ECE 513 - DIGITAL SIGNAL PROCESSING

HOMEWORK 10 Nazneen Percy Kotwal
(200205655)

1. Matlab Code:

```
%% Homework 10 Question 1
clc
clear all
close all
%% P(z) is a Half Band, Zero Phase Filter
syms a b c d z
g = expand((1+z)^4);
h = expand((1+(z^-1))^4);
i = expand((a*z^3)+(b*z^2)+(c*z)+d+(c*z^-1)+(b*z^-2)+(a*z^-3));
coeffs_z = collect((h*g*i), z);
display(coeffs_z)
% Determining the values of coefficients a b c and d
% P0 = 1 , Even Powers of z have coefficients zero.
% Solving the four equations
eqns =[16*a + 56*b + 112*c + 70*d - 1 == 0, 56*a + 28*b + 8*c + d == 0, 56*a + 71*b +
64*c + 28*d == 0, 8*a + b == 0];
vars = [a b c d];
[sola, solb, solc, sold] = solve(eqns, vars);
% Substituting the values of a, b, c, d to obtain H0, H1, G0, G1
r1 = expand((sola*z^3)+(solb*z^2)+(solc*z)+sold+(solc*z^-1)+(solb*z^-2)+(sola*z^-3));
coeffs_z1 = collect((h*g*r1*(z^-7)), z); % Factorizing out z^k
factor_z1 = factor(coeffs_z1);
display(factor_z1)
% H0(z) and G0(z) are 8-tap symmetric linear phase lowpass filters
coeffs_h0 = collect(((z^-7)*((z+1)^7)*(1/32)), z);
coeffs_g0 = collect(((5*z^6 - 40*z^5 + 131*z^4 - 208*z^3 + 131*z^2 - 40*z +
5)*(z+1)*(z^-7)*(-1/64)), z);
% H1(z) =  G0(-z)
% G1(z) = -H0(-z)
coeffs_h1 = collect(((5 - (35*(-z)^-1) + (91*z^-2) - (77*(-z)^-3) - (77*z^-4) + (91*(-
z)^-5) - (35*z^-6) + (5*(-z)^-7))*(-1/64)), z);
coeffs_g1 = collect(((1 + (7*(-z)^-1) + (21*z^-2) + (35*(-z)^-3) + (35*z^-4) + (21*(-
z)^-5) + (7*z^-6) + ((-z)^-7))*(1/32)*(-1)), z);
% Perfect Reconstruction Filters:
% A(z) = (1/2){(H0(-z)*G0(z))+(H1(-z)*G1(z))}
coeffs_h11 = collect(((5 - (35*(z)^-1) + (91*z^-2) - (77*(z)^-3) - (77*z^-4) +
(91*(z)^-5) - (35*z^-6) + (5*(z)^-7))*(-1/64)),z);
display(coeffs_h11)
coeffs_h01 = collect(((1 + (7*(-z)^-1) + (21*z^-2) + (35*(-z)^-3) + (35*z^-4) + (21*(-
z)^-5) + (7*z^-6) + ((-z)^-7))*(1/32)),z);
display(coeffs_h01)

% Coefficients of A(z)= 0
coeffs_az = collect (((coeffs_h01*coeffs_g0)+(coeffs_g1*coeffs_h11)),z);
display(coeffs_az)

% Coefficients of T(z)= C(z^-k)
coeffs_tz = collect(((1/2)*((coeffs_h0*coeffs_h11)-(coeffs_h1*coeffs_h01))),z);
display(coeffs_tz)
```

```matlab
%% Magnitude plots for Analysis Filter
mag_h1 = [-5  -35  -91  -77  77 91  35 5]./64;
mag_h0 = [1 7  21 35  35 21  7 1]./32;

figure(1)
[h1,w1]=freqz(mag_h0);
hmag_h0 = abs(h1);
tmp = angle(h1);
tol = 0.85*pi;
hang1 = unwrap(tmp, tol);
plot(w1,hmag_h0,'--')
hold on;

[h2,w2]=freqz(mag_h1,1);
hmag_h1 = abs(h2);
tmp = angle(h2);
tol = 0.85*pi;
hang2 = unwrap(tmp, tol);
plot(w2,hmag_h1)
legend('|H0(w)|','|H1(w)|')
grid on;
xlabel('Frequency (radians)','fontsize',12);
ylabel('Magnitude |H(w)|','fontsize',12);
title('Magnitude Plot for Analysis Filter');

figure(3)
plot(w1,hang1,'--')
hold on;
plot(w2,hang2)
legend('H0(w)','H1(w)')
grid on;
title('Phase Plot for Analysis Filter')
xlabel('Frequency (radians)')
ylabel('Phase (radians)')


%% Magnitude plots for Synthesis Filter
mag_g1 = [-1 7 -21 35 -35 21 -7 1]./32;
mag_g0 = [-5 35 -91 77  77 -91 35 -5]./64;

figure(2)
[g1,w1]=freqz(mag_g0);
hmag_g0 = abs(g1);
tmp = angle(g1);
tol = 0.85*pi;
hang3 = unwrap(tmp, tol);
plot(w1,hmag_g0,'--')
hold on;

[g2,w2]=freqz(mag_g1,1);
hmag_g1 = abs(g2);
tmp = angle(g2);
tol = 0.85*pi;
hang4 = unwrap(tmp, tol);
plot(w2,hmag_g1)
legend('|G0(w)|','|G1(w)|')
grid on;
xlabel('Frequency (radians)','fontsize',12);
ylabel('Magnitude |G(w)|','fontsize',12);
title('Magnitude Plot for Synthesis Filter');
```

```
figure(4)
plot(w1,hang3,'--')
hold on
plot(w2,hang4)
legend('G0(w)','G1(w)')
grid on;
title('Phase Plot for Synthesis Filter')
xlabel('Frequency (radians)')
ylabel('Phase (radians)')
```

Results:

Solution foe the value of coefficients of R(z):

a = -5/2048

b = 5/256

c = -131/2048

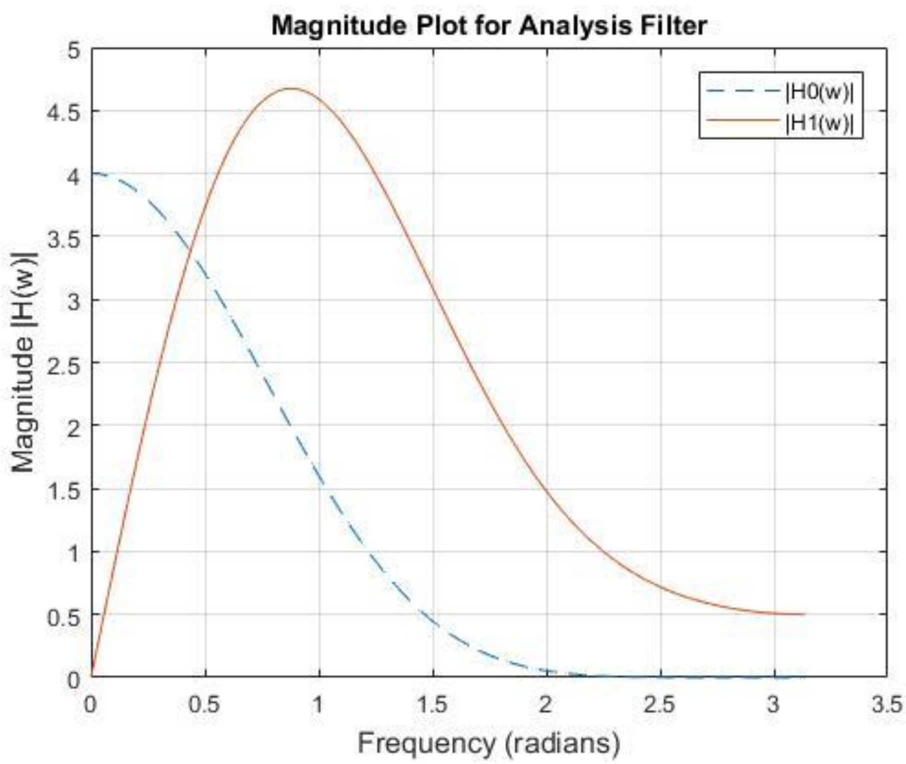
d = 13/128

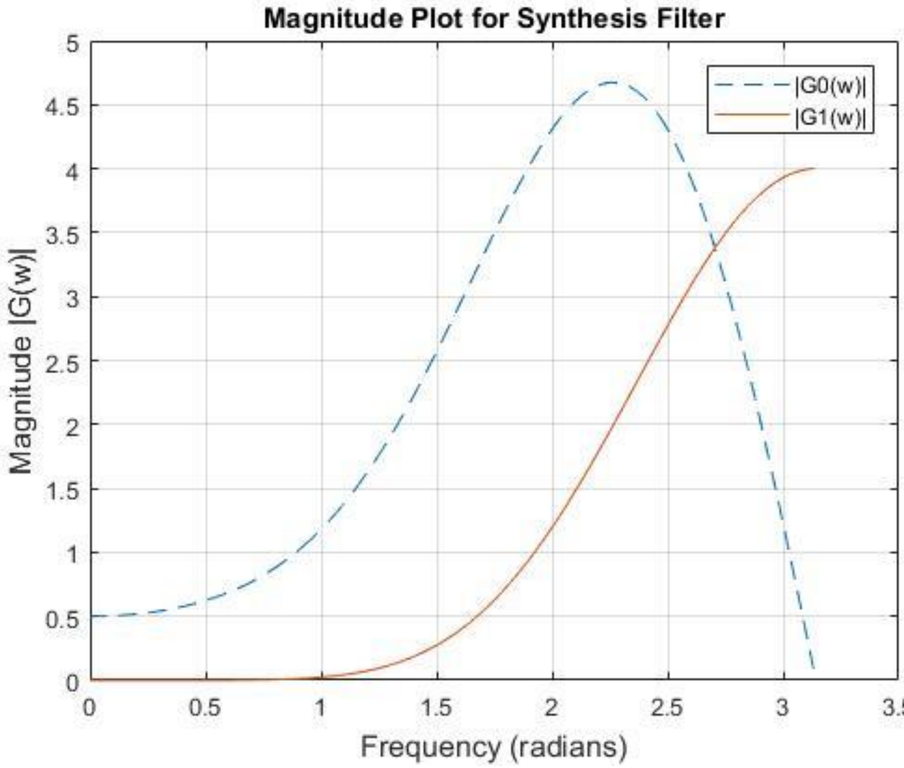
To prove the filters are perfect reconstruction filters:

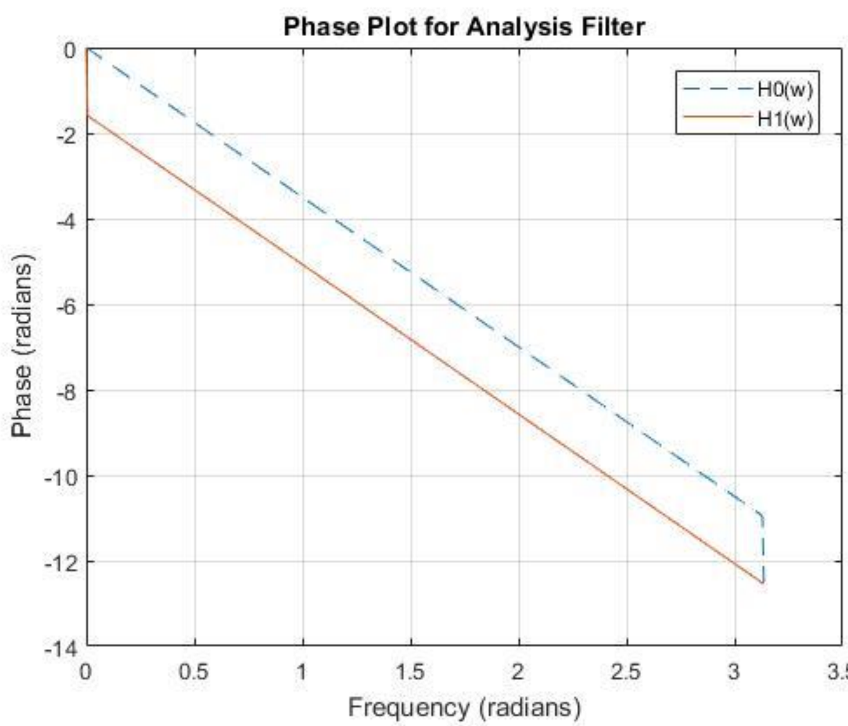
coeffs_az = 0

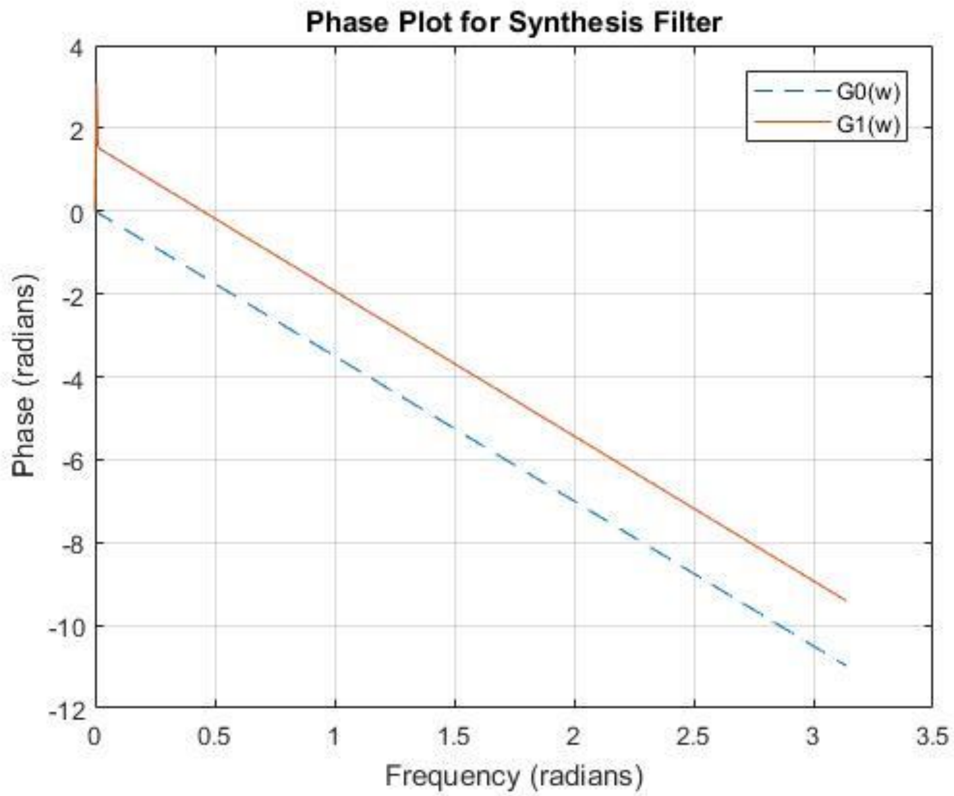
coeffs_tz = 1/z^7

Matlab Plots:




Magnitude Plot for Synthesis Filter
Magnitude |G(w)|
Frequency (radians)
|G0(w)|
|G1(w)|
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Phase Plot for Analysis Filter
Phase (radians)
Frequency (radians)
H0(w)
H1(w)
0
-2
-4
-6
-8
-10
-12
-14
0.5
1
1.5
2
2.5
3
3.5



2. Matlab Code:

```
clc
clear all;
close all;

syms a b c d z
g = expand((1+z)^4);
h = expand((1+(z^-1))^4);
i = expand((a*z^3)+(b*z^2)+(c*z)+d+(c*z^-1)+(b*z^-2)+(a*z^-3));
coeffs_z = collect((h*g*i), z);
display(coeffs_z)
% Determining the values of coefficients a b c and d
% P0 = 1 , Even Powers of z have coefficients zero.
% Solving the four equations
eqns =[16*a + 56*b + 112*c + 70*d - 1 == 0, 56*a + 28*b + 8*c + d == 0, 56*a + 71*b +
64*c + 28*d == 0, 8*a + b == 0];
vars = [a b c d];
[sola, solb, solc, sold] = solve(eqns, vars);
R = [sola solb solc sold solc solb sola];
Q = vpa(roots(R));

A = [1 1];
R1 = [1 -0.32887591778603086663314159546921];
R2 = [1 -(0.28409629819182161933311014347123+0.24322822591037988274771916445887i)];
R3 = [1 -(0.28409629819182161933311014347123-0.24322822591037988274771916445887i)];
R = conv(R1,conv(R2,R3));
h0 = conv(A,conv(A,conv(A,conv(A,R))));
H0 = h0;
G0 = flip(H0);
H1 = zeros(1,length(G0));
```

```
G1 = zeros(1,length(H0));
factor1 = 1/32;
factor2 = 1/64;
for i = 1:1:length(H0)
    if (mod(i,2) == 1)
        H1(i) = G0(i);
        G1(i) = H0(i);
    else
        H1(i) = -G0(i);
        G1(i) = -H0(i);
    end
end
H0 = factor1.*H0;
G0 = factor2.*G0;
H1 = (factor2).*H1;
G1 = (-factor1).*G1;

figure(1)
[h0,w0] = freqz(H0,1);
scale = sqrt(2)/max(abs(h0));
H0 = scale*H0;
[h0,w0] = freqz(H0,1);
tmp = angle(h0);
tol = 0.85*pi;
hang1 = unwrap(tmp, tol);
plot(w0,abs(h0))
hold on
[h1,w1] = freqz(H1,1);
scale = sqrt(2)/max(abs(h1));
H1 = scale*H1;
[h1,w1] = freqz(H1,1);
tmp = angle(h1);
tol = 0.85*pi;
hang2 = unwrap(tmp, tol);
plot(w1,abs(h1))
legend('|H0(w)|','|H1(w)|')
grid on;
xlabel('Frequency (radians)','fontsize',12);
ylabel('Magnitude |H(w)|','fontsize',12);
title('Magnitude Plot for Analysis Filter');

figure(2)
plot(w0,hang1)
hold on
plot(w1,hang2)
legend('H0(w)','H1(w)')
grid on;
title('Phase Plot for Analysis Filter')
xlabel('Frequency (radians)')
ylabel('Phase (radians)')

figure(3)
[g0,w0] = freqz(G0,1);
scale = sqrt(2)/max(abs(g0));
G0 = scale*G0;
[g0,w0] = freqz(G0,1);
tmp = angle(g0);
tol = 0.85*pi;
hang3 = unwrap(tmp, tol);
plot(w0,abs(g0))
hold on
[g1,w1] = freqz(G1,1);
```

```
scale = sqrt(2)/max(abs(g1));
G1 = scale*G1;
[g1,w1] = freqz(G1,1);
tmp = angle(g1);
tol = 0.85*pi;
hang4 = unwrap(tmp, tol);
plot(w1,abs(g1))
legend('|G0(w)|','|G1(w)|')
grid on;
xlabel('Frequency (radians)','fontsize',12);
ylabel('Magnitude |G(w)|','fontsize',12);
title('Magnitude Plot for Synthesis Filter');

figure(4)
plot(w0,hang3)
hold on
plot(w1,hang4)
legend('G0(w)','G1(w)')
grid on;
title('Phase Plot for Synthesis Filter')
xlabel('Frequency (radians)')
ylabel('Phase (radians)')

T1 = conv(H0,G0);
T2 = conv(H1,G1);
T = 1/2 * (T1 + T2);
Tz = poly2sym(T,z);
Tz = sym2poly(Tz);
A1 = conv(G0,-G1);
A2 = conv(G1,G0);
A = 1/2 * (A1 + A2);
Az = poly2sym(A,z);
Az = sym2poly(Az);
```

Results:

To prove the filters are perfect reconstruction filters:

Az = [-1.7347e-18 -3.469e-18 0 -3.4694e-18 2.7755e-17 1.7347e-18 -2.7755e-17

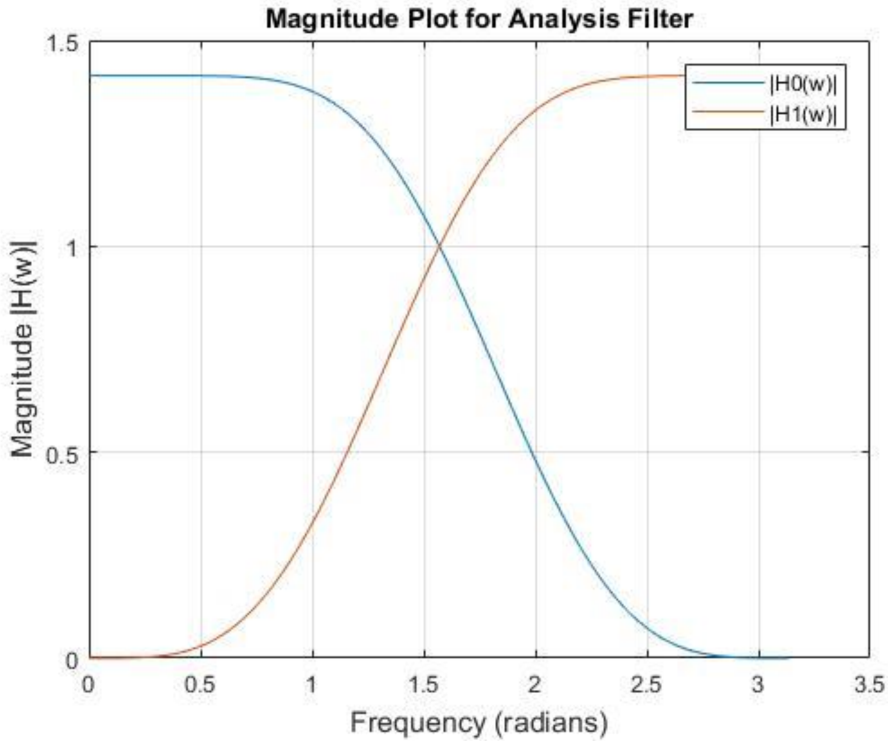
-3.4694e-18 0 -3.4694e-18 -1.734721e-18 0 0]

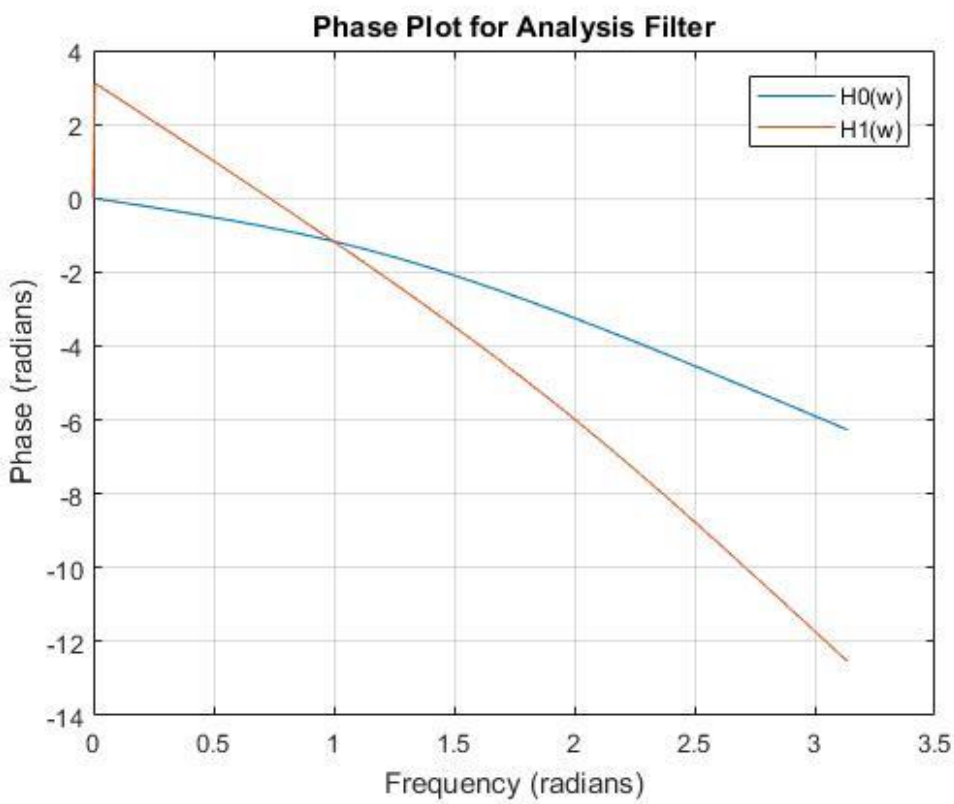
Tz = [1.734e-18 1.7347e-18 -4.9249e-17 6.9388e-18 -1.0500e-16 0 1 0 -

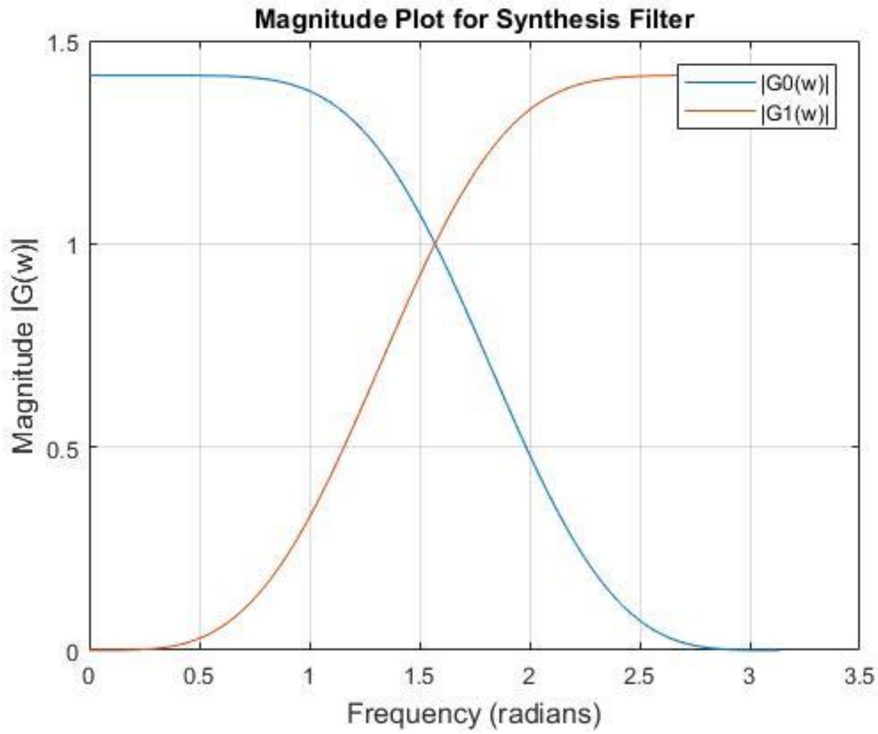
5.6866e-17 6.9388e-18 -4.9276e-17 -1.7347e-18 1.73471e-18 0]

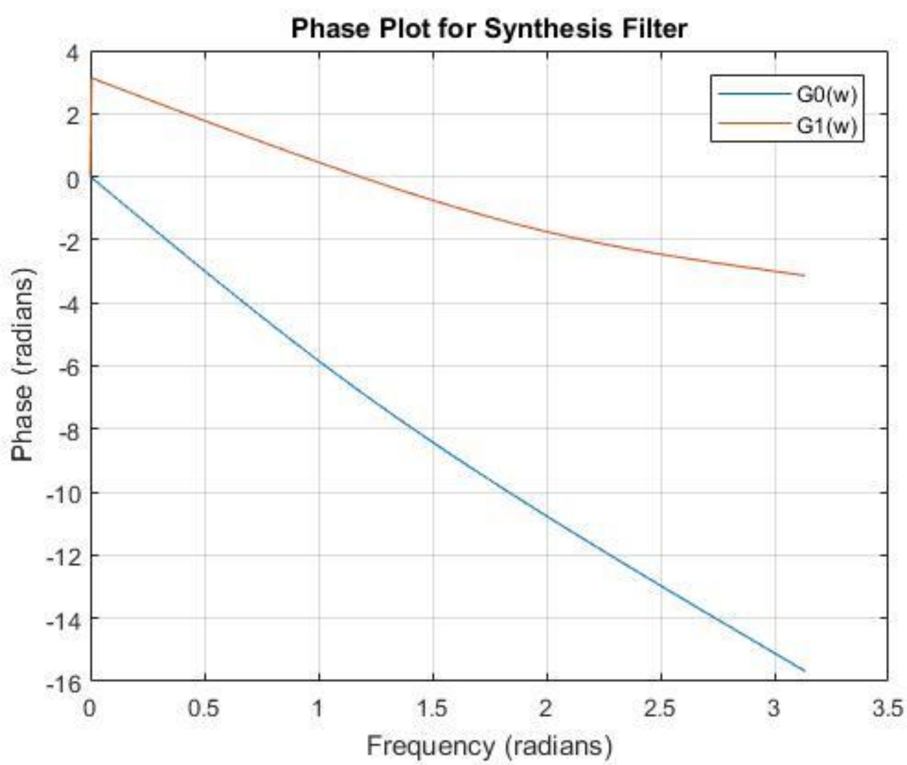
Since the values are of the order of 10^-17 we consider them to be negligible.

Matlab Plots:






Magnitude Plot for Synthesis Filter
|G0(w)|
|G1(w)|
Magnitude |G(w)|
Frequency (radians)

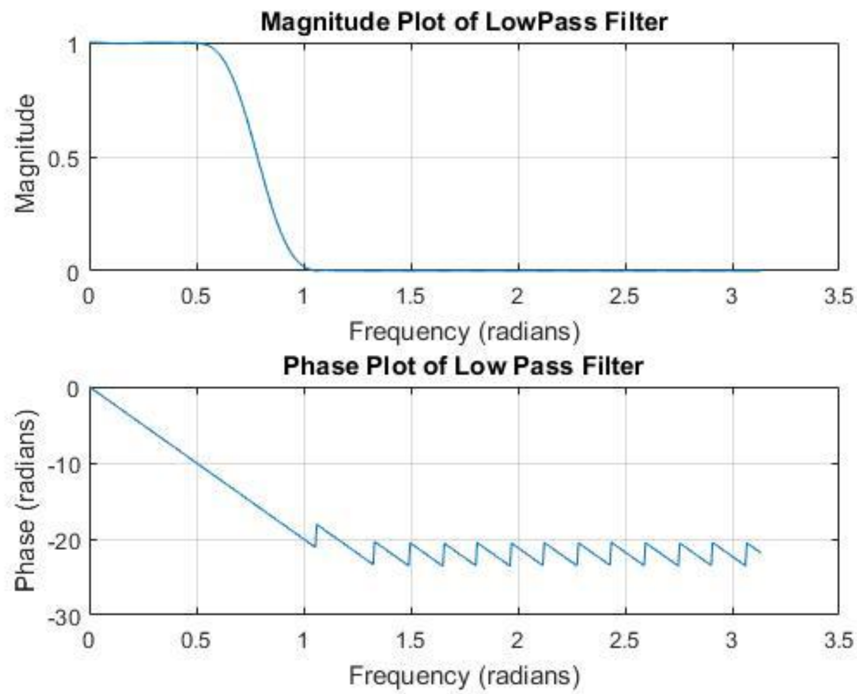

Phase Plot for Synthesis Filter
G0(w)
G1(w)
Phase (radians)
Frequency (radians)

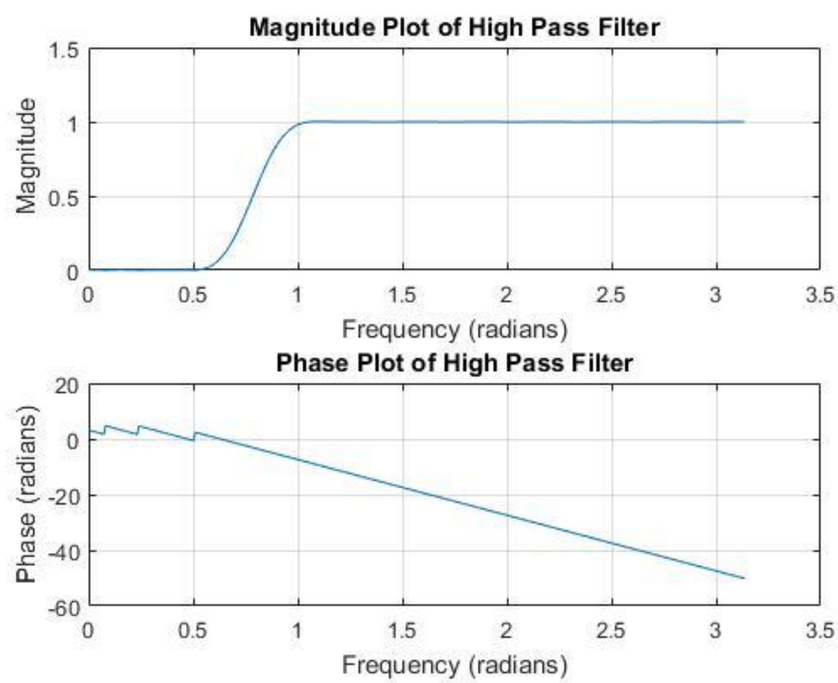
3. Matlab Code:

```
% Homework10 Question 3
Wn = 1/4;
n = 40;
h0_low = fir1(n,Wn,'low');
h1_high = fir1(n,Wn,'high');
[h1,w1] = freqz(h0_low,1);
hmag = abs(h1);
tmp = angle(h1);
tol = 0.85*pi;
hang = unwrap(tmp, tol);
figure(1)
subplot(2,1,1)
plot(w1, hmag);
grid on;
title('Magnitude Plot of LowPass Filter');
xlabel('Frequency (radians)')
ylabel(' Magnitude')
subplot(2,1,2)
plot(w1, hang);
grid on;
title('Phase Plot of Low Pass Filter')
xlabel('Frequency (radians)')
ylabel('Phase (radians)')

[h2,w2] = freqz(h1_high,1);
hmag = abs(h2);
tmp = angle(h2);
tol = 0.85*pi;
hang = unwrap(tmp, tol);
figure(2)
subplot(2,1,1)
plot(w2, hmag);
grid on;
title('Magnitude Plot of High Pass Filter');
xlabel('Frequency (radians)')
ylabel(' Magnitude')
subplot(2,1,2)
plot(w2, hang);
grid on;
title('Phase Plot of High Pass Filter')
xlabel('Frequency (radians)')
ylabel('Phase (radians)')

h00 = conv(h0_low,conv(upsample(h0_low,2),upsample(h0_low,4)));
h01 = conv(h0_low,conv(upsample(h0_low,2),upsample(h1_high,4)));
h02 = conv(h0_low,upsample(h1_high,2));
h03 = conv(h1_high,upsample(h1_high,2));
h04 = conv(h1_high,upsample(h0_low,2));
[h3,w3] = freqz(h00,1);
hmag3 = abs(h3);
[h4,w4] = freqz(h01,1);
hmag4 = abs(h4);
[h5,w5] = freqz(h02,1);
hmag5 = abs(h5);
[h6,w6] = freqz(h03,1);
hmag6 = abs(h6);
[h7,w7] = freqz(h04,1);
hmag7 = abs(h7);
figure(3)
plot(w3,hmag3,w4,hmag4,w5,hmag5,w6,hmag6,w7,hmag7)
title('Magnitude Plot');
```

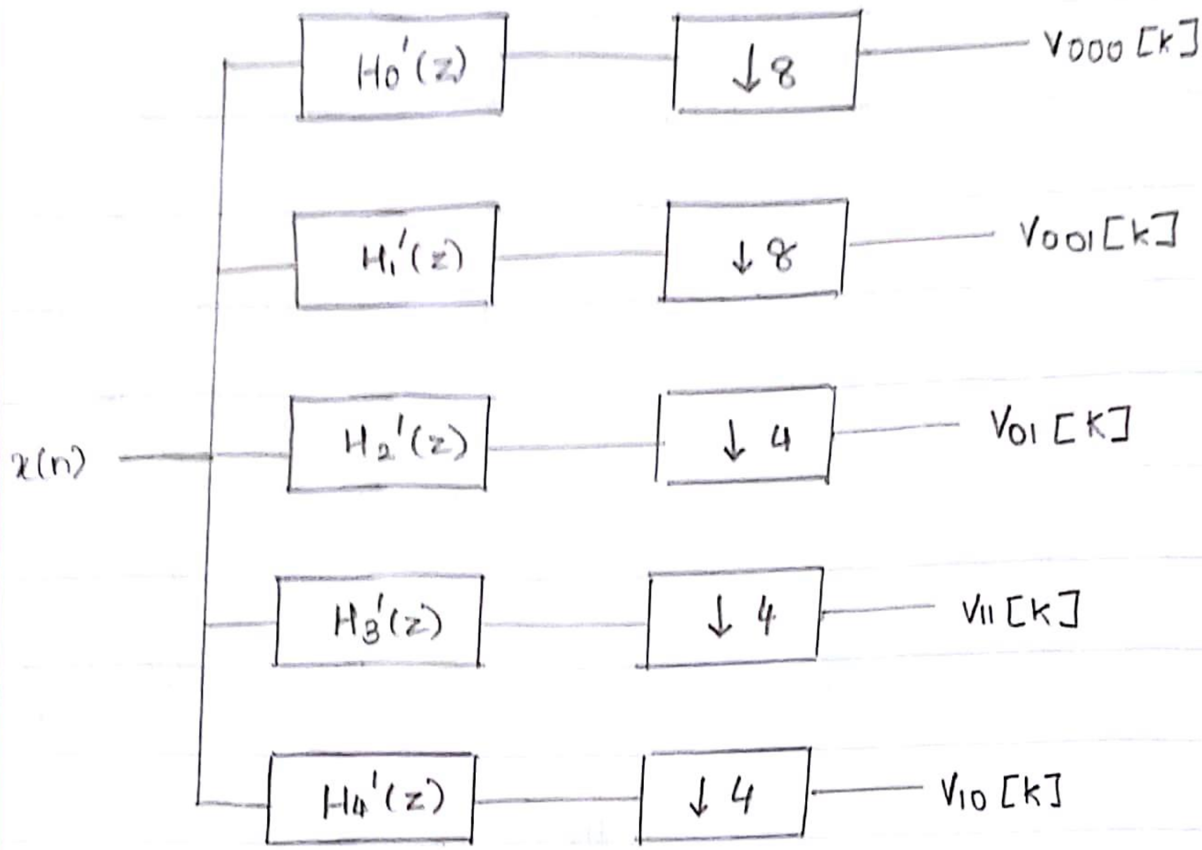
```
xlabel('Frequency (radians)')
ylabel('Magnitude')
grid on;
```

Matlab Plots:





b.



$$H_0'(z) = H_0(z)\,H_0(z^2)\,H_0(z^4)$$

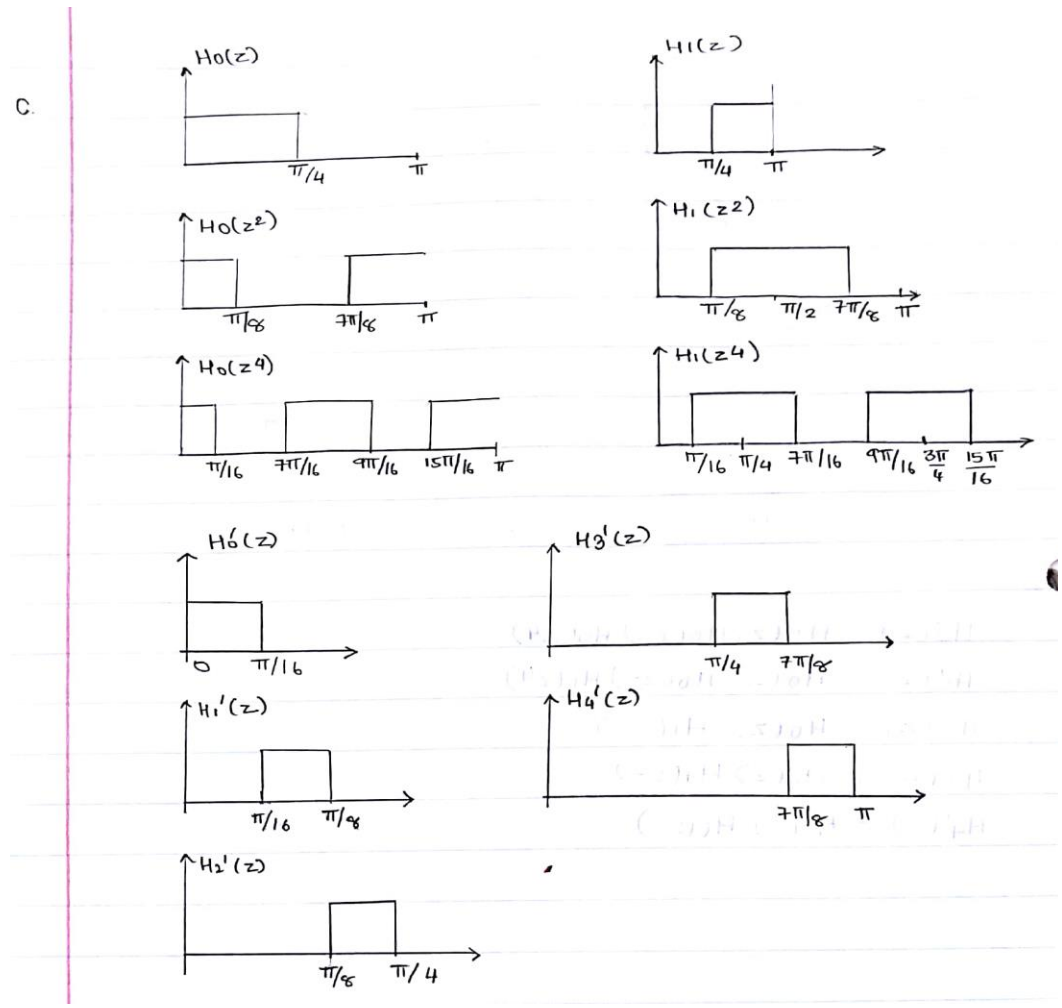
$$H_1'(z) = H_0(z)\,H_0(z^2)\,H_1(z^4)$$

$$H_2'(z) = H_0(z)\,H_1(z^2)$$

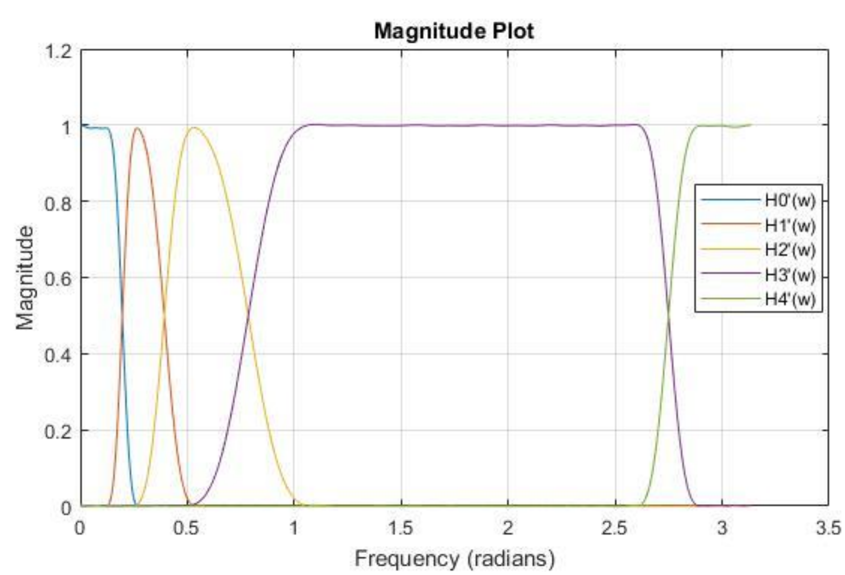
$$H_3'(z) = H_1(z)\,H_1(z^2)$$

$$H_4'(z) = H_1(z)\,H_0(z^2)$$

c.



d.



Comment: Apart from the obvious overlap, the expected plot and the Matlab plot match in both placement and proportion of Bandwidths

4. Matlab Code:

```
% Homework10 Question 4
clc
clear all
close all

%% Filters from problem 2
h0 = [0.2304    0.7148    0.6309   -0.0280   -0.1870    0.0308    0.0329   -
0.0106];
g0 = [-0.0106    0.0329    0.0308   -0.1870   -0.0280    0.6309    0.7148
0.2304];
h1 = [-0.0106   -0.0329    0.0308    0.1870   -0.0280   -0.6309    0.7148   -
0.2304];
g1 = [-0.2304    0.7148   -0.6309   -0.0280    0.1870    0.0308   -0.0329   -
0.0106];

%% Testing The algorithm
xold = sampdata();
[v000,v001,v01,v10,v11] = part1(xold,h0,h1);
yold = part2(v000,v001,v01,v10,v11,g0,g1);
figure(1)
subplot(2,1,1)
plot(1:1:length(xold),xold)
title('Original Sampdata')
xlabel('time')
ylabel('Amplitude')
grid on;
subplot(2,1,2)
plot(1:1:length(yold),yold)
title('Shifted Sampdata')
xlabel('time')
ylabel('Amplitude')
grid on;

%% load the noise free signal
load hwk10_signals.mat xn_prb2
figure(2)
subplot(2,1,1)
plot(1:1:length(xn_prb2),xn_prb2)
grid on;
title('Noise Free Sound')
xlabel('time')
ylabel('Amplitude')
%soundsc(xn_prb2)
subplot(2,1,2)
% load the noisy signal
xn_prb2_noisy = xn_prb2 + 0.1*randn(size(xn_prb2));
x = xn_prb2_noisy;
%soundsc(xn_prb2_noisy)
subplot(2,1,2)
plot(1:1:length(xn_prb2_noisy),xn_prb2_noisy)
title('Noisy Sound')
xlabel('time')
ylabel('Amplitude')
grid on;

%% Analysis Filtering
[v000,v001,v01,v10,v11] = part1(x,h0,h1);
```

```
%% Hard Thresholding 1
[Vnew001]= hard_thresh(v001,0.390);
[Vnew01]= hard_thresh(v01,0.390);
[Vnew10]= hard_thresh(v10,0.390);
[Vnew11]= hard_thresh(v11,0.353);

% Hard Thresholding 2
[Vnew2001]= hard_thresh(v001,0.200);
[Vnew201]= hard_thresh(v01,0.130);
[Vnew210]= hard_thresh(v10,0.120);
[Vnew211]= hard_thresh(v11,0.120);

% Hard Thresholding 3
[Vnew3001]= hard_thresh(v001,1.900);
[Vnew301]= hard_thresh(v01,0.960);
[Vnew310]= hard_thresh(v10,0.950);
[Vnew311]= hard_thresh(v11,0.930);

%% Synthesis Filtering
y = part2(v000,Vnew001,Vnew01,Vnew10,Vnew11,g0,g1);
%soundsc(y)
figure(4)
subplot(2,1,1)
plot(1:1:length(x),x)
title('Noisy Sound')
xlabel('time')
ylabel('Amplitude')
grid on;
subplot(2,1,2)
plot(1:1:length(y),y)
title('Sound After Thresholding (1)')
xlabel('time')
ylabel('Amplitude')
grid on;

y2 = part2(v000,Vnew2001,Vnew201,Vnew210,Vnew211,g0,g1);
%soundsc(y2)
figure(5)
subplot(2,1,1)
plot(1:1:length(x),x)
title('Noisy Sound')
xlabel('time')
ylabel('Amplitude')
grid on;
subplot(2,1,2)
plot(1:1:length(y2),y2)
title('Sound After Thresholding (2)')
xlabel('time')
ylabel('Amplitude')
grid on;

y3 = part2(v000,Vnew3001,Vnew301,Vnew310,Vnew311,g0,g1);
%soundsc(y3)
figure(6)
subplot(2,1,1)
plot(1:1:length(x),x)
title('Noisy Sound')
xlabel('time')
ylabel('Amplitude')
grid on;
subplot(2,1,2)
plot(1:1:length(y3),y3)
```

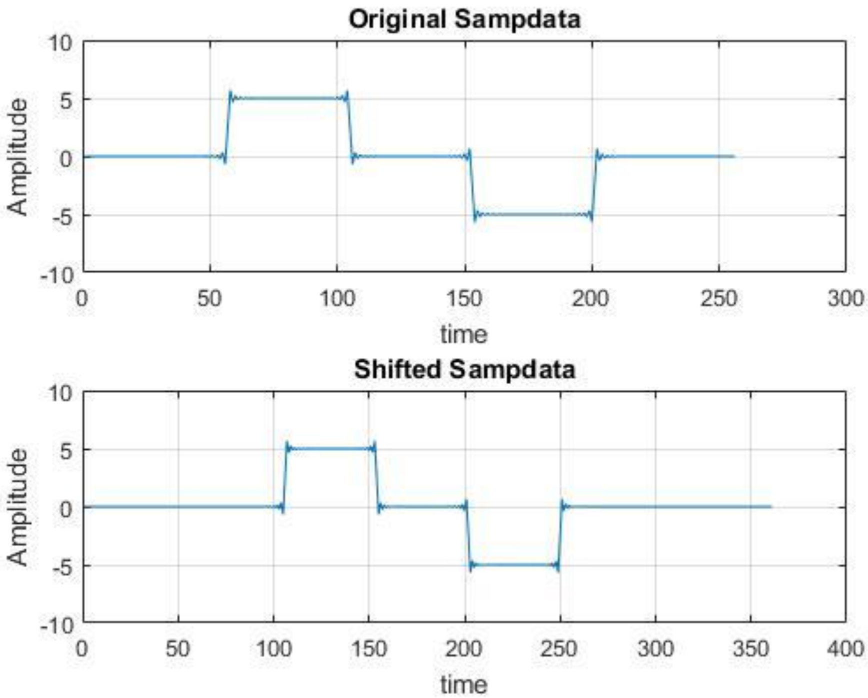
```
title('Sound After Thresholding (3)')
xlabel('time')
ylabel('Amplitude')
grid on;

%% Functions

function [v0,v1] = analyse(h0,h1,x)
v01 = conv(x,h0);
v0 = downsample(v01,2);
v11 = conv(x,h1);
v1 = downsample(v11,2);
end

function[y]= synth(x1,x2,g0,g1)
y1 = upsample(x1,2);
y2 = conv(y1,g0);
y3 = upsample(x2,2);
y4 = conv(y3,g1);
y = y4 + y2;
end

function[v000,v001,v01,v10,v11] = part1(x,h0,h1)
 [a1,b1] = analyse(h0,h1,x);
 [a2,v01] = analyse(h0,h1,a1);
 [v000,v001] = analyse(h0,h1,a2);
 [v10,v11] = analyse(h0,h1,b1);
end

function[y] = part2(v000,v001,v01,v10,v11,g0,g1)
  c1 = synth(v000,v001,g0,g1);
  d1 = synth(c1,[ zeros(1,7) v01 zeros(1,((length(c1)-length(v01))-7))],g0,g1);
  d2 = synth(v10,v11,g0,g1);
  y =  synth(d1,[zeros(1,14) d2 zeros(1,((length(d1)-length(d2))-14))],g0,g1);
end

% Hard Thresholding
function [v]= hard_thresh(x,beta_thresh)
for i = 1:1:length(x)
 if abs(x(i) <= beta_thresh)
 x(i) = 0.0;
 end
 v(i) = x(i);
end
end
```

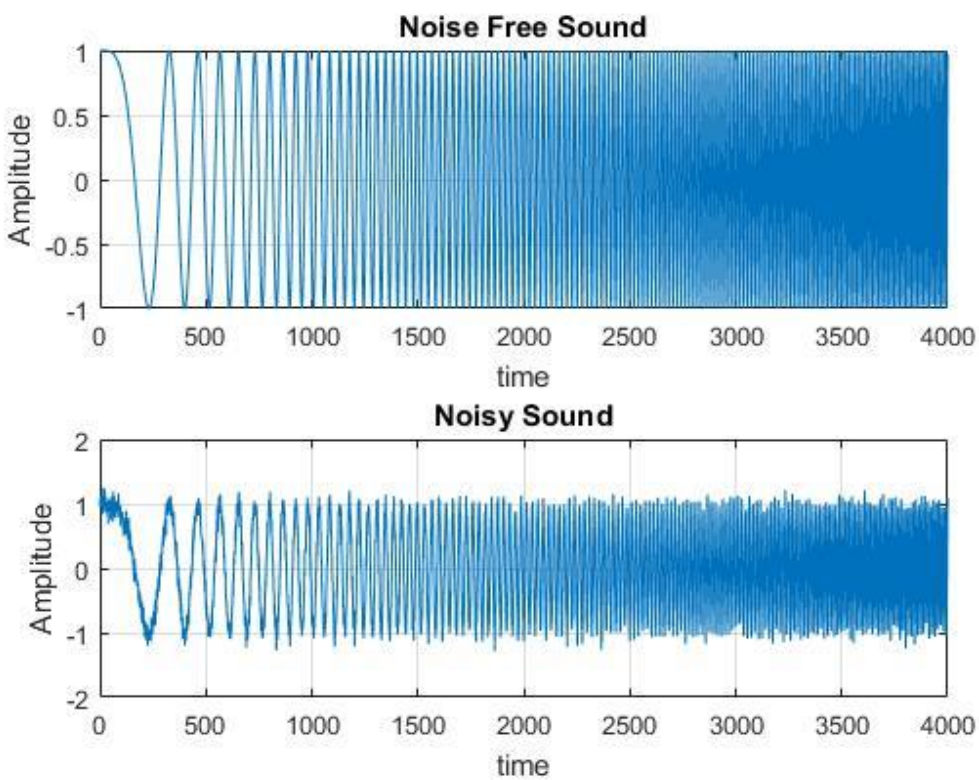
Comment:

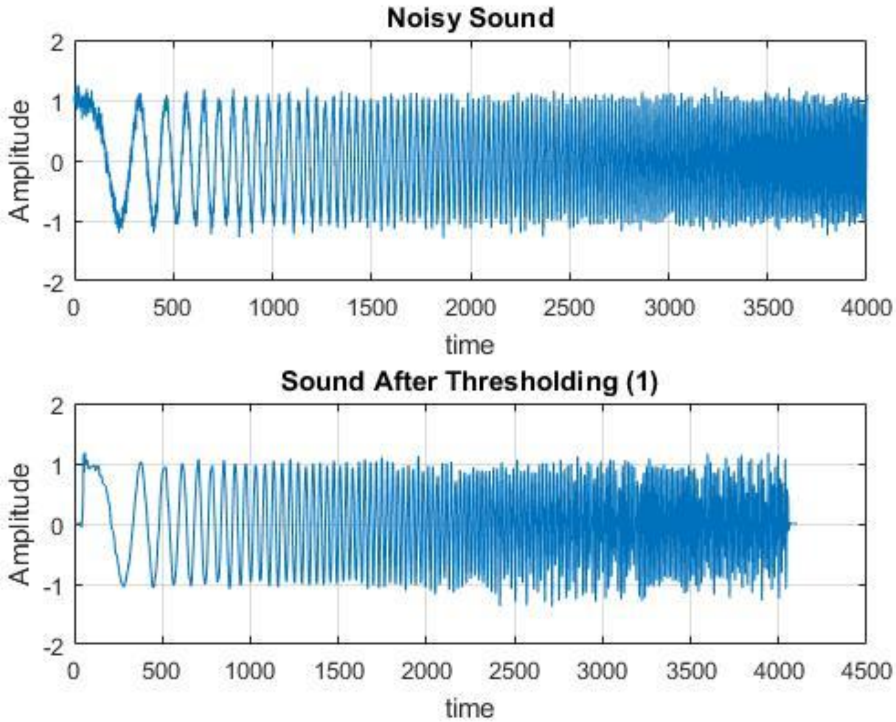
f. The denoised signal sounds better than the noisy signal using the values of the threshold given though it does not sound exactly like the original signal.

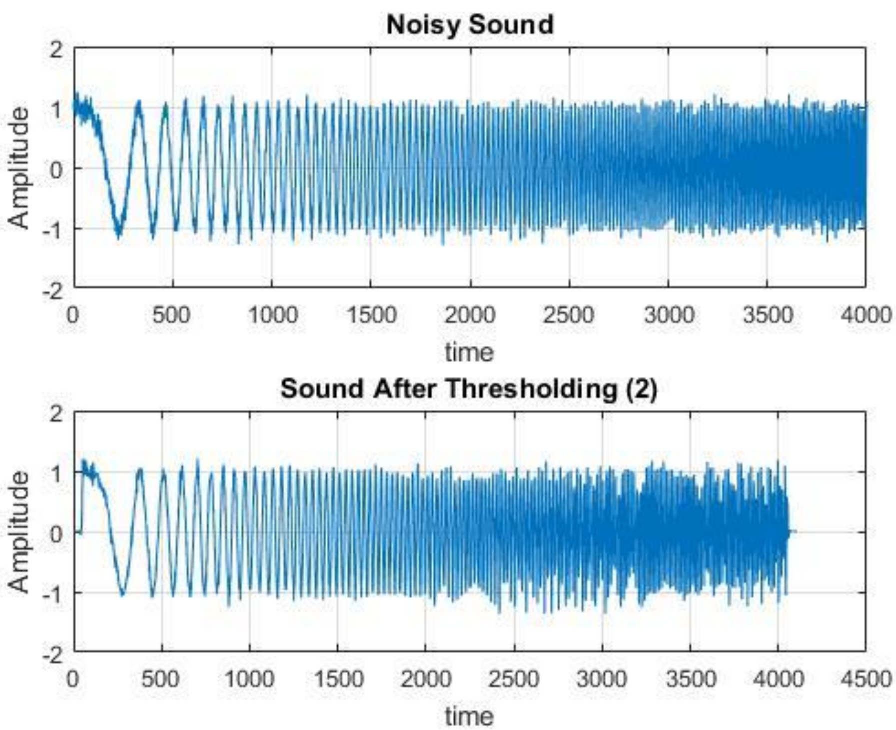
g. On increasing the value of the threshold we find that signal components that are not noise may also be falsely detected as noise and thus eliminated. A small threshold on the other hand may not be effective in eliminating the noise.

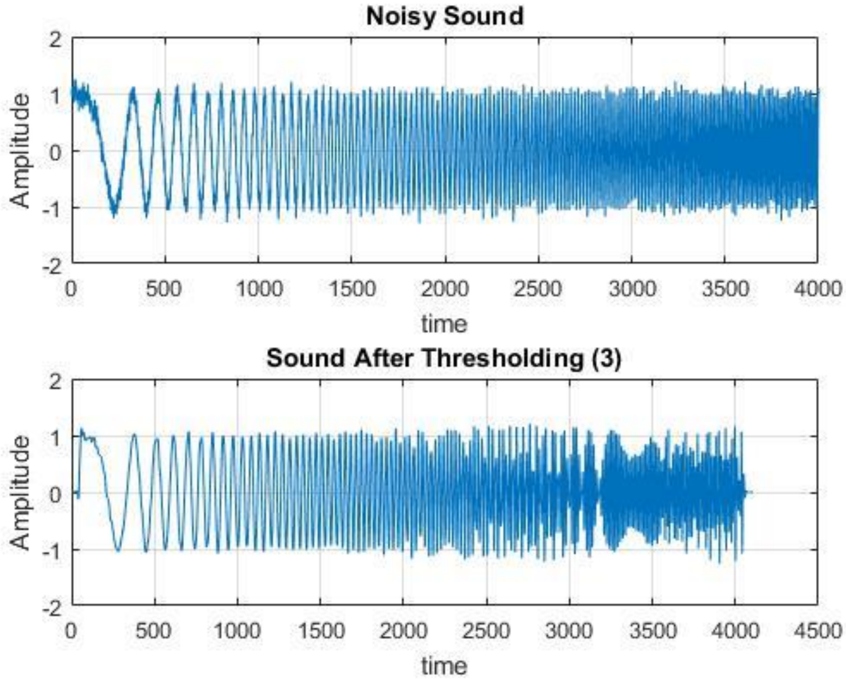
Matlab Plots:


Original Sampdata
10
5
0
-5
-10
Amplitude
0
50
100
150
200
250
300
time
Shifted Sampdata
10
5
0
-5
-10
Amplitude
0
50
100
150
200
250
300
350
400
time


Noise Free Sound
1
0.5
0
-0.5
-1
Amplitude
0
500
1000
1500
2000
2500
3000
3500
4000
time
Noisy Sound
2
1
0
-1
-2
Amplitude
0
500
1000
1500
2000
2500
3000
3500
4000
time


Noisy Sound
Amplitude
time
0
500
1000
1500
2000
2500
3000
3500
4000
2
1
0
-1
-2
Sound After Thresholding (1)
Amplitude
time
0
500
1000
1500
2000
2500
3000
3500
4000
4500
2
1
0
-1
-2


Noisy Sound
Amplitude
time
0
500
1000
1500
2000
2500
3000
3500
4000
2
1
0
-1
-2
Sound After Thresholding (2)
Amplitude
time
0
500
1000
1500
2000
2500
3000
3500
4000
4500
2
1
0
-1
-2


Noisy Sound
Amplitude
time
2
1
0
-1
-2
0
500
1000
1500
2000
2500
3000
3500
4000
Sound After Thresholding (3)
Amplitude
time
2
1
0
-1
-2
0
500
1000
1500
2000
2500
3000
3500
4000
4500