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

Assignment 3

Team Members

Num	Full Name in ARABIC	SEC	BN
1	عبد الرحمن محمد حمزه عثمان	1	37
2	زياد شريف محمد ابراهيم حسين	1	27
3	محمود عبدالحميد على مصطفى	2	22

Table of contents:

List of Figures

1. Part One

1.1

The function performs **the Gram-Schmidt orthogonalization** process to find two orthogonal basis functions that span the same subspace as the input signals, ensuring that **phi1** and **phi2** are orthogonal to each other and capture the essential characteristics of **s1** and **s2.**

Figure 1 Φ1 VS time after using the GM_Bases function

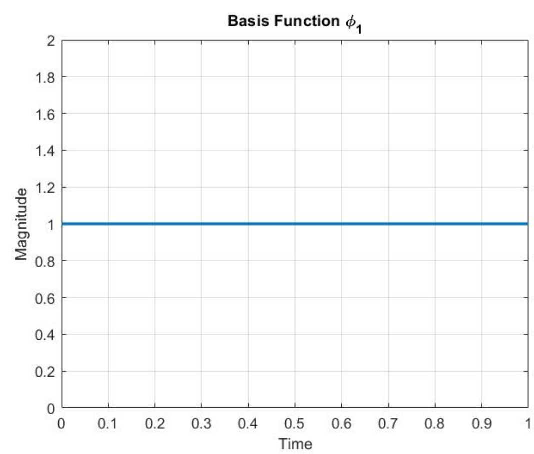
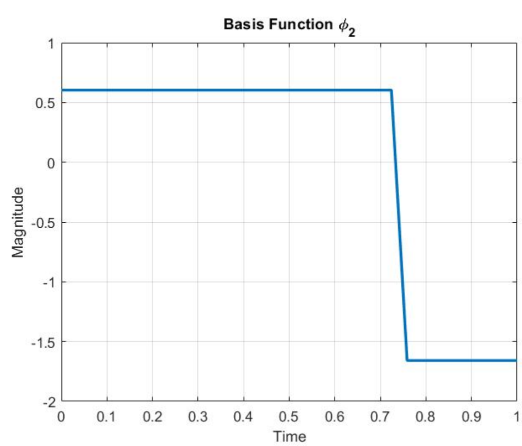


Figure 2 Φ2 VS time after using the GM_Bases function

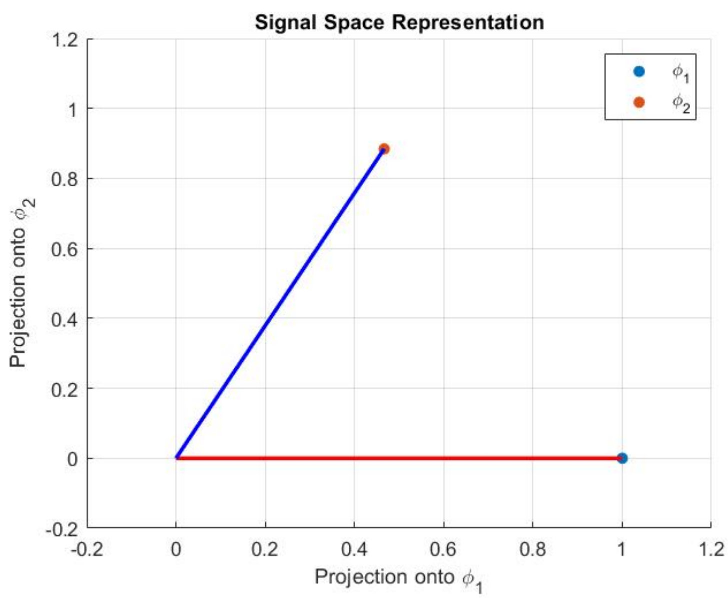


1.2 Signal Space Representation

Here we represent the signals using the base functions.

The signal_space function measures **the correlations** between the input signal s and the basis vectors **phi1** and **phi2**. It provides a way to represent the signal s in terms of the contributions from the two basis vectors.

Figure 3 Signal Space representation of signals s1,s2

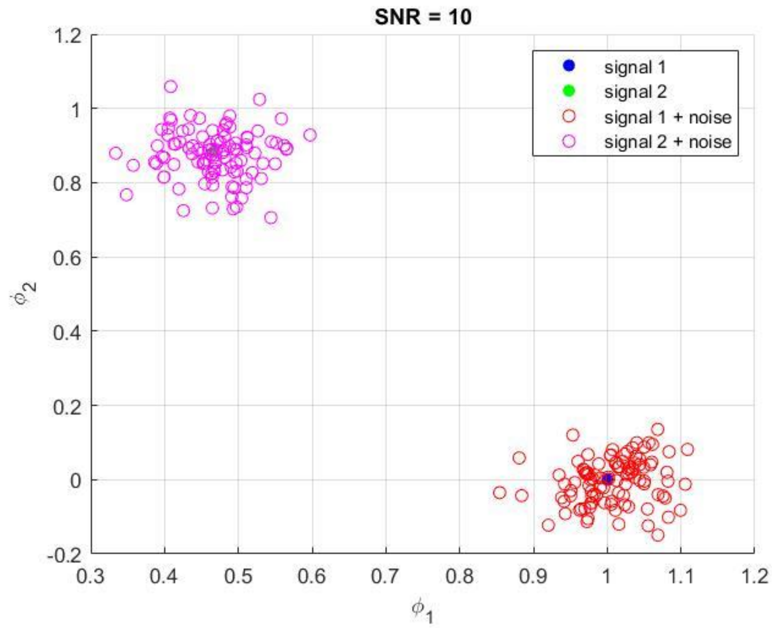


1.3 Signal Space Representation with adding AWGN

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

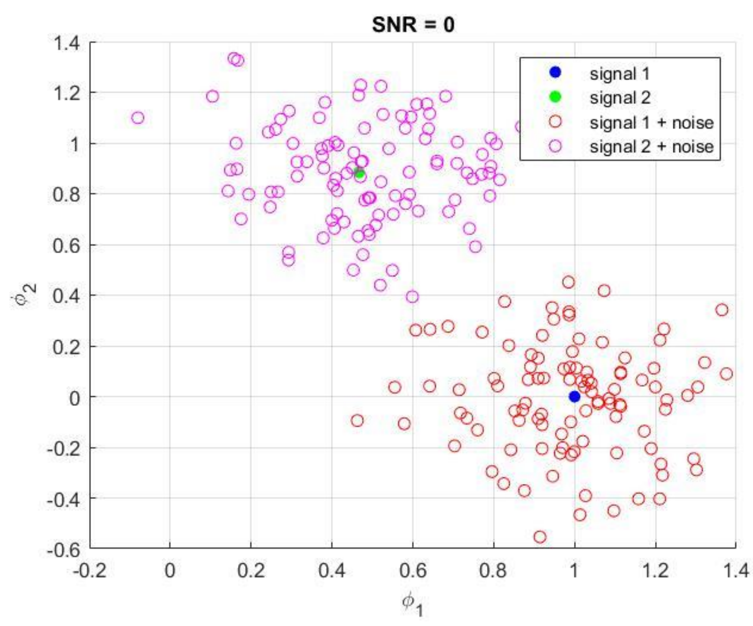
Case 1: $\mathbf{10\ log(E/\sigma^2)\ =\ 10\ dB}$

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



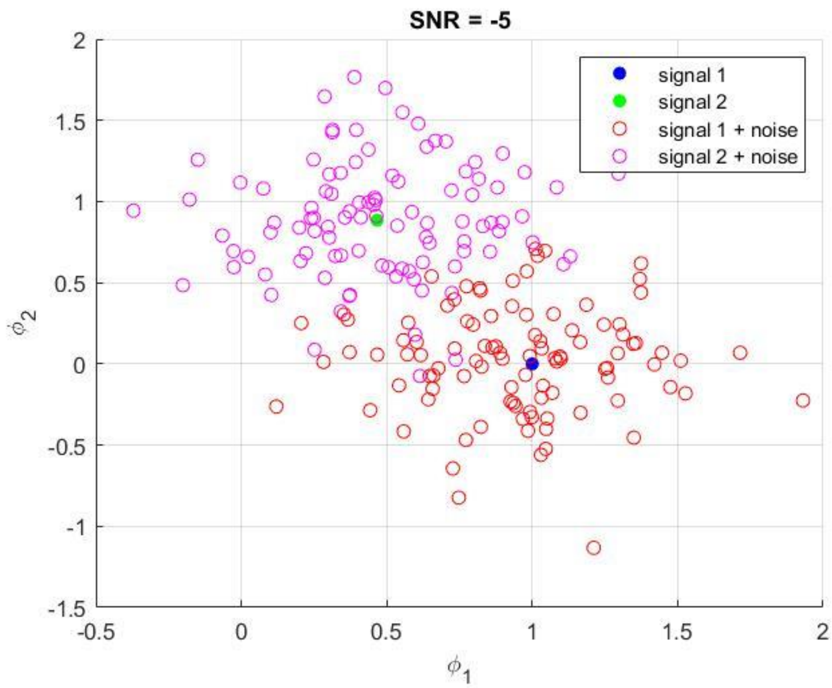
Case 2: $\mathbf{10}\ \boldsymbol{log(E/\sigma^2)\ =\ 0\ dB}$

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



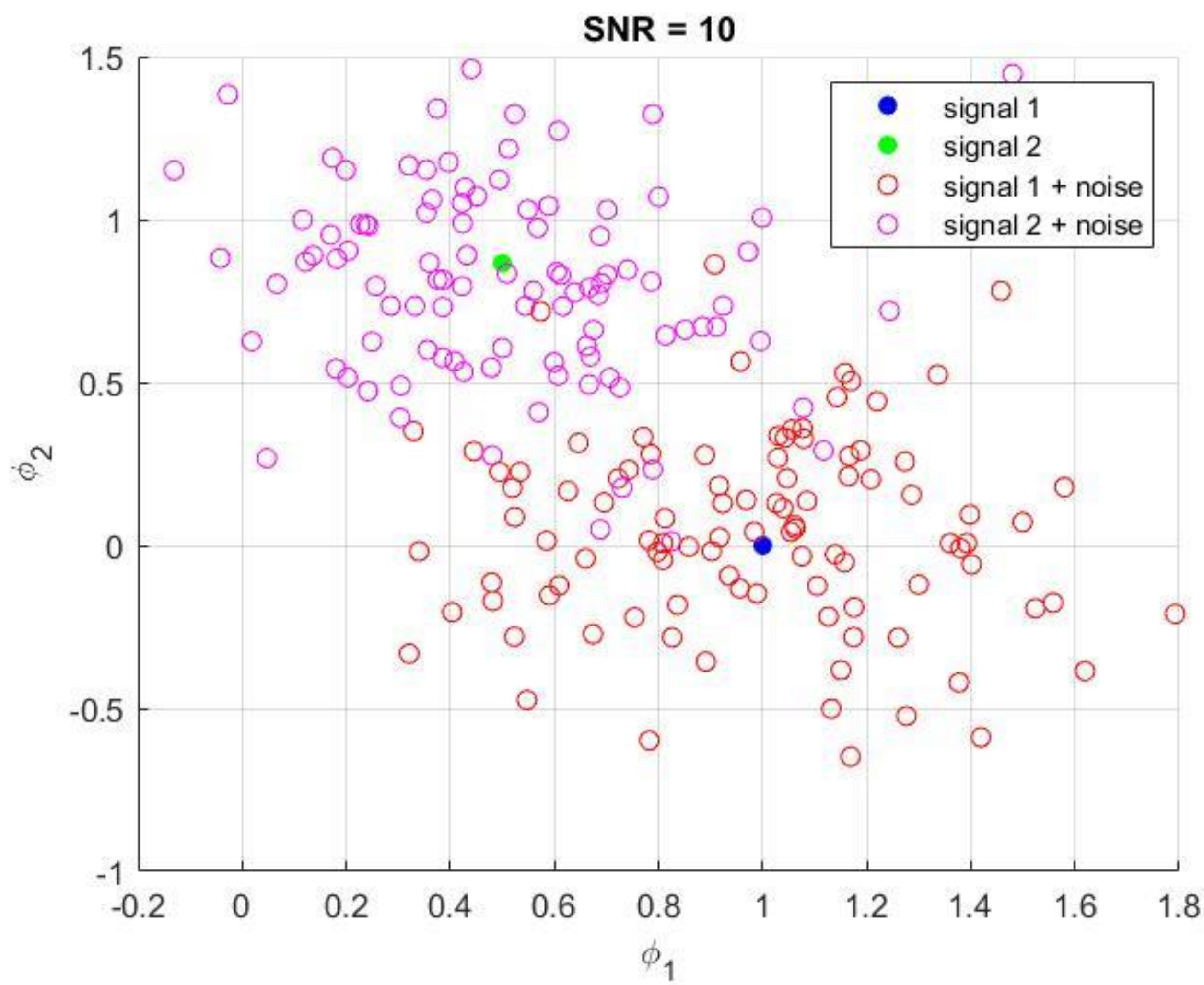
Case 3: $\mathbf{10}\, \boldsymbol{log}(\boldsymbol{E}/\boldsymbol{\sigma}^{\mathbf{2}}) \;=\; \mathbf{-5}\, \boldsymbol{dB}$

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

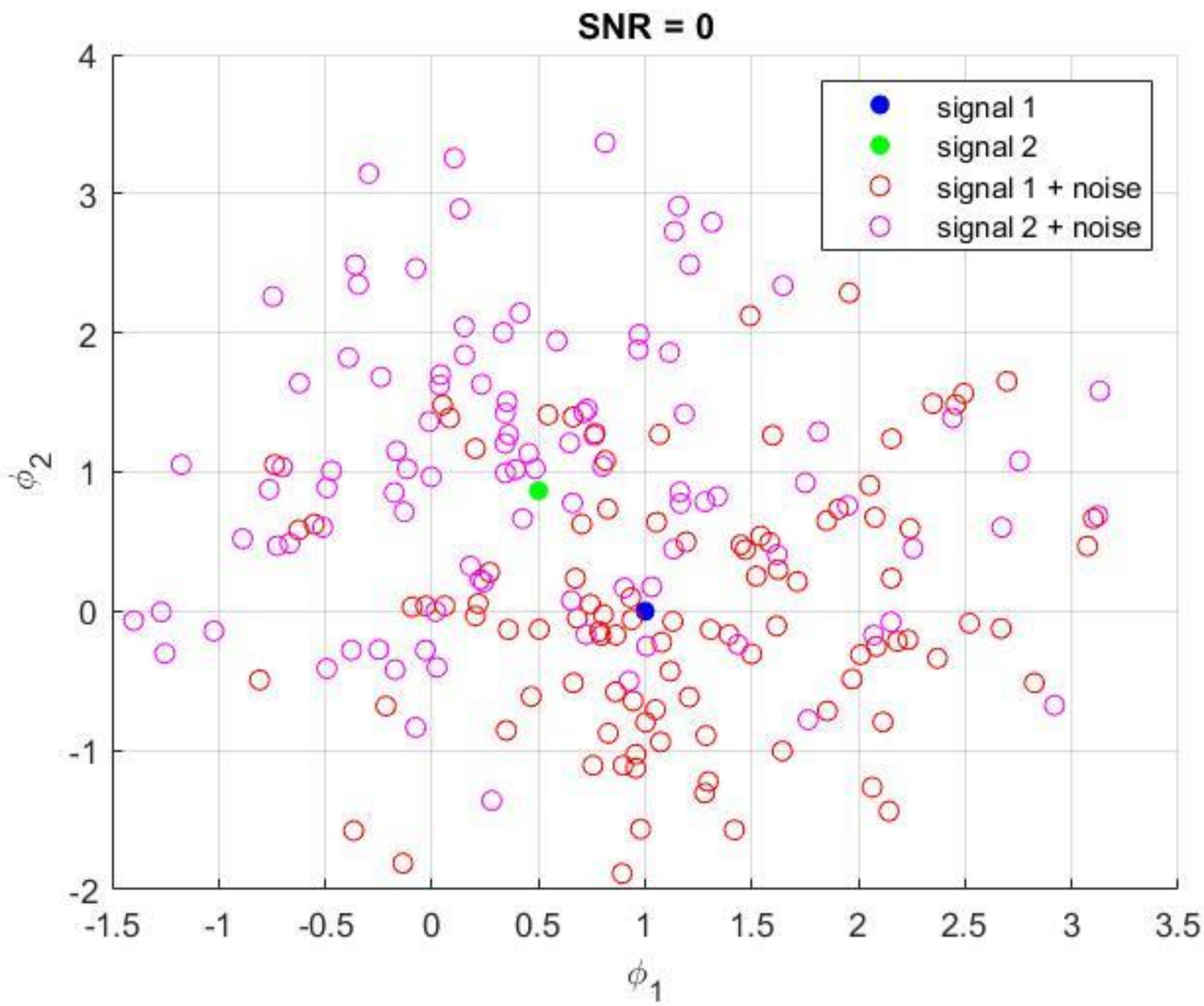


Note by taking energy into account we get this: -

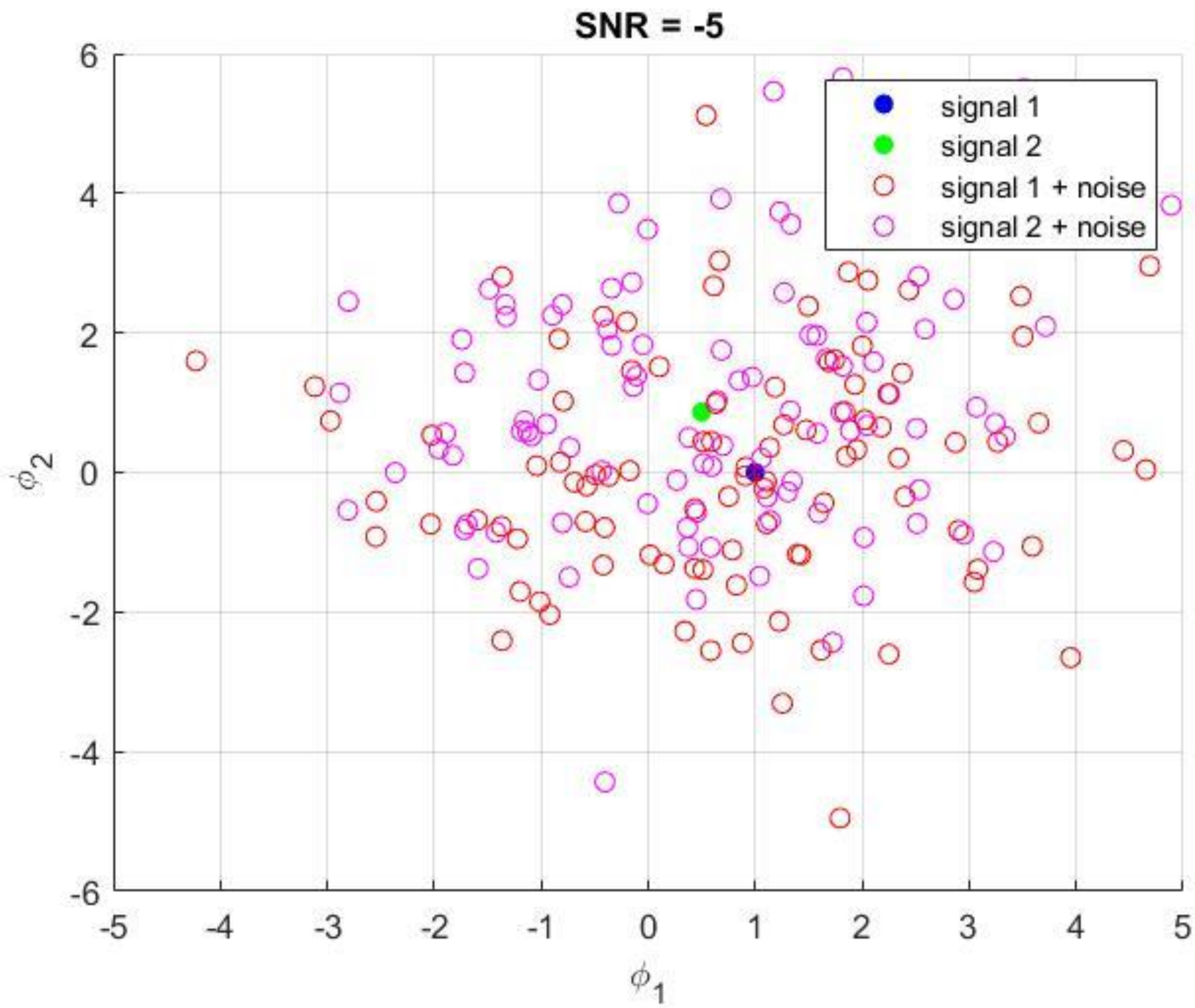
Case 1: $\mathbf{10}\, \boldsymbol{log}(\boldsymbol{E}/\boldsymbol{\sigma}^{\mathbf{2}}) \;=\; \mathbf{10}\, \boldsymbol{dB}$


SNR = 10
signal 1
signal 2
signal 1 + noise
signal 2 + noise
ϕ_1
ϕ_2

Case 1: $\mathbf{10}\ \boldsymbol{log(E/\sigma^2)} \ = \ \mathbf{0}\ \boldsymbol{dB}$



Case 1: $\mathbf{10}\ \boldsymbol{log\left(\frac{E}{\sigma^2}\right)} \ = \ \mathbf{-5}\ \boldsymbol{dB}$



1.4 Noise Effect on Signal Space

2. As the noise variance σ^2 increases, the effect of noise on the signal space becomes more noticeable. **This results in the signal points being spread out more widely and becoming less concentrated around their ideal positions.** The distinction between different signal points becomes less clear, making it harder to distinguish between signals or determine their relative positions in the signal space. In summary**, increasing σ^2 amplifies the impact of noise on the signal space representation.**

Appendix A: Codes for Part One:

A.1 Code for Gram-Schmidt Orthogonalization

```
function [phi1, phi2] = GM_Bases(s1, s2)
    % Check if the input signals have the same length
    if length(s1) ~= length(s2)
        error('Input signals must have the same length');
    end

    N = length(s1); % Length of the input signals

    % Initialize the basis functions
    phi1 = zeros(1, N);
    phi2 = zeros(1, N);

    % Calculate the first basis function (phi1)
    phi1 = s1 / (norm(s1/(N.^0.5)));

    % Calculate the projection of s2 onto phi1
    proj = dot(s2, phi1) * phi1/N;

    % Calculate the second basis function (phi2)
    phi2 = s2 - proj;
    phi2 = phi2 / norm(phi2/(N.^0.5));

    % Set phi2 to a zero vector if s1 and s2 have one basis function
    if norm(s2 - proj) == 0
        phi2 = zeros(1, N);
    end
end
```

A.2 Code for Signal Space representation

```
function [v1, v2] = signal_space(s, phi1, phi2)
    % Check if the input vectors have the same length
    if length(s) ~= length(phi1) || length(s) ~= length(phi2)
        error('Input vectors must have the same length');
    end

    % Calculate the projections (correlations) of s over phi1 and phi2
    v1 = dot(s, phi1) / length(phi1);
    v2 = dot(s, phi2) / length(phi2);
end
```

A.3 Code for plotting the bases functions

```
%Use your �GM_Bases� function to get the bases functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Obtain the basis functions using GM_Bases function
[phi1, phi2] = GM_Bases(s1, s2);

% Plot the obtained basis functions for s1
figure;
plot(t, phi1, 'LineWidth', 2);
xlabel('Time');
ylabel('Magnitude');
title('Basis Function \phi_1');
grid on;

figure;
plot(t, phi2, 'LineWidth', 2);
xlabel('Time');
ylabel('Magnitude');
title('Basis Function \phi_2');
grid on;
```

A.4 Code for plotting the Signal space Representations

```
% Calculate the signal space representation of s1
[v1_s1, v2_s1] = signal_space(s1, phi1, phi2);

% Calculate the signal space representation of s2
[v1_s2, v2_s2] = signal_space(s2, phi1, phi2);

% Plot the signal space representation for s1 and s2 as scatter plot
figure;
scatter(v1_s1, v2_s1, 'filled');
hold on;
scatter(v1_s2, v2_s2, 'filled');
hold on;
% Plot a line connecting the origin to the specified point
plot([0, v1_s1], [0, v2_s1], 'r','LineWidth', 2);
hold on;
% Plot a line connecting the origin to the specified point
plot([0, v1_s2], [0, v2_s2], 'b','LineWidth', 2);
xlabel('Projection onto \phi_1');
ylabel('Projection onto \phi_2');
xlim([-0.2,1.2]);
ylim([-0.2,1.2]);
title('Signal Space Representation');
legend('\phi_1', '\phi_2');
grid on;
```

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

```
% Effect of AWGN on signal space representation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Define the SNR levels (in dB)
SNR_levels = [-5, 0, 10];
% Generate samples of r1(t) and r2(t) for each SNR level
for i = 1:length(SNR_levels)
    figure;
    grid on;
    hold on;
    % Plot the signal points
    scatter(v1_s1, v2_s1,'b', 'filled');
    scatter(v1_s2, v2_s2,'g', 'filled');

    for j = 1:100
        % Calculate the noise variance based on the SNR level
        SNR_dB = SNR_levels(i);

        % Generate samples of r1(t) and r2(t) using awgn
        r1 = awgn(s1, SNR_dB, 'measured');
        r2 = awgn(s2, SNR_dB, 'measured');

        % Calculate the signal space representation of r1(t) and r2(t)
        [v1_r1, v2_r1] = signal_space(r1, phi1, phi2);
        [v1_r2, v2_r2] = signal_space(r2, phi1, phi2);

        % Plot the signal points
        scatter(v1_r1, v2_r1,'r');
        scatter(v1_r2, v2_r2,'m');
    end
    xlabel('\phi_1');
    ylabel('\phi_2');
    title(['SNR = ' , num2str(SNR_dB)]);

    % Add legends
    legend('signal 1', 'signal 2','signal 1 + noise','signal 2 + noise');
end
```

Note by taking energy into account we get this: -

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Effect of AWGN on signal space representation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Define the SNR levels (in dB)
SNR_levels = [-5, 0, 10];

signal_power = mean(s1.^2);
SNR_linear = 10.^(SNR_levels/10);

disp(SNR_value);

% Generate samples of r1(t) and r2(t) for each SNR level
for i = 1:length(SNR_levels)
    figure;
    grid on;
    hold on;
    % Plot the signal points
    scatter(v1_s1, v2_s1,'b', 'filled');
    scatter(v1_s2, v2_s2,'g', 'filled');

    for j = 1:100
        % Calculate the noise variance based on the SNR level
        SNR_dB = SNR_levels(i);

        % Generate samples of r1(t) and r2(t) using awgn
        r1 = s1 + sqrt(sum(s1.^2) / SNR_linear(i)) * randn(size(s1));
        r2 = s2 + sqrt(sum(s2.^2) / SNR_linear(i)) * randn(size(s2));

        % Calculate the signal space representation of r1(t) and
r2(t)
        [v1_r1, v2_r1] = signal_space(r1, phi1, phi2);
        [v1_r2, v2_r2] = signal_space(r2, phi1, phi2);

        % Plot the signal points
        scatter(v1_r1, v2_r1,'r');
        scatter(v1_r2, v2_r2,'m');
```

```
    end
    xlabel('\phi_1');
    ylabel('\phi_2');
    title(['SNR = ' , num2str(SNR_dB)]);

    % Add legends
    legend('signal 1', 'signal 2','signal 1 + noise','signal 2 +
noise');
end
```