## 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
```