MODULE 07 HOMEWORK

3/25/18

EN.525.718.81.SP18 MULTIRATE SIGNAL PROCESSING

SPRING 2018

By: Zachary H. Sweep

Contents

Problem 1

Design a 2-stage downsampler for M=16 that minimizes the overall computation rate. Use the Parks-McClellan design algorithm for each filter stage. The requirements for the anti-aliasing filter are as follows: ω_p=7/8 π/M, ω_s=π/M, δ_1=0.01, δ_2=0.001. Show clearly that your design meets all specifications by plotting the specification template on the frequency response graph. Compare the computation rate for the single-stage implementation with your design.

The first step to designing the most efficient 2-stage downsampler is finding the stretch factor M1 for stage 1 and M2 for stage 2 (where M1xM2 = M) such that the computational cost is minimized. With the given design parameters, the stretch factors that will lead to the most efficient downsampler are M1 = 8 and M2 = 2. Using these stretch factors, the 2-stage downsampler can be designed with a stage 1 filter order of 46 and a stage 2 filter order 90 for a total order of 136. If the above filter was designed in direct form it would require an order of approximately 651 (almost 5 times larger than the 2-stage implementation).

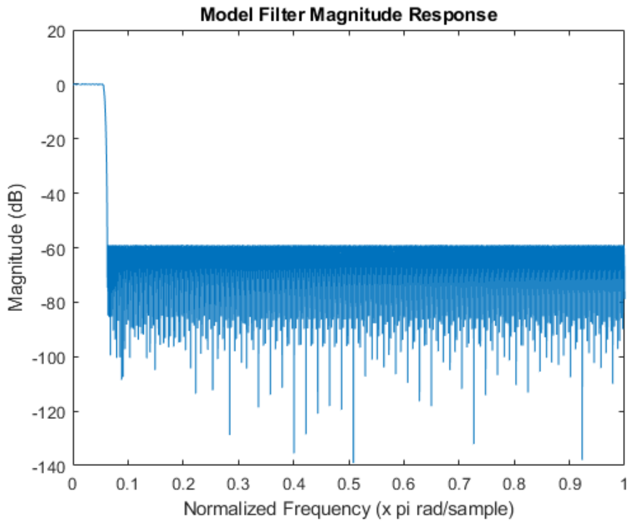
```
% Decimation factor
M = 16;

% Passband/stopband requirements
wp = (7/8)*(pi/M);
ws = pi/M;

% Peak Ripple parameters
d1 = 0.01;
d2 = 0.001;

% Model filter (Single Stage)
[n,f0,m0,w] = firpmord([wp/pi ws/pi],[1 0],[d1 d2]);
N_single_stage = n;
b = firpm(n,f0,m0,w);

% Plot model filter magnitude response (single stage)
figure(1)
[H,w] = freqz(b,1,10000);
plot(w/pi,20*log10(abs(H)));
title('Model Filter Magnitude Response');xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');

% COMPUTE MULTISTAGE DECIMATOR FILTER PARAMETERS
% Find stretch factors that minimize total computational cost
% Compute factors of M (not including 1 and M)
M1 = 2:M-1;
M1 = M1(~(rem(M, M1)));
M2 = flip(M1);

D = (-10*log10(d1/2*d2) - 13)/2.324;
cost = zeros(1,length(M1));
for i=1:length(M1)
    Ng = D/(M1(i)^2*M2(i)*(ws-wp));
    Ni = D/((2*pi)-M1(i)*(ws+wp));
    cost(i) = Ng+Ni;
end
% Find min cost and corresponding index. Use index to find M1 and M2
% stretch factors
[min_cost,i] = min(cost);
M1 = M1(i);
M2 = M2(i);
fprintf('\nStretch Factors:\nM1 = %d, M2 = %d\n',M1,M2);
```

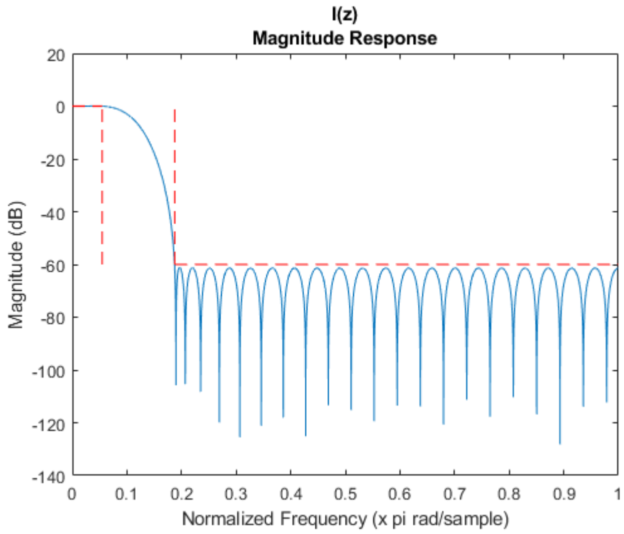
```
%************ STAGE 1 I(z) ************************************************
% Stage 1 passband/stop band parameters
wp_i = wp;
ws_i = (2*pi/M1)-ws;
wc = (wp_i+ws_i)/2;

% Estimate the order for stage 1 lowpass filter. Note that filter I(z) uses
% d1/2 for the passband peak ripple.
[n,f0,m0,w] = firpmord([wp_i/pi ws_i/pi],[1 0],[d1/2 d2]);

% Increase filter order since firpmord() underestimated the required order
n = n+5;
N_stage_1 = n;

% Compute stage 1 impulse response
b_i = firpm(n,f0,m0,w);

% Compute frequency response I(z)
[H,w] = freqz(b_i,1,10000);

% Plot magnitude response for I(z)
figure(2)
plot(w/pi,20*log10(abs(H)));
title({'I(z)','Magnitude Response'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp_i/pi],[20*log10(1+d1/2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([0 wp_i/pi],[20*log10(1-d1/2) 20*log10(1-d1/2)],'color','red','LineStyle','--');
line([ws_i/pi 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp_i/pi wp_i/pi],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([ws_i/pi ws_i/pi],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');

% Plot zoomed view of passband
figure(3)
plot(w/pi,20*log10(abs(H)));
title({'I(z)','Zoomed View of Passband'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp_i/pi],[20*log10(1+d1/2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([0 wp_i/pi],[20*log10(1-d1/2) 20*log10(1-d1/2)],'color','red','LineStyle','--');
line([ws_i/pi 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp_i/pi wp_i/pi],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([ws_i/pi ws_i/pi],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
axis([0 0.06 -0.065 0.05]);

% Plot zoomed view of stopband
figure(4)
plot(w/pi,20*log10(abs(H)));
title({'I(z)','Zoomed View of Stopband'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp_i/pi],[20*log10(1+d1/2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([0 wp_i/pi],[20*log10(1-d1/2) 20*log10(1-d1/2)],'color','red','LineStyle','--');
line([ws_i/pi 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp_i/pi wp_i/pi],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([ws_i/pi ws_i/pi],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
axis([0.184 0.203 -65 -54]);

%**************************************************************************
%************ STAGE 2 G(z) ************************************************
```

```
% Stage 2 passband/stop band parameters
wp_g = M1*wp;
ws_g = M1*ws;

% Estimate the order for stage 2 lowpass filter. Note that filter G(z) uses
% d1/2 for the passband peak ripple.
[n,f0,m0,w] = firpmord([wp_g/pi ws_g/pi],[1 0],[d1/2 d2]);

% Increase filter order since firpmord() underestimated the required order
n = n+2;
N_stage_2 = n;

% Compute stage 2 impulse response
b_g = firpm(n,f0,m0,w);

% Upsample G(z) impulse response
b_g_interp = upsample(b_g,M1);

% Compute frequency response G(z)
[H,w] = freqz(b_g_interp,1,10000);

% Plot magnitude response of stage 2 filter
figure(5)
plot(w/pi,20*log10(abs(H)));
title({'G(z^8)','Magnitude Response'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp_g/(pi*M1)],[20*log10(1+d1/2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([0 wp_g/(pi*M1)],[20*log10(1-d1/2) 20*log10(1-d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp_g/(pi*M1) wp_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) ws_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');

% Plot zoomed view of passband
figure(6)
plot(w/pi,20*log10(abs(H)));
title({'G(z^8)','Zoomed View of Passband'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp_g/(pi*M1)],[20*log10(1+d1/2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([0 wp_g/(pi*M1)],[20*log10(1-d1/2) 20*log10(1-d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp_g/(pi*M1) wp_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) ws_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
axis([0 0.06 -0.05 0.05])

% Plot zoomed view of passband @ wp_g
figure(7)
plot(w/pi,20*log10(abs(H)));
title({'G(z^8)','Zoomed View of Passband @ wp'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp_g/(pi*M1)],[20*log10(1+d1/2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([0 wp_g/(pi*M1)],[20*log10(1-d1/2) 20*log10(1-d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp_g/(pi*M1) wp_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) ws_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
axis([0.0528 0.0548 -0.05 0.045])

% Plot zoomed view of stopband
figure(8)
plot(w/pi,20*log10(abs(H)));
```

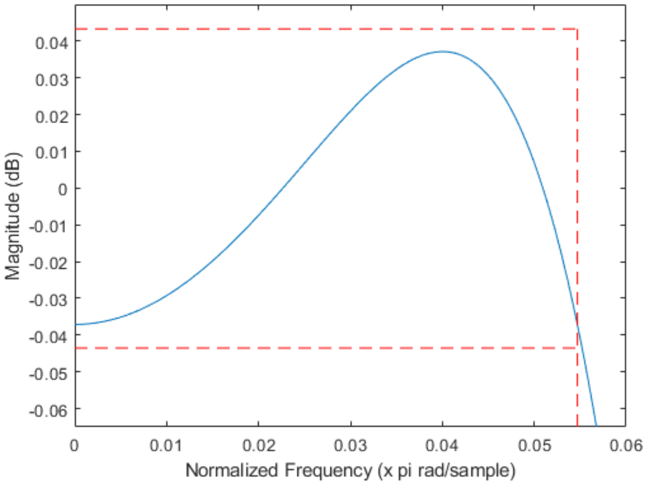
```
title({'G(z^8)','Zoomed View of Stopband'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp_g/(pi*M1)],[20*log10(1+d1/2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([0 wp_g/(pi*M1)],[20*log10(1-d1/2) 20*log10(1-d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp_g/(pi*M1) wp_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) ws_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
axis([0 1 -66 -53]);

% Plot zoomed view of stopband at ws
figure(9)
plot(w/pi,20*log10(abs(H)));
title({'G(z^8)','Zoomed View of Stopband @ws'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp_g/(pi*M1)],[20*log10(1+d1/2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([0 wp_g/(pi*M1)],[20*log10(1-d1/2) 20*log10(1-d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp_g/(pi*M1) wp_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
line([ws_g/(pi*M1) ws_g/(pi*M1)],[20*log10(d2) 20*log10(1+d1/2)],'color','red','LineStyle','--');
axis([0.06244 0.06255 -60.0007 -59.9988])


%************ Convolve Filters *********************************************
% Convolve I(z) and G(z) impulse responses
h = conv(b_i,b_g_interp);

% Compute frequency response of filter
[H,w] = freqz(h,1,10000);

% Plot magnitude response of filter
figure(10)
plot(w/pi,20*log10(abs(H)));
title({'I(z)G(z^8)','Magnitude Response'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp/pi],[20*log10(1+d1) 20*log10(1+d1)],'color','red','LineStyle','--');
line([0 wp/pi],[20*log10(1-d1) 20*log10(1-d1)],'color','red','LineStyle','--');
line([ws/pi 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp/pi wp/pi],[20*log10(d2) 20*log10(1+d1)],'color','red','LineStyle','--');
line([ws/pi ws/pi],[20*log10(d2) 0],'color','red','LineStyle','--');

% Plot zoomed view of passband
figure(11)
plot(w/pi,20*log10(abs(H)));
title({'I(z)G(z^8)','Zoomed View of Passband'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp/pi],[20*log10(1+d1) 20*log10(1+d1)],'color','red','LineStyle','--');
line([0 wp/pi],[20*log10(1-d1) 20*log10(1-d1)],'color','red','LineStyle','--');
line([ws/pi 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp/pi wp/pi],[20*log10(d2) 20*log10(1+d1)],'color','red','LineStyle','--');
line([ws/pi ws/pi],[20*log10(d2) 0],'color','red','LineStyle','--');
axis([0 0.06 -0.2 0.2]);

% Plot zoomed view of passband at wp
figure(12)
plot(w/pi,20*log10(abs(H)));
title({'I(z)G(z^8)','Zoomed View of Passband @ wp'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp/pi],[20*log10(1+d1) 20*log10(1+d1)],'color','red','LineStyle','--');
line([0 wp/pi],[20*log10(1-d1) 20*log10(1-d1)],'color','red','LineStyle','--');
line([ws/pi 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp/pi wp/pi],[20*log10(d2) 20*log10(1+d1)],'color','red','LineStyle','--');
line([ws/pi ws/pi],[20*log10(d2) 0],'color','red','LineStyle','--');
axis([0.05462 0.05479 -0.2 0.1]);

% Plot zoomed view of stopband
figure(13)
plot(w/pi,20*log10(abs(H)));
title({'I(z)G(z^8)','Zoomed View of Stopband'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp/pi],[20*log10(1+d1) 20*log10(1+d1)],'color','red','LineStyle','--');
line([0 wp/pi],[20*log10(1-d1) 20*log10(1-d1)],'color','red','LineStyle','--');
line([ws/pi 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp/pi wp/pi],[20*log10(d2) 20*log10(1+d1)],'color','red','LineStyle','--');
line([ws/pi ws/pi],[20*log10(d2) 0],'color','red','LineStyle','--');
axis([0 1 -100 -20]);

% Plot zoomed view of stopband @ ws
figure(14)
plot(w/pi,20*log10(abs(H)));
title({'I(z)G(z^8)','Zoomed View of Stopband @ ws'});xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
line([0 wp/pi],[20*log10(1+d1) 20*log10(1+d1)],'color','red','LineStyle','--');
line([0 wp/pi],[20*log10(1-d1) 20*log10(1-d1)],'color','red','LineStyle','--');
line([ws/pi 1],[20*log10(d2) 20*log10(d2)],'color','red','LineStyle','--');
line([wp/pi wp/pi],[20*log10(d2) 20*log10(1+d1)],'color','red','LineStyle','--');
line([ws/pi ws/pi],[20*log10(d2) 0],'color','red','LineStyle','--');
axis([0.06242 0.06259 -60.0015 -59.9980]);

fprintf('\nDirect-Form Filter Order = %d',N_single_stage);
fprintf('\nStage 1 Filter Order = %d',N_stage_1);
fprintf('\nStage 2 Filter Order = %d\n',N_stage_2);
```

```
Stretch Factors:
M1 = 8, M2 = 2

Direct-Form Filter Order = 651
Stage 1 Filter Order = 46
Stage 2 Filter Order = 90
```

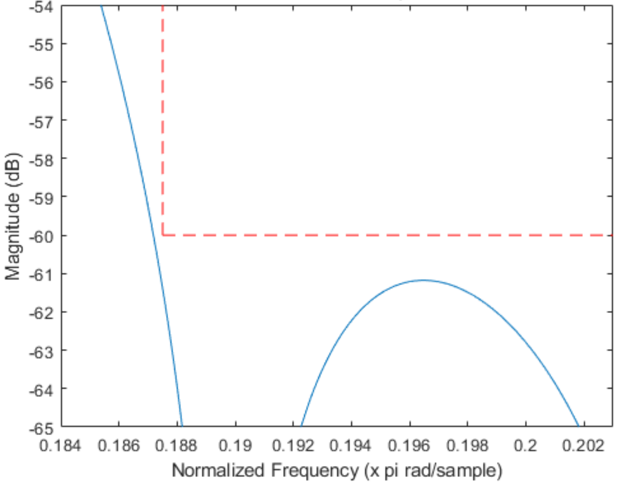




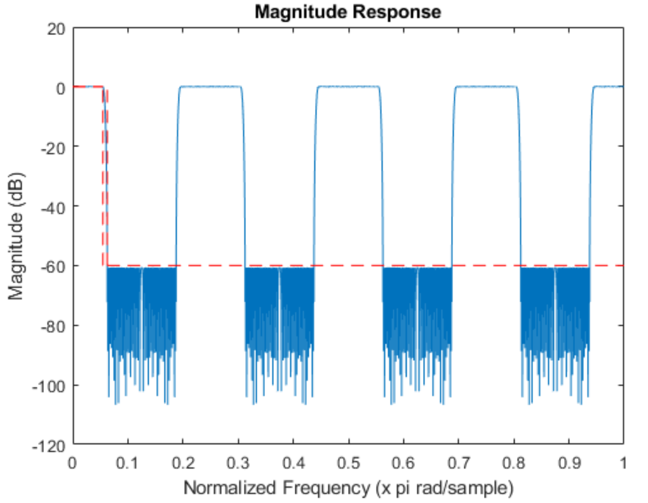
I(z)
Zoomed View of Passband



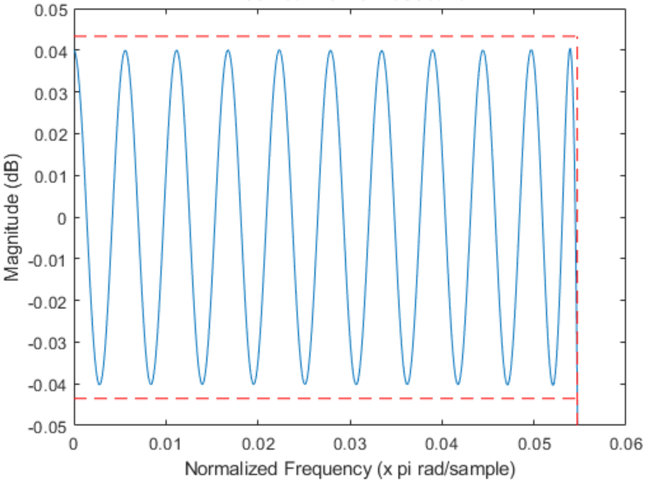
I(z)
Zoomed View of Stopband



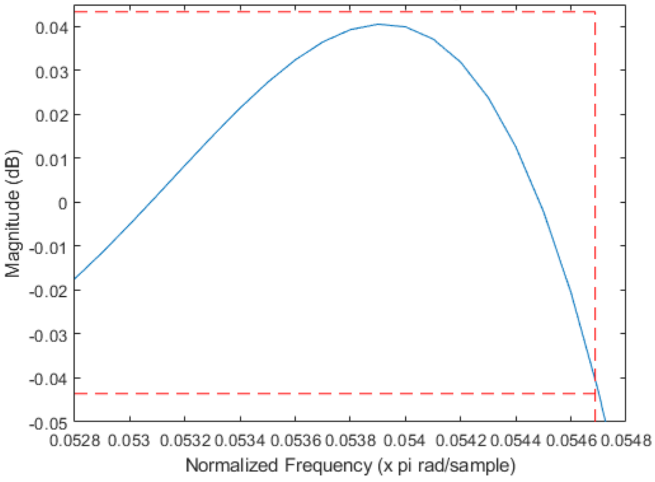
$G(z^8)$
Magnitude Response



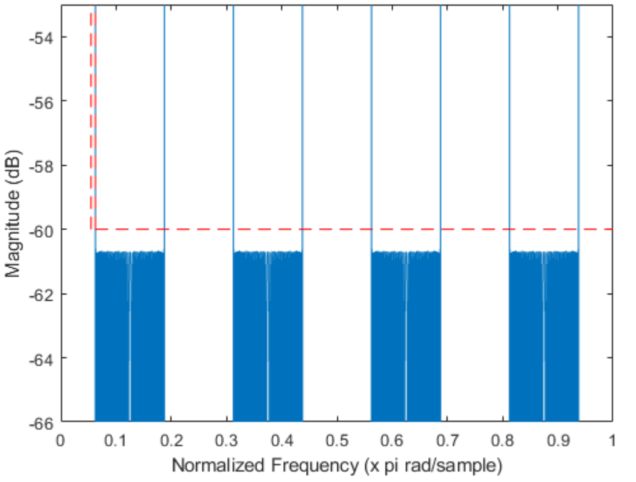
$G(z^8)$
Zoomed View of Passband



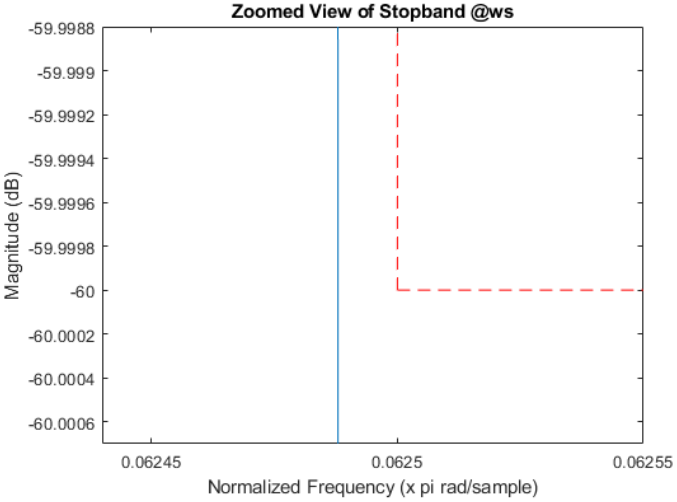
$G(z^8)$
Zoomed View of Passband @ wp



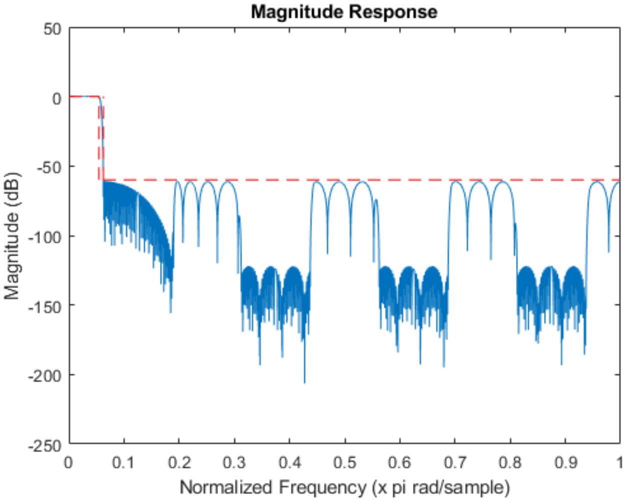
$G(z^8)$
Zoomed View of Stopband



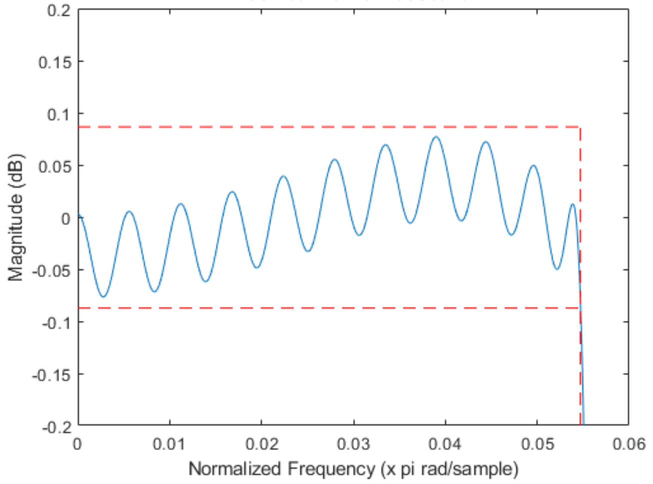
$G(z^8)$
Zoomed View of Stopband @ws



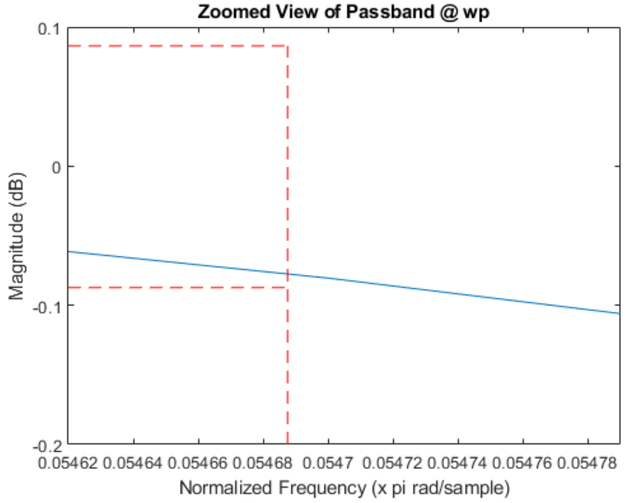
$I(z)G(z^8)$
Magnitude Response



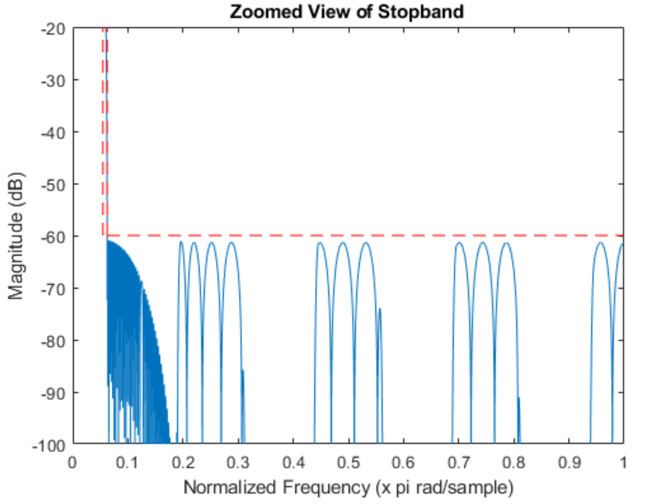
$I(z)G(z^8)$
Zoomed View of Passband



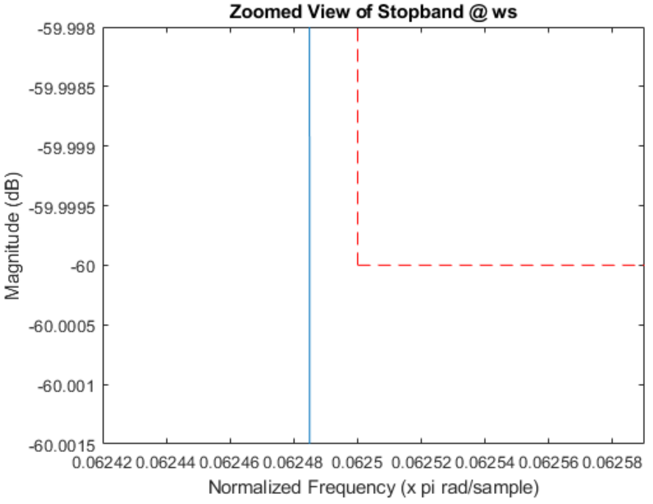
$I(z)G(z^8)$
Zoomed View of Passband @ wp



$I(z)G(z^8)$
Zoomed View of Stopband



$I(z)G(z^8)$
Zoomed View of Stopband @ ws

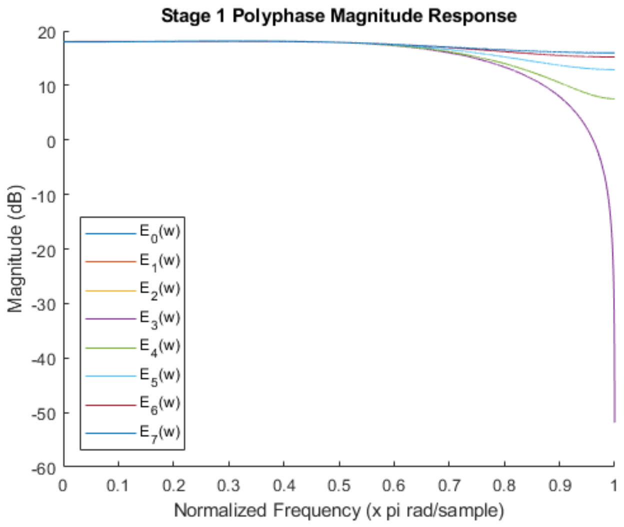


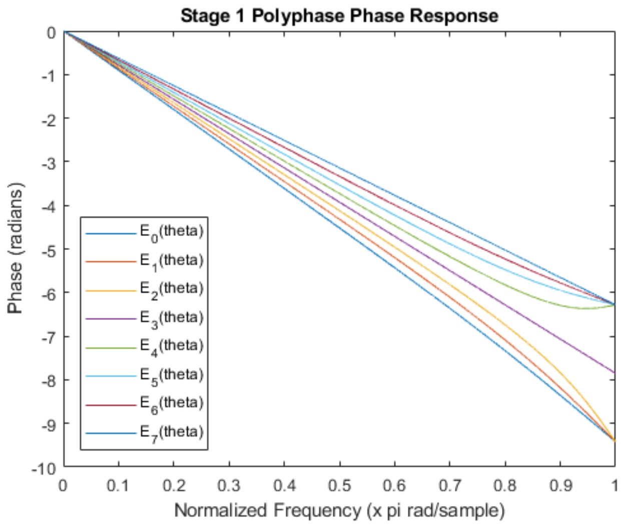
Problem 2

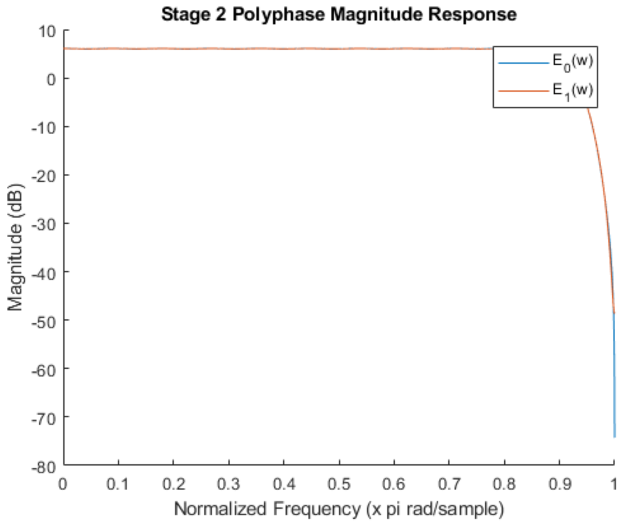
Determine the Type I polyphase representation for each filter stage designed in Part 1 and plot the magnitude and phase response for each polyphase section.

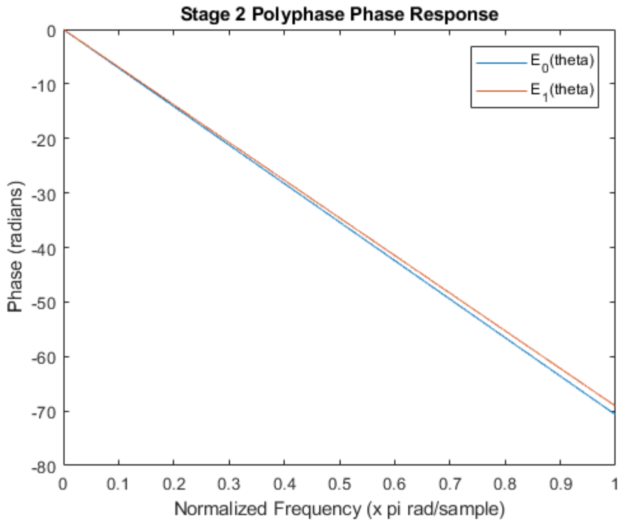
```
% Impulse responses from problem 1
hi = M1*b_i;    % I(z) impulse response
hg = M2*b_g;    % G(z) impulse response

%************ STAGE 1 I(z) ***********************************************
% Compute impulse response, frequency response, and phase
% phase response for each stage 1 polyphase component. Since length of
% hi(8:M1:end) is less than the others i need to add if/else statements to
% handle it.
% ei --> Impulse response
% EZ --> Frequency response
% PZ --> Phase response
ei = zeros(round(length(hi)/M1),M1);
EZ = zeros(10000,M1);
PZ = zeros(10000,M1);
for i = 1:M1
    if i == 8
        ei(1:end-1,i) = M1*hi(i:M1:end);
        EZ(:,i) = freqz(ei(1:end-1,i),1,10000);
        PZ(:,i) = phasez(ei(1:end-1,i),1,10000);
    else
        ei(:,i) = M1*hi(i:M1:end);
        EZ(:,i) = freqz(ei(:,i),1,10000);
        PZ(:,i) = phasez(ei(:,i),1,10000);
    end
end

% Plot magnitude response of polyphase components
figure(16)
hold on
w = linspace(0,pi,10000);
plot(w/pi,20*log10(abs(EZ)));
title('Stage 1 Polyphase Magnitude Response');xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
legend('E_0(w)','E_1(w)','E_2(w)','E_3(w)','E_4(w)','E_5(w)','E_6(w)','E_7(w)','Location','southwest');

% Plot phase response of polyphase components
figure(17)
plot(w/pi,PZ);
title('Stage 1 Polyphase Phase Response');xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Phase (radians)');
legend('E_0(theta)','E_1(theta)','E_2(theta)','E_3(theta)','E_4(theta)','E_5(theta)','E_6(theta)','E_7(theta)','
Location','southwest');

%************ STAGE 2 G(z) ***********************************************
% Compute polyphase impulse response components for stage 2 (G(z))
e0 = M2*(hg(1:M2:end));%/sum(hg(1:M2:end)));
e1 = M2*(hg(2:M2:end));%/sum(hg(2:M2:end)));

% Compute frequency response for each stage 2 (G(z)) polyphase component
EZ = zeros(10000,M2);
EZ(:,1) = freqz(e0,1,10000);
EZ(:,2) = freqz(e1,1,10000);
```

```
% Compute phase response for each stage 2 (G(z)) polyphase component
PZ = zeros(10000,M2);
PZ(:,1) = phasez(e0,1,10000);
PZ(:,2) = phasez(e1,1,10000);

% Plot magnitude response of polyphase components
figure(19)
w = linspace(0,pi,10000);hold on
plot(w/pi,20*log10(abs(EZ)));
title('Stage 2 Polyphase Magnitude Response');xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
legend('E_0(w)','E_1(w)');

% Plot phase response of polyphase components
figure(20)
plot(w/pi,PZ);
title('Stage 2 Polyphase Phase Response');xlabel('Normalized Frequency (x pi rad/sample)');
ylabel('Phase (radians)');
legend('E_0(theta)','E_1(theta)');
```









Published with MATLAB® R2017b