Faculty of Engineering
Computer Department
Communications (ELC 325B) – Spring 2023

Assignment 3

Team Members

Num	Full Name in ARABIC	SEC	BN
1	بيتر عاطف فتحي	1	19
2	بيشوي مراد عطية	1	20

Table of contents:

List of Figures

1. Part One

1.1 Gram-Schmidt Orthogonalization

$$E_{s1} = \int_{-\infty}^{\infty} (S_1(t))^2 \, dt = S_1 \,.\, S_1 = 1$$

$$\emptyset_1(t) = \frac{S_1(t)}{\sqrt{E_{s1}}} = S_1(t)$$

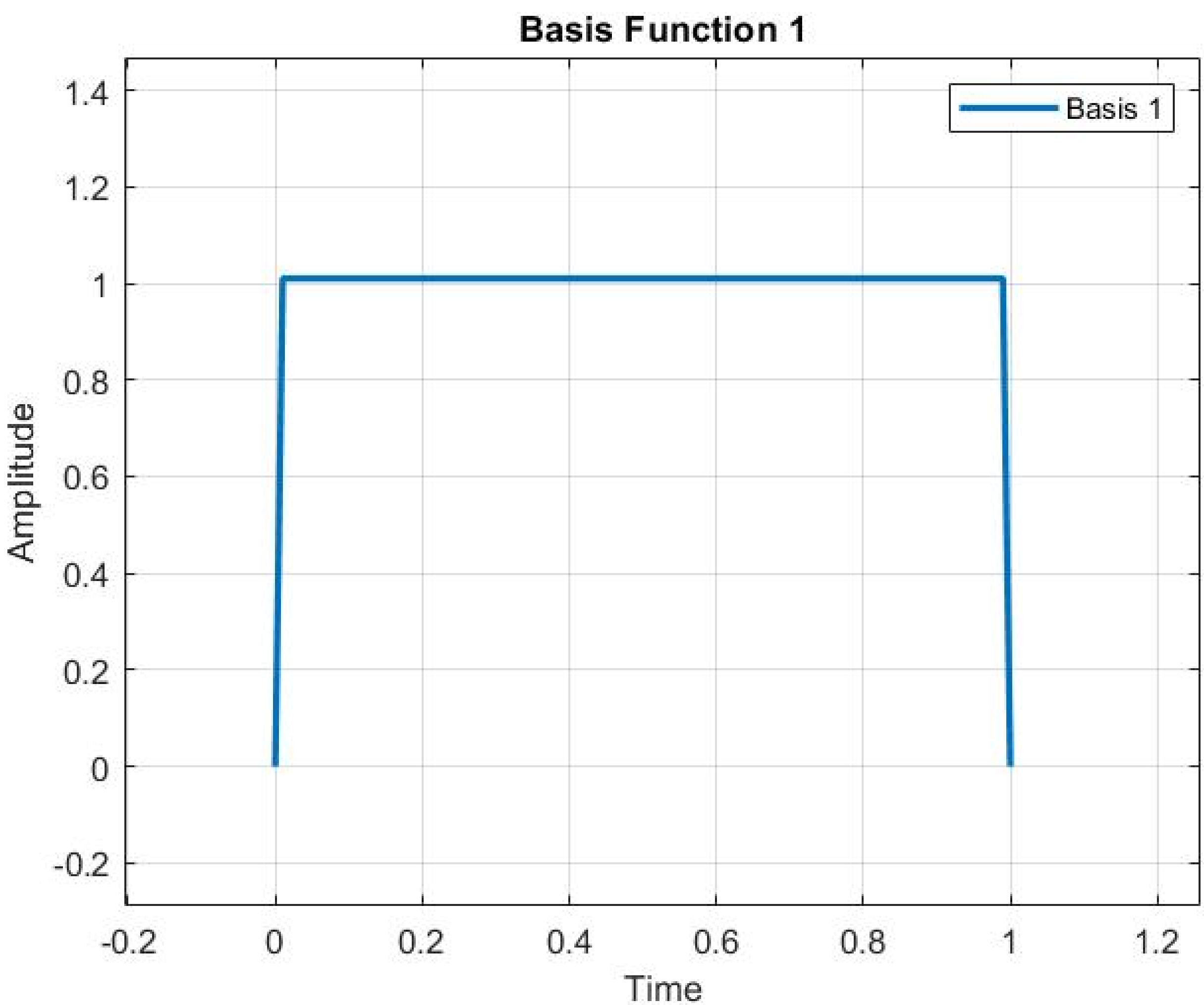


Figure 1 Φ1 VS time after using the GM_Bases function

$$s_{21} = \int_{-\infty}^{\infty} S_2(t) * \emptyset_1(t)\, dt = \int_{0}^{0.75} 1 * 1\, dt + \int_{0.75}^{1} -1 * 1\, dt = 0.75 - 0.25 = 0.5$$

$$g_2(t) = S_2(t) - s_{21} * \emptyset_1(t)$$

$$E_{g_2} = \int_{-\infty}^{\infty} \left(g_2(t)\right)^2 dt = \int_{0}^{0.75} \left(g_2(t)\right)^2 dt + \int_{0.75}^{1} \left(g_2(t)\right)^2 dt$$

$$= 0.75 * 0.25 + 0.25 * 1.5^2 = 0.75$$

$$\emptyset_2(t) = \frac{g_2(t)}{\sqrt{E_{g_2}}} = \frac{g_2(t)}{\sqrt{0.75}} = \frac{2}{\sqrt{3}} g_2(t)$$

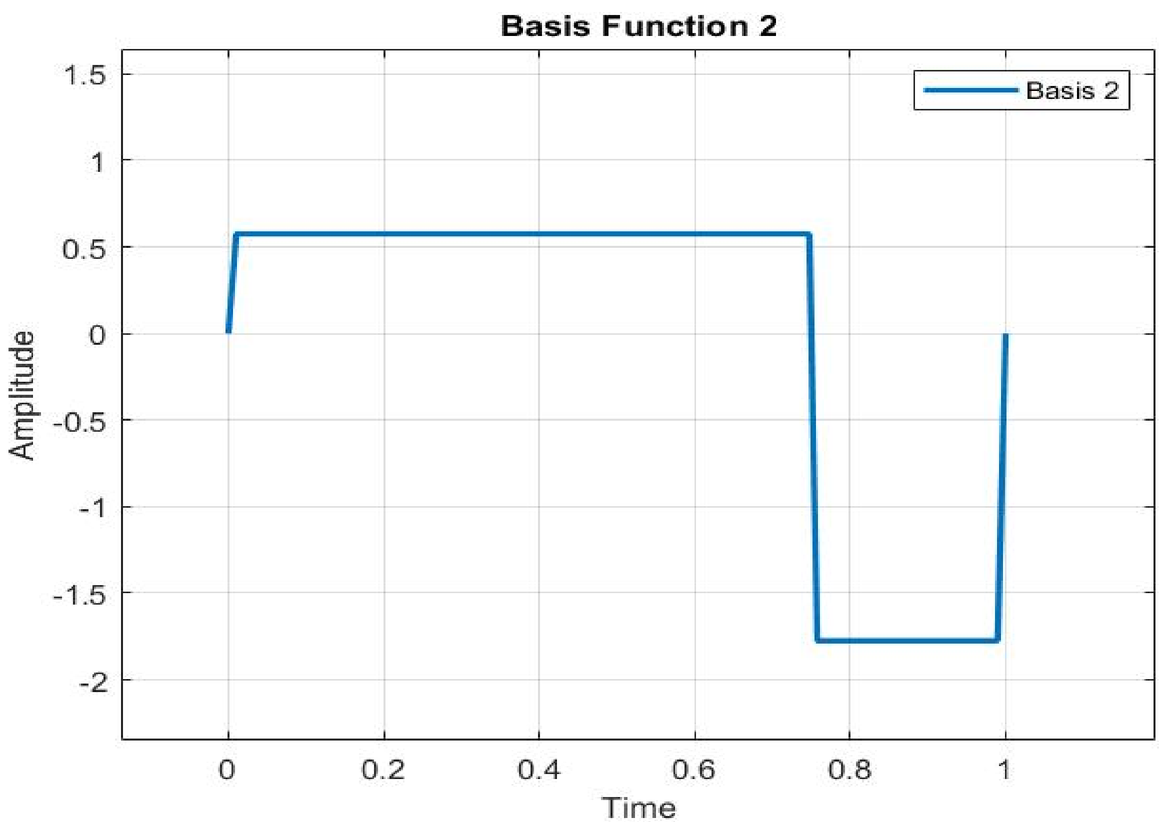


Figure 2 Φ2 VS time after using the GM_Bases function.

Solution verification:

$$\int_{-\infty}^{\infty} (\emptyset_1(t))^2\, dt = 1$$

$$\int_{-\infty}^{\infty} (\emptyset_2(t))^2\, dt = \int_{0}^{0.75} \left(\frac{\sqrt{3}}{3}\right)^2\, dt + \int_{0.75}^{1} \left(\sqrt{3}\right)^2\, dt = 1$$

$$\int_{-\infty}^{\infty} \emptyset_1(t) * \emptyset_2(t)\, dt = \int_{0}^{0.75} \frac{\sqrt{3}}{3} * 1\, dt + \int_{0.75}^{1} \sqrt{3} * -1\, dt = 0$$

So, the bases are orthogonal.

1.2 Signal Space Representation

Here we represent the signals using the base functions. “Without any noise”

$$S_1 = \binom{1}{0}, S_2 = \begin{pmatrix} 0.5 \\ \frac{\sqrt{3}}{2} \end{pmatrix},$$

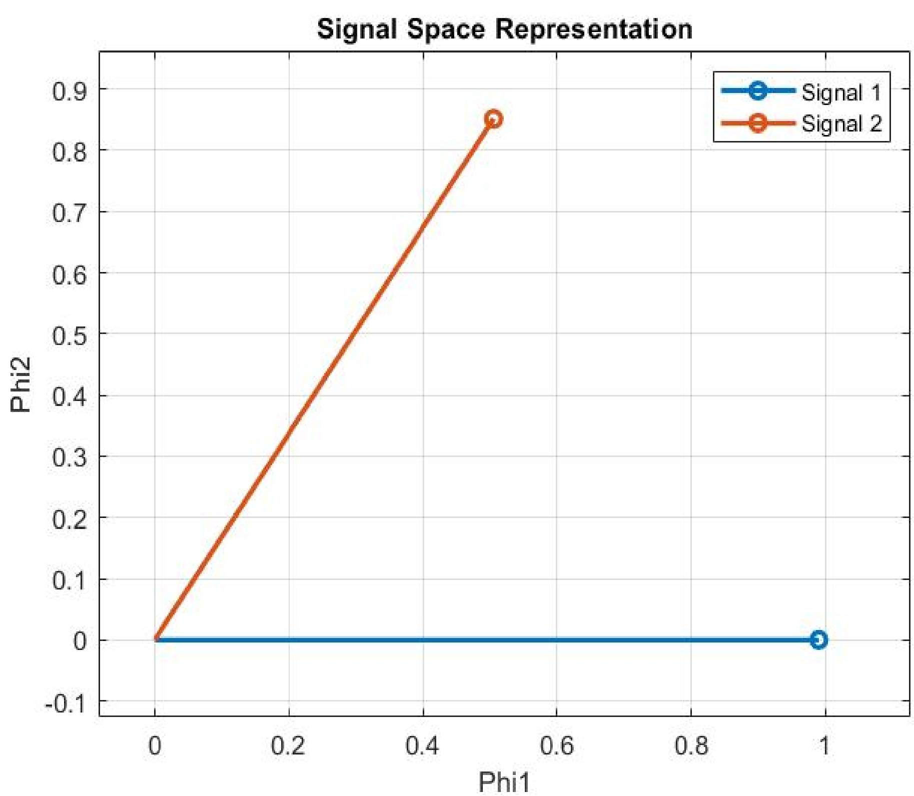


Figure 3 Signal Space representation of signals s1, s2

1.3 Signal Space Representation with adding AWGN.

-the expected real points will be solid and the received will be hollow

Case 1: $10\,log(E/\sigma^2)\ =\ 10\,dB$

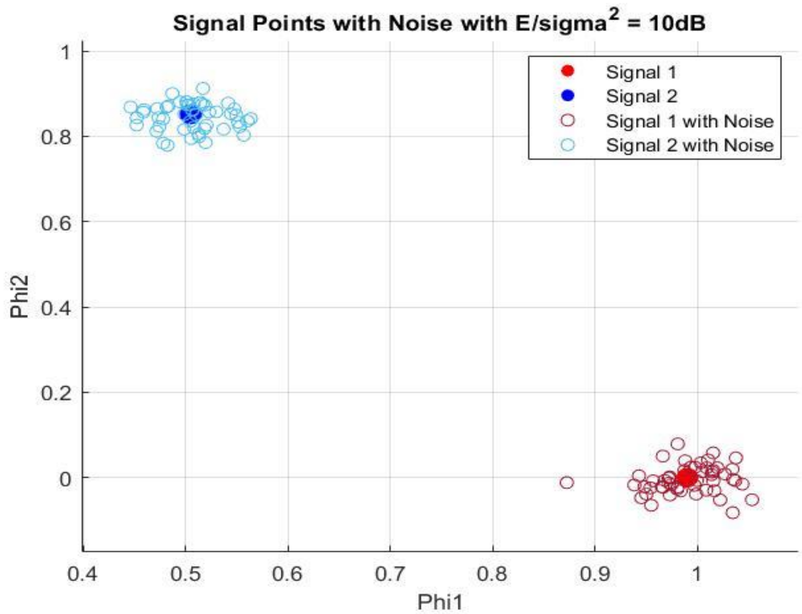
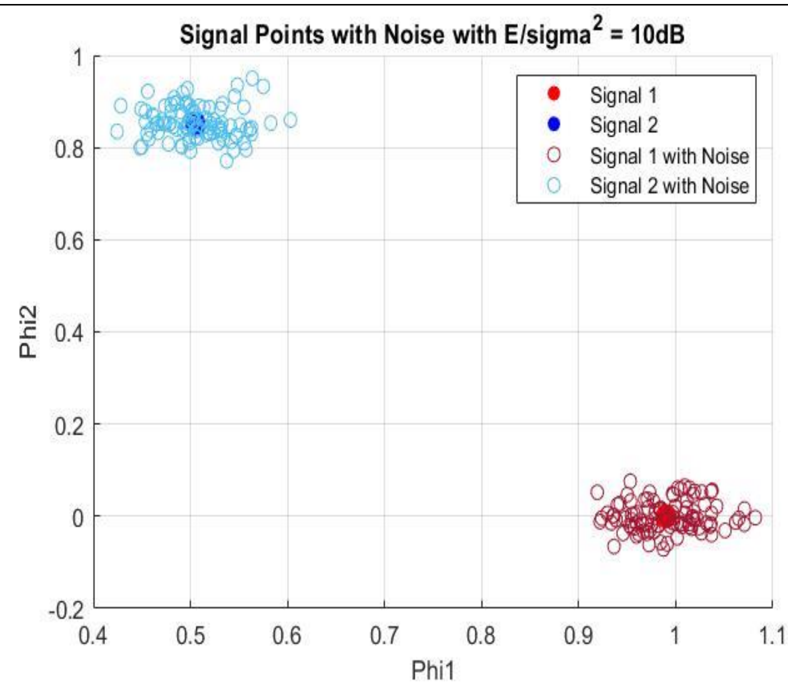
	50 sample of signal#1 and signal#2 with noise
	100 sample of signal#1 and signal#2 with noise

Figure 4 Signal Space representation of signals s1, s2 with E/σ¬2 =10dB

Case 2: $\mathbf{10}\ \boldsymbol{log}(\boldsymbol{E}/\boldsymbol{\sigma}^{\mathbf{2}})\ =\ \mathbf{0}\ \boldsymbol{dB}$

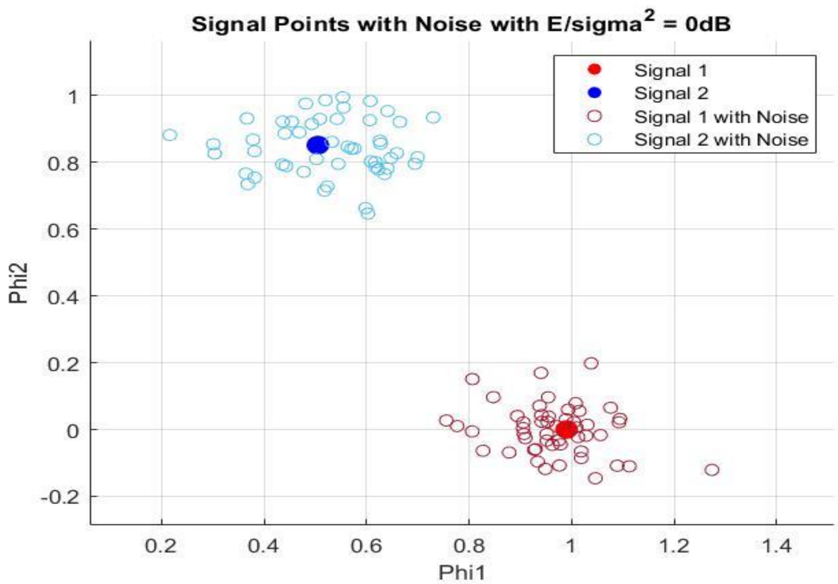
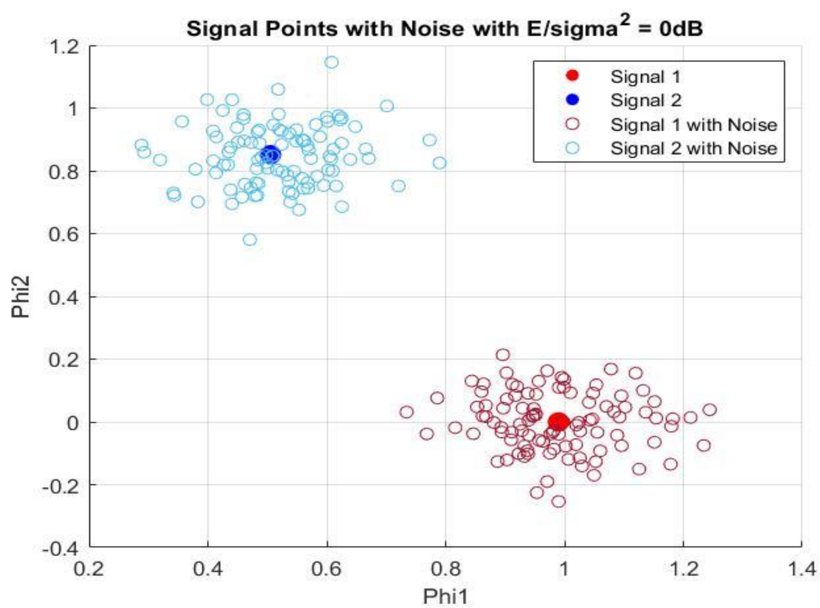
	50 sample of signal#1 and signal#2 with noise
	100 sample of signal#1 and signal#2 with noise

Figure 5 Signal Space representation of signals s1,s2 with E/σ¬2 =0dB

Case 3: $\mathbf{10}\ \boldsymbol{log(E/\sigma^2)\ =\ -5\ dB}$

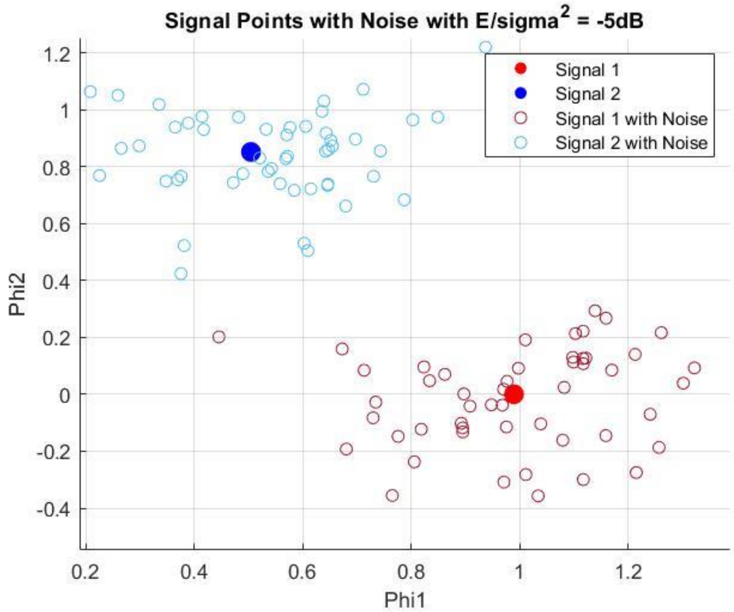
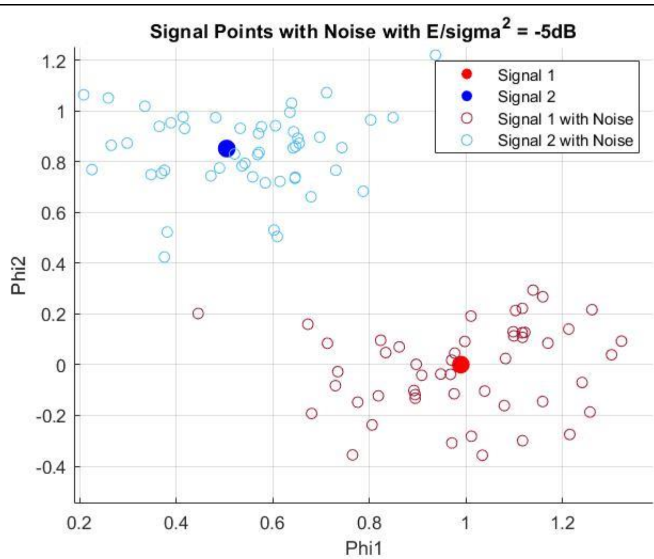
	50 sample of signal#1 and signal#2 with noise
	100 sample of signal#1 and signal#2 with noise

Figure 6 Signal Space representation of signals s1, s2 with E/σ¬2 =-5dB

1.4 Noise Effect on Signal Space

How does the noise affect the signal space?

Receiver will get $X(t) = S_i(t) + w(t)$

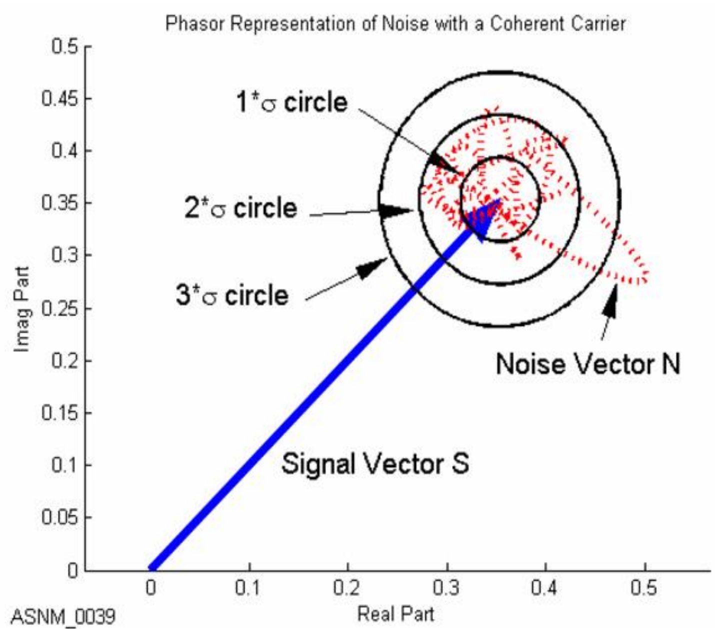
Then we find $y_j = \int_0^T X(t) * \emptyset_j(t) dt$, $for\ 1 \le j \le M$

$$y_{ij} = \int_0^T S_i(t) * \emptyset_j(t)\, dt + \int_0^T w(t) * \emptyset_j(t)\, dt$$

The receiver will get a signal y_{ij} whose space vector near to the space vector of S_{ij} “the space vector of the sent pulse.”, and we detect the actual signal space using gaussian probability.

Does the noise effect increase or decrease with increasing σ^2?

It’s noticed that with the increase of the variance of the noise, the signal to noise ratio decreases then the probability of error increases “the uncertainty increases” as observed from the following image.



References

1. Matlab documentations for rectangularPulse function
2. Wikipedia

Appendix A: Codes for Part One:

A.1 Code for Gram-Schmidt Orthogonalization

```
function [phi1, phi2] = GM_Bases(s1, s2)
    % Eenergy of s1 = dot(s1, s1)
    phi1 = s1 / sqrt(dot(s1, s1));
    % s21 = dot(s2, phi1) which is the projection of s2 onto phi1
    v2 = s2 - dot(s2, phi1) * phi1;
    phi2 = v2 / sqrt(dot(v2, v2));
    % to normalize the bases
    phi1 = phi1 * sqrt(length(s1));
    phi2 = phi2 * sqrt(length(s2));
end
```

A.2 Code for Signal Space representation

```
function [v1, v2] = signal_space(s, phi1, phi2)
    v1 = dot(s, phi1) / length(s); %v1 = Si1 = dot<s, phi1> / ||s||
    v2 = dot(s, phi2) / length(s); %v2 = Si2 = dot<s, phi2> / ||s||
end
```

A.3 Code for plotting the bases functions

```
% ================================================================
% Construct the signals
t = linspace(0, 1, 100);
s1 = rectangularPulse(0, 1, t);
s1(1) = 0; s1(end) = 0;

s2 = rectangularPulse(0, 0.75, t) - 1 * rectangularPulse(0.75, 1, t);
s2(1) = 0; s2(end) = 0;
% ================================================================

% ================================================================
% REQUIREMENTS 1:
[phi1, phi2] = GM_Bases(s1, s2);

% Plot the signals
figure('Name', 'Basis Functions', 'NumberTitle', 'off');
plot(t, phi1, 'LineWidth', 2);
legend('Basis 1');
xlabel('Time');
ylabel('Amplitude');
title('Basis Function 1');
grid on;
```

```
figure('Name', 'Basis Functions', 'NumberTitle', 'off');
plot(t, phi2, 'LineWidth', 2);
legend('Basis 2');
xlabel('Time');
ylabel('Amplitude');
title('Basis Function 2');
grid on;
```

A.4 Code for plotting the Signal space Representations

```
[s1_v1, s1_v2] = signal_space(s1, phi1, phi2);
[s2_v1, s2_v2] = signal_space(s2, phi1, phi2);

% Plot the signal
figure('Name', 'Signal Space Representation', 'NumberTitle', 'off');

plot([0 s1_v1], [0 s1_v2], '-o', 'MarkerIndices', [2 2], 'LineWidth', 2);
hold on;
plot([0 s2_v1], [0 s2_v2], '-o', 'MarkerIndices', [2 2], 'LineWidth', 2);

legend('Signal 1', 'Signal 2');
xlabel('Phi1');
ylabel('Phi2');
title('Signal Space Representation');
grid on;
```

A.5 Code for effect of noise on the Signal space Representations

```
% signal_space_with_noise.m
function r = signal_space_with_noise(s, sigma)
    % Generate AWGN
    noise = sigma * randn(1, length(s));
    % Add noise to the signals
    r = s + noise;
end
% ==========================================================
% plot_signal_with_noise.m
function plot_signal_with_noise(testCase, s1_v1, s1_v2, s2_v1, s2_v2, s1, s2, phi1, phi2)
    % Draw the signal space representation of the signals before adding noise
    figure('Name', 'Signal Points with Noise', 'NumberTitle', 'off');
    scatter(s1_v1, s1_v2, 100, 'r', 'filled');
    hold on;
    scatter(s2_v1, s2_v2, 100, 'b', 'filled');

    % E / sigma^2 list in dB
    EoSigma = [-5, 0, 10];
```

```
    Es1 = dot(s1, s1) / length(s1);
    Es2 = dot(s2, s2) / length(s2);
    disp(Es1);
    disp(Es2);
    sigma1 = sqrt(Es1 ./ db2mag(EoSigma));
    sigma2 = sqrt(Es2 ./ db2mag(EoSigma));

    for i = 1 : 100
        r1 = signal_space_with_noise(s1, sigma1(testCase));
        r2 = signal_space_with_noise(s2, sigma2(testCase));

        % Calculate signal space representation of the generated samples
        [r1_v1, r1_v2] = signal_space(r1, phi1, phi2);
        [r2_v1, r2_v2] = signal_space(r2, phi1, phi2);

        % Draw the signal space representation of the signals after adding noise
        scatter(r1_v1, r1_v2, [], [0.6350 0.0780 0.1840]);
        scatter(r2_v1, r2_v2, [], [0.3010 0.7450 0.9330]);
    end

    legend("Signal 1", "Signal 2", "Signal 1 with Noise", "Signal 2 with Noise");
    xlabel('Phi1');
    ylabel('Phi2');
    title('Signal Points with Noise with E/sigma^2 = ' + string(EoSigma(testCase)) + 'dB');
    grid on;
end
% ================================================================
% Function calls to plot the signals with noise in main.m
plot_signal_with_noise(1, s1_v1, s1_v2, s2_v1, s2_v2, s1, s2, phi1, phi2);
plot_signal_with_noise(2, s1_v1, s1_v2, s2_v1, s2_v2, s1, s2, phi1, phi2);
plot_signal_with_noise(3, s1_v1, s1_v2, s2_v1, s2_v2, s1, s2, phi1, phi2);
% ================================================================
```