**Appendix – MATLAB source code for the entire project**

%-----------------------------%

% Sean Hwang %

% ECE 538 - MATLAB Project 2 %

% Source Code File %

%-----------------------------%

clc

clf

clear all

close all

% A) h\_0 = {1/sqrt(2), 1/sqrt(2)} ,, h\_1 = {1/sqrt(2), -1/sqrt(2)}

% Setting up M=8 channel DFT filter bank

M = 8;

h0 = [1 1] / sqrt(2);

h1 = [1 -1] / sqrt(2);

h00 = [1 0 1] / sqrt(2);

h10 = [1 0 -1] / sqrt(2);

h000 = [1 0 0 0 1] / sqrt(2);

h100 = [1 0 0 0 -1] / sqrt(2);

H\_temp = conv(h0, h00);

H(1,:) = conv(H\_temp, h000); G(1,:) = H(1, :);

H(2,:) = conv(H\_temp, h100); G(2,:) = -H(2, :);

H\_temp = conv(h0, h10);

H(3,:) = conv(H\_temp, h000); G(3,:) = -H(3, :);

H(4,:) = conv(H\_temp, h100); G(4,:) = H(4, :);

H\_temp = conv(h1, h00);

H(5,:) = conv(H\_temp, h000); G(5,:) = -H(5, :);

H(6,:) = conv(H\_temp, h100); G(6,:) = H(6, :);

H\_temp = conv(h1, h10);

H(7,:) = conv(H\_temp, h000); G(7,:) = H(7, :);

H(8,:) = conv(H\_temp, h100); G(8,:) = -H(8, :);

% i) All corresponding DTFT's H\_m(w)

h\_m(1,:)=abs(fftshift(fft(H(1,:),512)));

h\_m(2,:)=abs(fftshift(fft(H(2,:),512)));

h\_m(3,:)=abs(fftshift(fft(H(3,:),512)));

h\_m(4,:)=abs(fftshift(fft(H(4,:),512)));

h\_m(5,:)=abs(fftshift(fft(H(5,:),512)));

h\_m(6,:)=abs(fftshift(fft(H(6,:),512)));

h\_m(7,:)=abs(fftshift(fft(H(7,:),512)));

h\_m(8,:)=abs(fftshift(fft(H(8,:),512)));

domega = 2 \* pi / 512;

omega = -pi:domega:pi-domega;

figure(1) %Figure 1(a)

plot(omega, h\_m(1,:), ...

omega, h\_m(2,:), ...

omega, h\_m(3,:), ...

omega, h\_m(4,:), ...

omega, h\_m(5,:), ...

omega, h\_m(6,:), ...

omega, h\_m(7,:), ...

omega, h\_m(8,:));

axis([-pi pi 0 3]);

title('Figura 1(a): All corresponding DTFT''s H\_m(\omega)')

ylabel('H\_m(\omega)');

xlabel('Omega, \omega (rad/sec)');

legend('H\_1(\omega)', 'H\_2(\omega)', ...

'H\_3(\omega)', 'H\_4(\omega)', ...

'H\_5(\omega)', 'H\_6(\omega)', ...

'H\_7(\omega)', 'H\_8(\omega)');

grid on

%ii) 8x8 matrix HH^H

table1 = H\*H';

table1 = round(table1)

%iii) DTFT of the Gaussian random process input signal

x = randn(1,128);

for m = 1:M

W(m,:) = conv(x,H(m,:));

X(m,:) = W(m,1:M:length(W(m,:)));

end

for m = 1:M

Z(m,:) = zeros(1,M\*length(X(m,:)));

Z(m,1:M:length(Z(m,:))) = X(m,:);

Y(m,:) = conv(Z(m,:),G(m,:));

end

y = zeros(1,length(Y(1,:)));

for m = 1:M

y = y+Y(m,:);

end

domega = 2\*pi/1024;

omega = -pi:domega:pi-domega;

yf1 = abs(fftshift(fft(x,1024)));

yf2 = abs(fftshift(fft(y,1024)));

figure(2) %Figure 1(b)

plot(omega, yf1)

axis([-pi pi 0 max(yf1)])

xlabel('Omega, \omega (rad/sec)');

ylabel('Magnitude of DTFT')

title('Figure 1(b): DTFT of Gaussian random process input signal')

grid on

%iv) DTFT of the corresponding output of the filter

figure(3) %Figure 1(c)

plot(omega, yf2, 'r')

axis([-pi pi 0 max(yf2)])

xlabel('Omega, \omega (rad/sec)');

ylabel('Magnitude of DTFT')

title('Figure 1(c): DTFT of the Gaussian random process output')

grid on

% B) h\_0 = h ,, h\_1 = (-1)^n \* h\_0 ,, beta = 0.35

N = 16;

beta = 0.35;

n = -N:(N-1);

n = n+0.5;

h = 2 \* beta \* cos((1+beta)\*pi\*n/2)./(pi\*(1-4\*beta^2\*n.^2));

h = h + sin((1-beta)\*pi\*n/2)./(pi\*(n-4\*beta^2\*n.^3));

h = h \* sqrt(2);

h0 = h;

h1 = (-1).^(0:(length(n)-1)).\*h;

h00 = zeros(1,2\*length(h));

h10 = h00;

h00(1,1:2:length(h00)) = h0;

h10(1,1:2:length(h10)) = h1;

h000 = zeros(1,4\*length(h));

h100 = h000;

h000(1,1:4:length(h000)) = h0;

h100(1,1:4:length(h100)) = h1;

H\_tempB = conv(h0, h00);

H\_B(1,:) = conv(H\_tempB, h000); G\_B(1,:) = H\_B(1, :);

H\_B(2,:) = conv(H\_tempB, h100); G\_B(2,:) = -H\_B(2, :);

H\_tempB = conv(h0, h10);

H\_B(3,:) = conv(H\_tempB, h000); G\_B(3,:) = -H\_B(3, :);

H\_B(4,:) = conv(H\_tempB, h100); G\_B(4,:) = H\_B(4, :);

H\_tempB = conv(h1, h00);

H\_B(5,:) = conv(H\_tempB, h000); G\_B(5,:) = -H\_B(5, :);

H\_B(6,:) = conv(H\_tempB, h100); G\_B(6,:) = H\_B(6, :);

H\_tempB = conv(h1, h10);

H\_B(7,:) = conv(H\_tempB, h000); G\_B(7,:) = H\_B(7, :);

H\_B(8,:) = conv(H\_tempB, h100); G\_B(8,:) = -H\_B(8, :);

% i) All corresponding DTFT's h\_mB(w)

h\_mB(1,:)=abs(fftshift(fft(H\_B(1,:),512)));

h\_mB(2,:)=abs(fftshift(fft(H\_B(2,:),512)));

h\_mB(3,:)=abs(fftshift(fft(H\_B(3,:),512)));

h\_mB(4,:)=abs(fftshift(fft(H\_B(4,:),512)));

h\_mB(5,:)=abs(fftshift(fft(H\_B(5,:),512)));

h\_mB(6,:)=abs(fftshift(fft(H\_B(6,:),512)));

h\_mB(7,:)=abs(fftshift(fft(H\_B(7,:),512)));

h\_mB(8,:)=abs(fftshift(fft(H\_B(8,:),512)));

domega\_B = 2 \* pi / 512;

omega\_B = -pi:domega\_B:pi-domega\_B;

figure(4) %Figure 2(a)

plot(omega\_B, h\_mB(1,:), ...

omega\_B, h\_mB(2,:), ...

omega\_B, h\_mB(3,:), ...

omega\_B, h\_mB(4,:), ...

omega\_B, h\_mB(5,:), ...

omega\_B, h\_mB(6,:), ...

omega\_B, h\_mB(7,:), ...

omega\_B, h\_mB(8,:));

axis([-pi pi 0 3]);

title('Figura 2(a): All corresponding DTFT''s h\_m(\omega)')

ylabel('h\_m(\omega)');

xlabel('omega, \omega (rad/sec)');

legend('H\_1(\omega)', 'H\_2(\omega)', ...

'H\_3(\omega)', 'H\_4(\omega)', ...

'H\_5(\omega)', 'H\_6(\omega)', ...

'H\_7(\omega)', 'H\_8(\omega)');

grid on

%ii) 8x8 matrix HH^H

table2 = H\_B\*H\_B'

%iii) DTFT of the Gaussian random process input signal

for m = 1:M

W\_B(m,:) = conv(x,H\_B(m,:));

X\_B(m,:) = W\_B(m,1:M:length(W\_B(m,:)));

end

for m = 1:M

Z\_B(m,:) = zeros(1,M\*length(X\_B(m,:)));

Z\_B(m,1:M:length(Z\_B(m,:))) = X\_B(m,:);

Y\_B(m,:) = conv(Z\_B(m,:),G\_B(m,:));

end

y\_B = zeros(1,length(Y\_B(1,:)));

for m = 1:M

y\_B = y\_B + Y\_B(m,:);

end

domega\_B = 2\*pi/1024;

omega\_B = -pi:domega\_B:pi-domega\_B;

yf1\_B = abs(fftshift(fft(x,1024)));

yf2\_B = M\*abs(fftshift(fft(y\_B,1024)));

figure(5);

plot(omega\_B,yf1\_B)

axis([-pi pi 0 max(yf1\_B)])

xlabel('Omega, \omega (rad/sec)');

ylabel('Magnitude of DTFT')

title('Figure 2(b): DTFT of Gaussian random process input signal')

grid on

%iv) DTFT of the corresponding output of the filter

figure(6);

plot(omega\_B,yf2\_B)

axis([-pi pi 0 max(yf2\_B)])

xlabel('Omega, \omega (rad/sec)');

ylabel('Magnitude of DTFT')

title('Figure 2(c): DTFT of the Gaussian random process output')

grid on

% C) h\_0 = h ,, h\_1 = (-1)^n \* h\_0 ,, beta = 0.1

N = 24;

beta = 0.1;

n = -N:(N-1);

n = n+0.5;

h = 2 \* beta \* cos((1+beta)\*pi\*n/2)./(pi\*(1-4\*beta^2\*n.^2));

h = h + sin((1-beta)\*pi\*n/2)./(pi\*(n-4\*beta^2\*n.^3));

h = h \* sqrt(2);

h0 = h;

h1 = (-1).^(0:(length(n)-1)).\*h;

h00 = zeros(1,2\*length(h));

h10 = h00;

h00(1,1:2:length(h00)) = h0;

h10(1,1:2:length(h10)) = h1;

h000 = zeros(1,4\*length(h));

h100 = h000;

h000(1,1:4:length(h000)) = h0;

h100(1,1:4:length(h100)) = h1;

H\_tempB = conv(h0, h00);

H\_C(1,:) = conv(H\_tempB, h000); G\_C(1,:) = H\_C(1, :);

H\_C(2,:) = conv(H\_tempB, h100); G\_C(2,:) = -H\_C(2, :);

H\_tempB = conv(h0, h10);

H\_C(3,:) = conv(H\_tempB, h000); G\_C(3,:) = -H\_C(3, :);

H\_C(4,:) = conv(H\_tempB, h100); G\_C(4,:) = H\_C(4, :);

H\_tempB = conv(h1, h00);

H\_C(5,:) = conv(H\_tempB, h000); G\_C(5,:) = -H\_C(5, :);

H\_C(6,:) = conv(H\_tempB, h100); G\_C(6,:) = H\_C(6, :);

H\_tempB = conv(h1, h10);

H\_C(7,:) = conv(H\_tempB, h000); G\_C(7,:) = H\_C(7, :);

H\_C(8,:) = conv(H\_tempB, h100); G\_C(8,:) = -H\_C(8, :);

% i) All corresponding DTFT's h\_mB(w)

h\_mB(1,:)=abs(fftshift(fft(H\_C(1,:),512)));

h\_mB(2,:)=abs(fftshift(fft(H\_C(2,:),512)));

h\_mB(3,:)=abs(fftshift(fft(H\_C(3,:),512)));

h\_mB(4,:)=abs(fftshift(fft(H\_C(4,:),512)));

h\_mB(5,:)=abs(fftshift(fft(H\_C(5,:),512)));

h\_mB(6,:)=abs(fftshift(fft(H\_C(6,:),512)));

h\_mB(7,:)=abs(fftshift(fft(H\_C(7,:),512)));

h\_mB(8,:)=abs(fftshift(fft(H\_C(8,:),512)));

domega\_C = 2 \* pi / 512;

omega\_C = -pi:domega\_C:pi-domega\_C;

figure(7) %Figure 3(a)

plot(omega\_C, h\_mB(1,:), ...

omega\_C, h\_mB(2,:), ...

omega\_C, h\_mB(3,:), ...

omega\_C, h\_mB(4,:), ...

omega\_C, h\_mB(5,:), ...

omega\_C, h\_mB(6,:), ...

omega\_C, h\_mB(7,:), ...

omega\_C, h\_mB(8,:));

axis([-pi pi 0 3]);

title('Figura 3(a): All corresponding DTFT''s h\_m(\omega)')

ylabel('h\_m(\omega)');

xlabel('omega, \omega (rad/sec)');

legend('H\_1(\omega)', 'H\_2(\omega)', ...

'H\_3(\omega)', 'H\_4(\omega)', ...

'H\_5(\omega)', 'H\_6(\omega)', ...

'H\_7(\omega)', 'H\_8(\omega)');

grid on

%ii) 8x8 matrix HH^H

table3 = H\_C\*H\_C'

%iii) DTFT of the Gaussian random process input signal

for m = 1:M

W\_C(m,:) = conv(x,H\_C(m,:));

X\_C(m,:) = W\_C(m,1:M:length(W\_C(m,:)));

end

for m = 1:M

Z\_C(m,:) = zeros(1,M\*length(X\_C(m,:)));

Z\_C(m,1:M:length(Z\_C(m,:))) = X\_C(m,:);

Y\_C(m,:) = conv(Z\_C(m,:),G\_C(m,:));

end

y\_C = zeros(1,length(Y\_C(1,:)));

for m = 1:M

y\_C = y\_C + Y\_C(m,:);

end

domega\_C = 2\*pi/1024;

omega\_C = -pi:domega\_C:pi-domega\_C;

yf1\_C = abs(fftshift(fft(x,1024)));

yf2\_C = M\*abs(fftshift(fft(y\_C,1024)));

figure(8);

plot(omega\_C,yf1\_C)

axis([-pi pi 0 max(yf1\_C)])

xlabel('Omega, \omega (rad/sec)');

ylabel('Magnitude of DTFT')

title('Figure 3(b): DTFT of Gaussian random process input signal')

grid on

%iv) DTFT of the corresponding output of the filter

figure(9);

plot(omega\_C,yf2\_C)

axis([-pi pi 0 max(yf2\_C)])

xlabel('Omega, \omega (rad/sec)');

ylabel('Magnitude of DTFT')

title('Figure 3(c): DTFT of the Gaussian random process output')

grid on