))/acosh(W2/W1))       % see the lecture notes; N = 6
                                          % N smaller than Butterworth

                                          % Check
N= ceil(acosh(1/(epsilon*del_2))/acosh(W2/W1))   % see the lecture notes; N = 8

                                          % poles and zeros of H(s)
k=[0:1:N-1];
theta=pi/2 + pi/(2*N) + k*pi/N;
beta=((sqrt(1+epsilon^2)+1)/epsilon)^(1/N);   % see the lecture notes
r1=W1*(beta^2+1)/(2*beta);
r2=W1*(beta^2-1)/(2*beta);
v=r2*cos(theta) + j*r1*sin(theta);
v=v'
                                          % v =
                                          % -0.1713 - 1.5424i
                                          % -0.4681 - 1.1291i
                                          % -0.6394 - 0.4133i
                                          % -0.6394 + 0.4133i
                                          % -0.4681 + 1.1291i
                                          % -0.1713 + 1.5424i

plot(real(v),imag(v),'x')                 % plot poles in the s-plane
axis([-2 2 -2 2]), axis('square')
pause
                                          % plot poles and zeros of H(z) obtained
                                          % using the bilinear transformation
c=-ones(size(k))';
```

```
d=(1+v/2)./(1-v/2);
tmp=exp(j*[0:pi/90:2*pi]);
plot(real(d),imag(d),'x',real(tmp),imag(tmp),'-'),  axis('square'),  pause
```



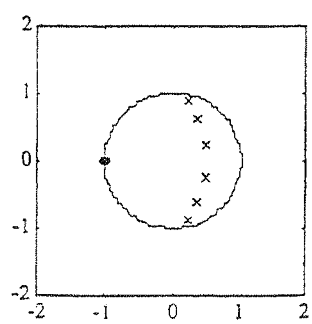


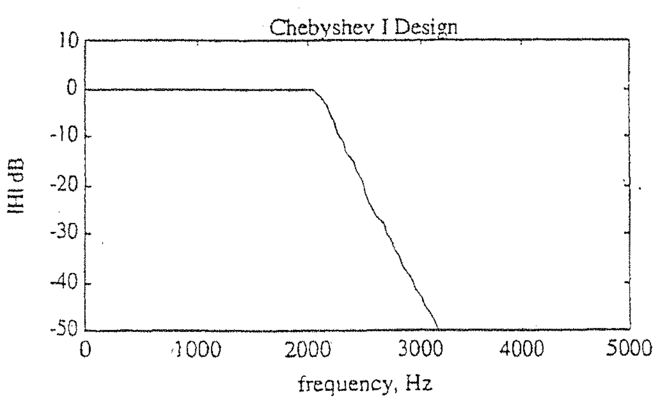

```
b=poly(c)                          % Find the Digital Filter Coefficients
a=poly(d);
                                   % b = [1  6  15  20  15  6  1]
                                   % a =  [1.0000  -2.0877   2.9694
                                   %        -2.6267  1.5887  -0.5982  0.1124]

b0 = real(prod(1-d)/prod(1-c));    % gain factor
if (rem(N,2) == 0), b0=b0/sqrt(1+epsilon^2), end
%                                  % Gain factor is determined by |H| at w =0.
                                   % b0 = 0.0055

                                   % Implement using first order and
                                   % biquad sections
% Skipped here; similar examples done in lecture and homework

[H,w]=freqz(b,a,128);              % plot the frquency response
H=b0*H;
plot(f_sam*w/(2*pi), abs(H),'-')
axis([0 5000 0.95 1.01]);
xlabel('frequency, Hz')
ylabel('|H|')
title('Digital LP Filter Based on Chebyshev I Design')
axis; pause
```

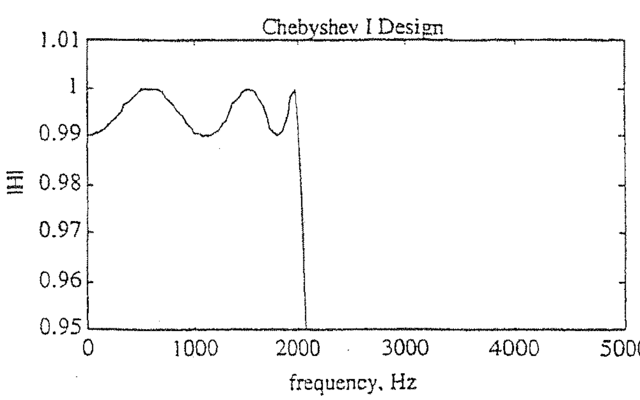


B. Using Matlab as a CAD Tool

```
% Script File: IIR_Analog_4.m

                                        % Estimate Required Order
[N,wn]=cheb1ord(w1/pi,w2/pi,A1,A2);     % Matlab normalizes all frequencies to f_sam /2
                                        % N = 6
[b,a]=cheby1(N,A1,wn);                  % Determine filter coefficients

c=roots(b);                             % Plot poles and zeros
d=roots(a);
tmp=exp(i*[0:pi/90:2*pi]);
axis('square');
axis([-2 2 -2 2]);
plot(real(c),imag(c),'o',real(d),imag(d),'x',real(tmp),imag(tmp),'-')
pause


[H,w]=freqz(b,a,128);                   %  Plot Frequency Resposnse
plot(f_sam*w/(2*pi), 20*log10(abs(H)),'-')   % dB plot
axis([0 5000 -50 10]);
xlabel('frequency, Hz')
ylabel('|H| dB')
title('Chebyshev I Design')
pause
%
plot(f_sam*w/(2*pi), abs(H),'-')        % Linear Scale Plot
axis([0 5000 .95 1.01])
xlabel('frequency, Hz')
ylabel('|H|')
title('Chebyshev I Design'), pause
```







III. Chebyshev II Design Using Matlab as a CAD Tool

```
% Script File: IIR_Analog_5.m

[N,wn]=cheb2ord(w1/pi,w2/pi,A1,A2);          % Estimate Required Order
                                             % N = 6 (same as Cheby1)
[b,a]=cheby2(N,A2,wn);                       % Determine filter coefficients

c=roots(b);                                  % Plot poles and zeros
d=roots(a);
plot(real(c),imag(c),'ob',real(d),imag(d),'xr',real(tmp),imag(tmp),'-r')
axis('square');
axis([-2 2 -2 2]);
pause

[H,w]=freqz(b,a,128);                        % Plot Frequency Resposnse
plot(f_sam*w/(2*pi), 20*log10(abs(H)),'-')   % dB plot
axis([0 5000 -50 10]);
xlabel('frequency, Hz')
ylabel('|H| dB')
title('Chebyshev II Design Using Matlab as a CAD Tool')
pause

plot(f_sam*w/(2*pi), abs(H),'-')             % Linear Scale Plot
axis([0 5000 .95 1.01])
xlabel('frequency, Hz')
ylabel('|H|')
title('Chebyshev II Design')
axis;
```