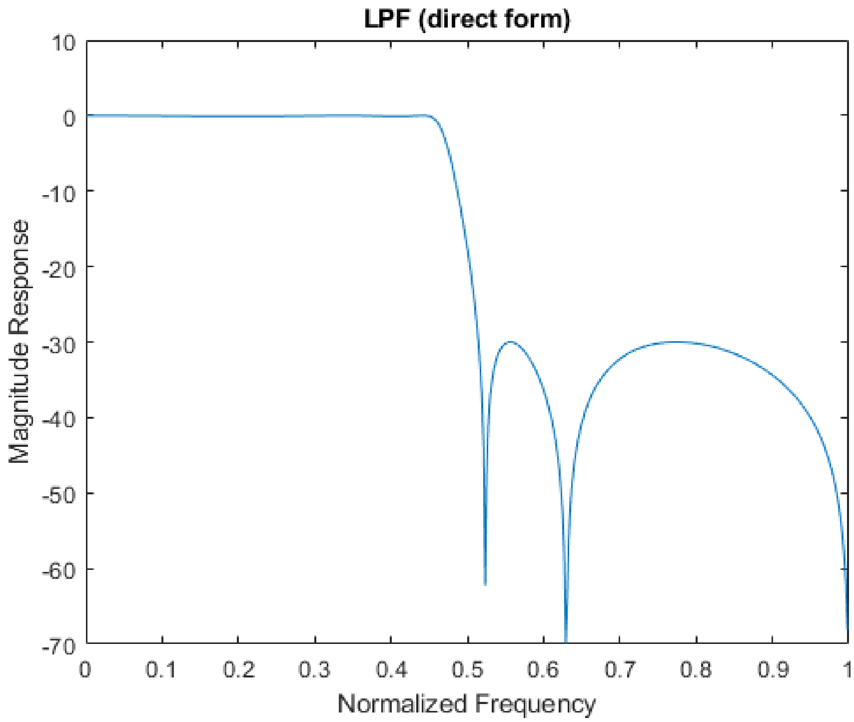
Homework 3
By: Zachary Sweep
2/18/18

Part 1

Design a lowpass elliptic filter with the following specifications: ωp=0.45π, ωs=0.55π and Rp=0.1dB, Rs=30dB. List the coefficients for the resulting transfer function in direct form. Plot the magnitude response and group delay for the filter design.

```
wp = 0.45*pi;
ws = 0.55*pi;
Rp = 0.1; %dB
Rs = 30;  %dB

[N,wpnew] = ellipord(wp/pi,ws/pi,Rp,Rs);
[z,p,k]   = ellip(N,Rp,Rs,wpnew);

% Calculate Z domain numerator polynomial
Bz = poly(z);
Bz = k*Bz;
disp('Numerator Coefficients:\n');
disp(Bz);

% Calculate Z domain denominator polynomial
Az = poly(p);
disp('Denominator Coefficients:\n');
disp(Az);

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

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

% Plotting Magnitude Response and Group Delay
figure(1)
plot(wh/pi,20*log10(abs(H_orig)),'-');
title('LPF (direct form)');
xlabel('Normalized Frequency'); ylabel('Magnitude Response');

figure(2)
plot(wg/pi,Gpd,'-');
title('LPF (direct form)');
xlabel('Normalized Frequency'); ylabel('Group Delay (s)');
```

```
Numerator Coefficients:\n
    0.1234    0.2395    0.3775    0.3775    0.2395    0.1234


Denominator Coefficients:\n
    1.0000   -0.7051    1.4503   -0.6273    0.4411   -0.0781
```


LPF (direct form)
Magnitude Response
Normalized Frequency
10
0
-10
-20
-30
-40
-50
-60
-70
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

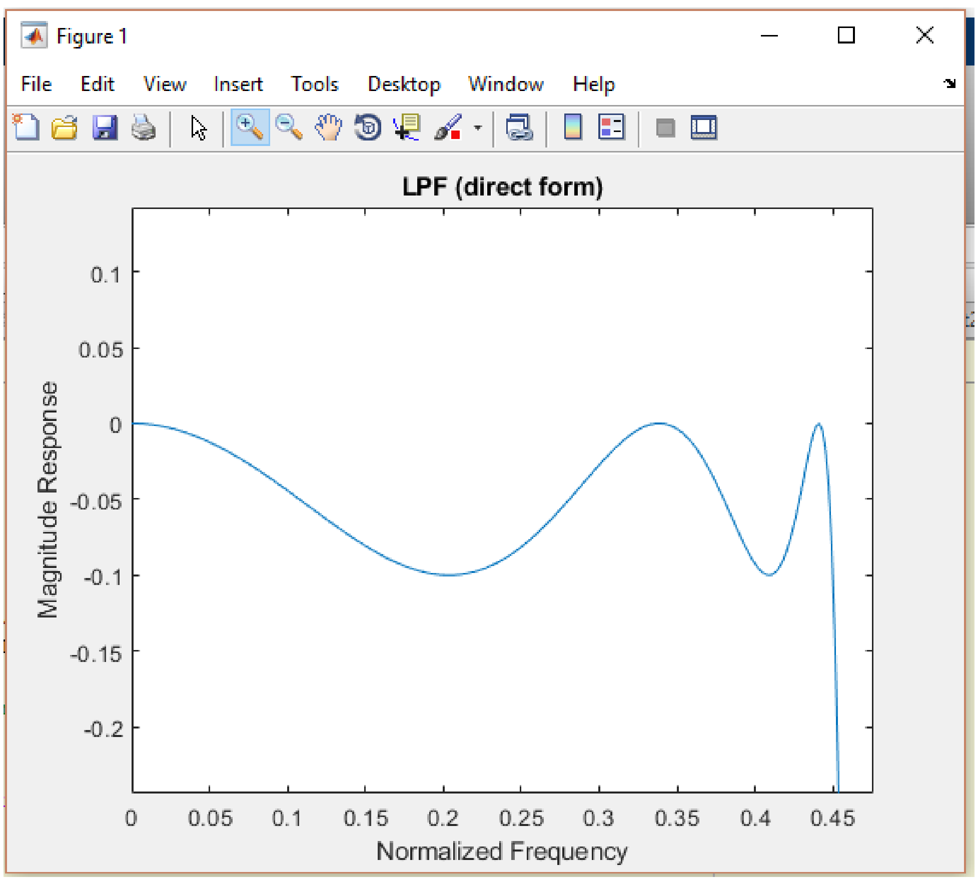
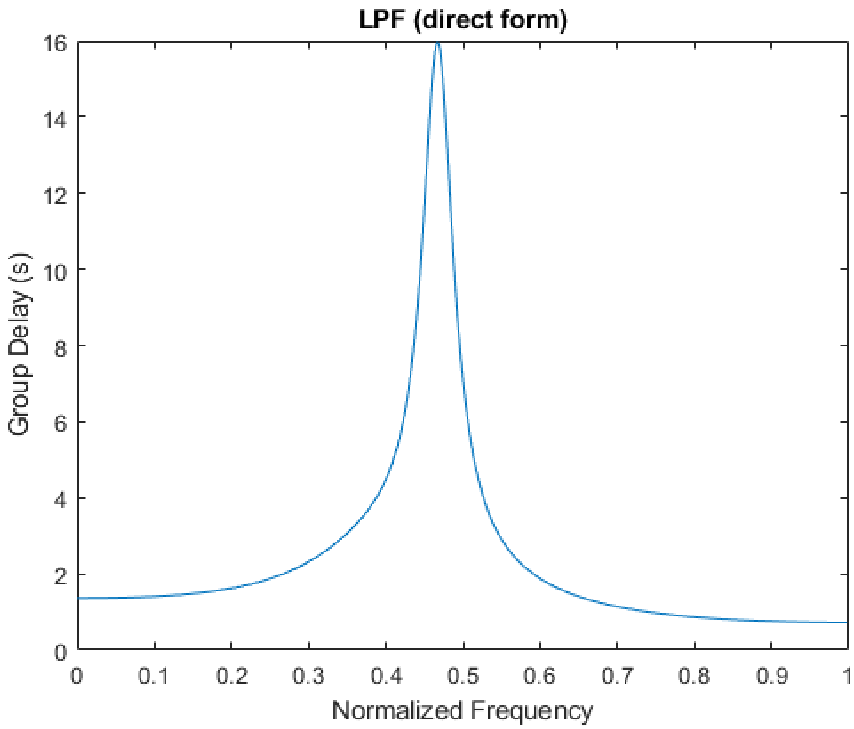


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
LPF (direct form)
Magnitude Response
Normalized Frequency
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45



Published with MATLAB® R2017b

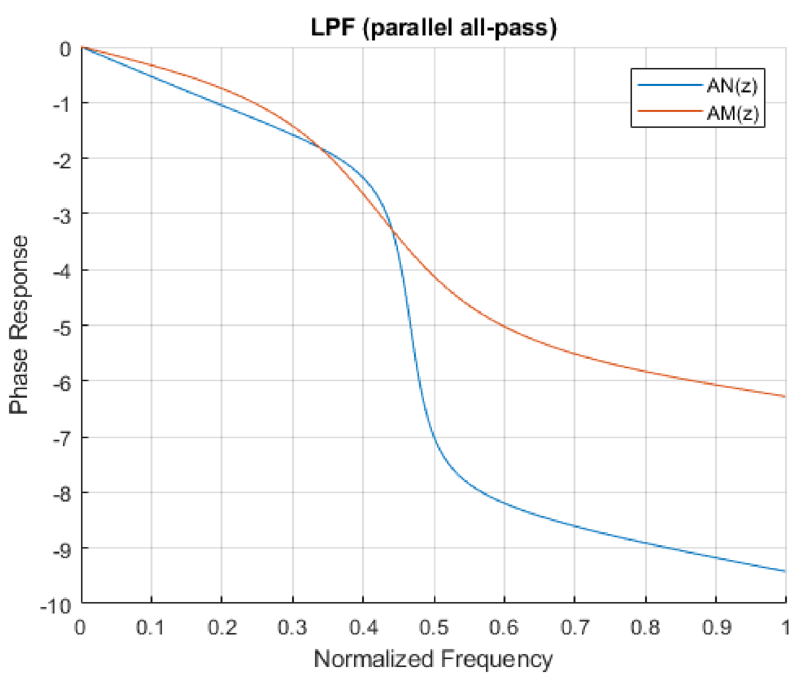
Part 2

Realize the lowpass filter as a parallel combination of allpass sections. List the coefficients and plot the phase response for each allpass section.

```
%.......................... REPEAT OF PART 1
wp = 0.45*pi;
ws = 0.55*pi;
Rp = 0.1; %dB
Rs = 30;  %dB

[N,wpnew] = ellipord(wp/pi,ws/pi,Rp,Rs);
[z,p,k]   = ellip(N,Rp,Rs,wpnew);

%........................... START OF PART 2
% 2nd Order All-Pass Filter
AMaz = poly([p(3) p(4)]);
AMbz = flip(AMaz);

% 3nd Order All-Pass Filter
ANaz = poly([p(1) p(2) p(5)]);
ANbz = flip(ANaz);

disp('2nd Order All-Pass Numerator:');
```

```
disp(AMbz);
disp('2nd Order All-Pass Denominator:');
disp(AMaz);

disp('3rd Order All-Pass Numerator:');
disp(ANbz);
disp('3rd Order All-Pass Denominator:');
disp(ANaz);

figure(3)
hold on
phasez(ANbz,ANaz);
phasez(AMbz,AMaz);
title('LPF (parallel all-pass)');
xlabel('Normalized Frequency'); ylabel('Phase Response');
legend('AN(z)','AM(z)');
hold off
```

```
2nd Order All-Pass Numerator:
    0.4290   -0.3033    1.0000

2nd Order All-Pass Denominator:
    1.0000   -0.3033    0.4290

3rd Order All-Pass Numerator:
   -0.1822    0.8994   -0.4018    1.0000

3rd Order All-Pass Denominator:
    1.0000   -0.4018    0.8994   -0.1822
```

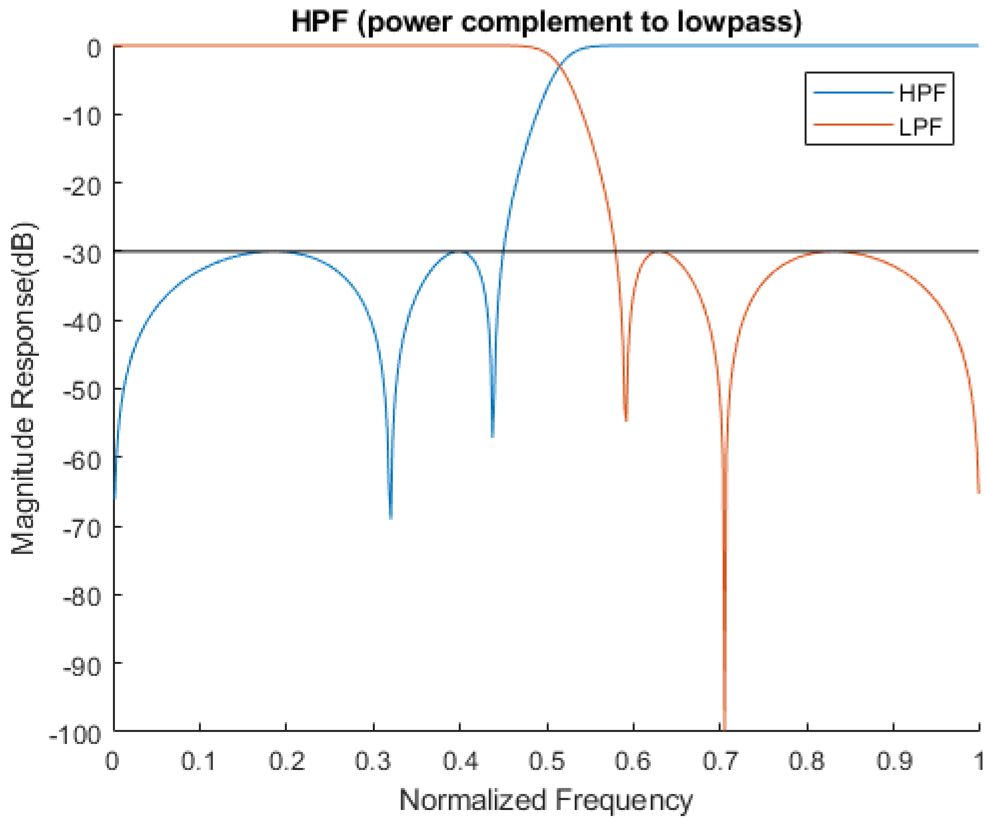


Part 3

Find the magnitude response for the highpass power complement to the lowpass filter. Adjust the passband ripple specification (Rp) for the lowpass filter to achieve equal stopband attenuation for both lowpass and highpass filters (within ±1dB). Plot the magnitude response for both filters.

**I got the stopband to be within +-1dB of the LPF stopband by keeping the filter order at 5 and changing Rp to 0.00435;

```
wp = 0.45*pi;
ws = 0.55*pi;

% new Rp
Rp = 0.00435; %dB
Rs = 30;      %dB

% Recompute order and w
[N,wpnew] = ellipord(wp/pi,ws/pi,Rp,Rs);

% fix filter order to 5th order
[z,p,k]   = ellip(5,Rp,Rs,wpnew);


% Recompute allpass filters
AMaz = poly([p(3) p(4)]);
AMbz = flip(AMaz);
ANaz = poly([p(1) p(2) p(5)]);
ANbz = flip(ANaz);
[pAM,wm]  = phasez(AMbz,AMaz);
[pAN,wn]  = phasez(ANbz,ANaz);

% High pass filter response
Hhp = (exp(1j*pAM)-exp(1j*pAN))/2;

% Low pass filter response
Hlp = (exp(1j*pAM)+exp(1j*pAN))/2;

figure(5)
hold on
plot(wn/pi,20*log10(abs(Hhp)));
plot(wn/pi,20*log10(abs(Hlp)));
line([0 wn(end)/pi],[-30 -30],'color','black');
title('HPF (power complement to lowpass)');
xlabel('Normalized Frequency'); ylabel('Magnitude Response(dB)');
legend('HPF','LPF');
```



Published with MATLAB® R2017b

Part 4

Determine the lattice realization for each allpass section. List the multiplier coefficients for each stage of the lattice for both allpass sections (use the MATLAB function tf2latc).

```
%........................... REPEAT OF PART 1
wp = 0.45*pi;
ws = 0.55*pi;
Rp = 0.1; %dB
Rs = 30;  %dB

[N,wpnew] = ellipord(wp/pi,ws/pi,Rp,Rs);
[z,p,k]   = ellip(N,Rp,Rs,wpnew);

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

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

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

%......................... REPEAT OF PART 2
% 2nd Order All-Pass Filter
AMaz = poly([p(3) p(4)]);
AMbz = flip(AMaz);

% 3nd Order All-Pass Filter
ANaz = poly([p(1) p(2) p(5)]);
ANbz = flip(ANaz);

%......................... START OF PART 4
K1 = tf2latc(ANaz);
K2 = tf2latc(AMaz);

disp('All-Pass filter 1 multiplier coefficients (3rd Order):')
disp(K1)

disp('All-Pass filter 2 multiplier coefficients (2nd Order):')
disp(K2)
```

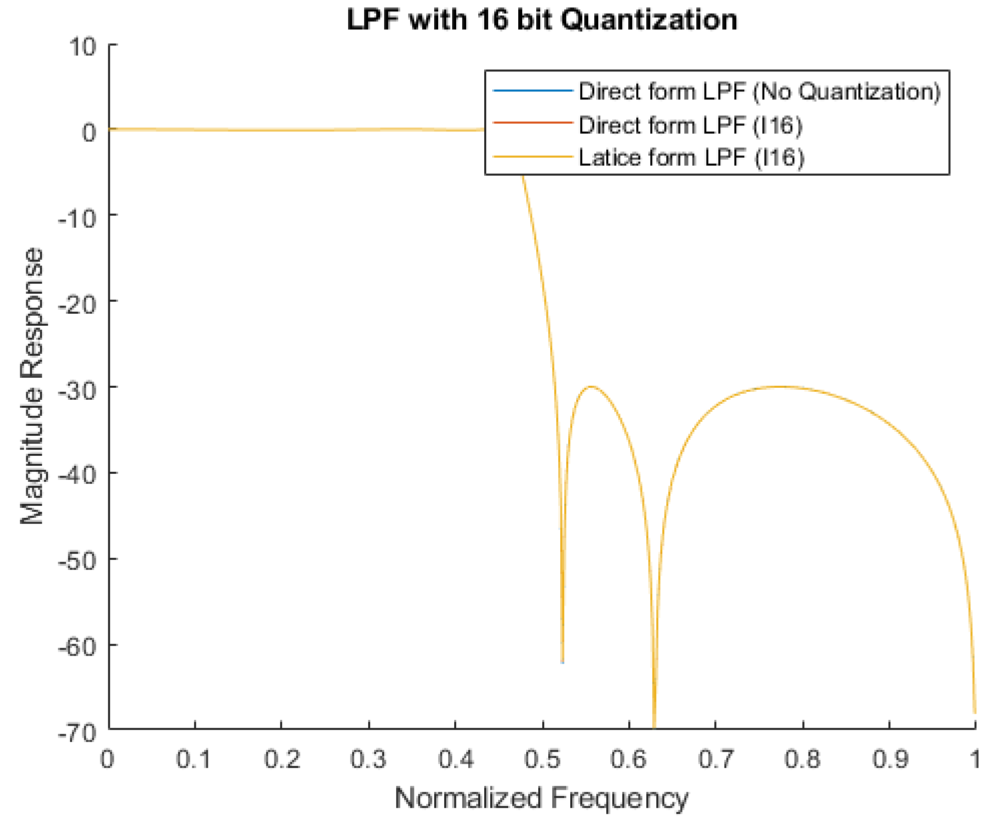
```
All-Pass filter 1 multiplier coefficients (3rd Order):
   -0.1327
    0.8546
   -0.1822

All-Pass filter 2 multiplier coefficients (2nd Order):
   -0.2122
    0.4290
```

Part 5

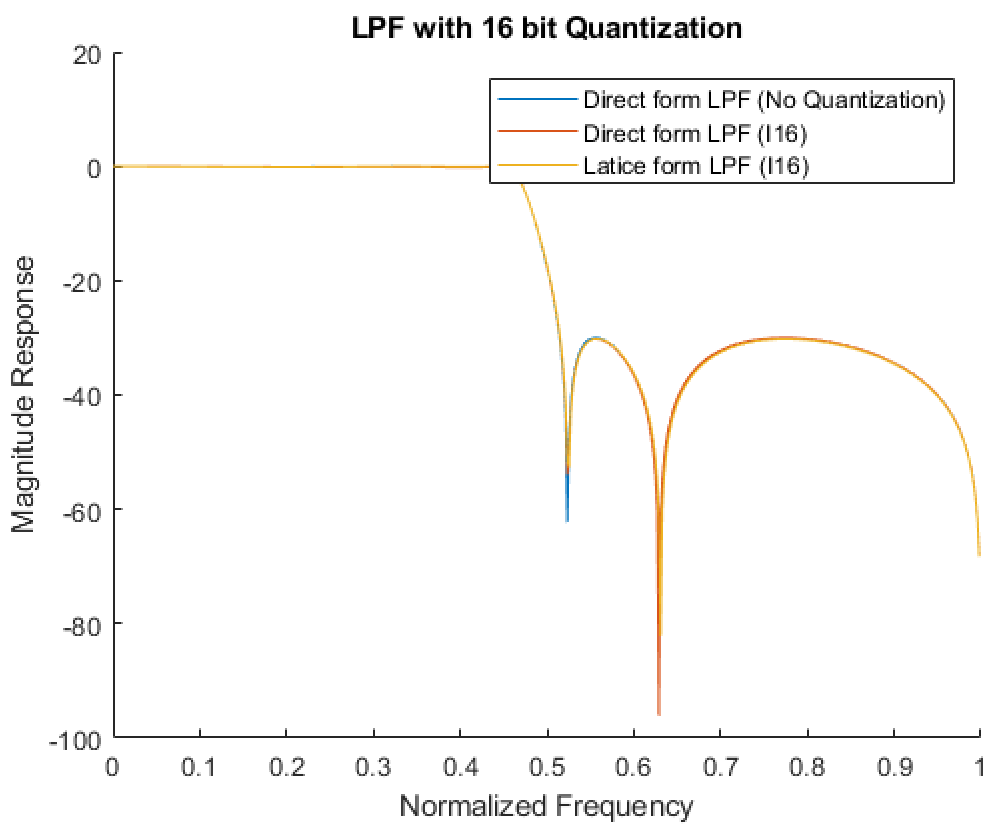
Quantize the filter coefficients for the lowpass filter (direct form) designed in step 1 to 16 bits. Quantize the lattice coefficients to 16 bits. Determine the frequency response for both direct form and lattice realizations with quantized coefficients (use the MATLAB function latc2tf to convert the lattice coefficients back to direct form). Compare both to the direct form frequency response without coefficient quantization. Repeat step 5 with 10 bit coefficient quantization. To ensure uniformity of results, please use the supplied MATLAB function quantize.m.

```
%..........16 bit Quantization

% Quantize direct form LPF coefficients and Compute
% Frequency response of quantized direct form LPF
[H_dir,wh]   = freqz(quantize(Bz,16),quantize(Az,16));
```

```
% Quantize Latice form LPF coefficients and
% Compute Latice realization transfer function for both all-pass sections
[num1,den1]=latc2tf(quantize(K1,16),'allpass');
[num2,den2]=latc2tf(quantize(K2,16),'allpass');

% Compute phase response for each all-pass section
[phAN,wn] = phasez(num1,den1);
[phAM,wm] = phasez(num2,den2);

% Compute frequency response of parallel all-pass sections
H_lat = (exp(1j*phAN)+exp(1j*phAM))/2;

figure(6)
hold on
plot(wh/pi,20*log10(abs(H_orig)));
plot(wh/pi,20*log10(abs(H_dir)));
plot(wn/pi,20*log10(abs(H_lat)));
title('LPF with 16 bit Quantization');
xlabel('Normalized Frequency'); ylabel('Magnitude Response');
legend('Direct form LPF (No Quantization)','Direct form LPF (I16)','Latice form LPF (I16)')


%..........10 bit Quantization

% Quantize direct form LPF coefficients and Compute
% Frequency response of quantized direct form LPF
[H_dir,wh]   = freqz(quantize(Bz,10),quantize(Az,10));

% Quantize Latice form LPF coefficients and
% Compute Latice realization transfer function for both all-pass sections
[num1,den1]=latc2tf(quantize(K1,10),'allpass');
[num2,den2]=latc2tf(quantize(K2,10),'allpass');

% Compute phase response for each all-pass section
[phAN,wn] = phasez(num1,den1);
[phAM,wm] = phasez(num2,den2);

% Compute frequency response of parallel all-pass sections
H_lat = (exp(1j*phAN)+exp(1j*phAM))/2;

figure(7)
hold on
plot(wh/pi,20*log10(abs(H_orig)));
plot(wh/pi,20*log10(abs(H_dir)));
plot(wn/pi,20*log10(abs(H_lat)));
title('LPF with 16 bit Quantization');
xlabel('Normalized Frequency'); ylabel('Magnitude Response');
legend('Direct form LPF (No Quantization)','Direct form LPF (I16)','Latice form LPF (I16)')
```



LPF with 16 bit Quantization
Direct form LPF (No Quantization)
Direct form LPF (I16)
Latice form LPF (I16)
Magnitude Response
Normalized Frequency
10
0
-10
-20
-30
-40
-50
-60
-70
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



LPF with 16 bit Quantization
Direct form LPF (No Quantization)
Direct form LPF (I16)
Latice form LPF (I16)
Magnitude Response
Normalized Frequency
20
0
-20
-40
-60
-80
-100
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Published with MATLAB® R2017b