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

Assignment #3

Submitted to:

Dr. Mai

Dr. Hala

Eng. Mohamed Khaled

Team Members

Num	Full Name in ARABIC	SEC	BN
1	أحمد حسني عبد الرازق عبد الغني	1	2
1	إسلام أشرف ابراهيم	1	13
2	نور زياد الملحم	2	31

Table of contents:

List of Figures

1. Part One

1.1 Gram-Schmidt Orthogonalization

Gram-Schmidt orthogonalization is a process of transforming a set of linearly independent vectors into an orthonormal basis for the same subspace.
The Algorithm used:

```
for j=1:n                        % Gram-Schmidt orthogonalization
  v=A(:,j);                      % v begins as column j of A
  for i=1:j-1
    R(i,j)=Q(:,i)'*A(:,j);       % modify A(:,j) to v for more accuracy
    v=v-R(i,j)*Q(:,i);           % subtract the projection (q_i^T a_j)q_i = (q_i^T v)q_i
  end                            % v is now perpendicular to all of q_1, ..., q_{j-1}
  R(j,j)=norm(v);
  Q(:,j)=v/R(j,j);               % normalize v to be the next unit vector q_j
end
```

<u>Phi1:</u>

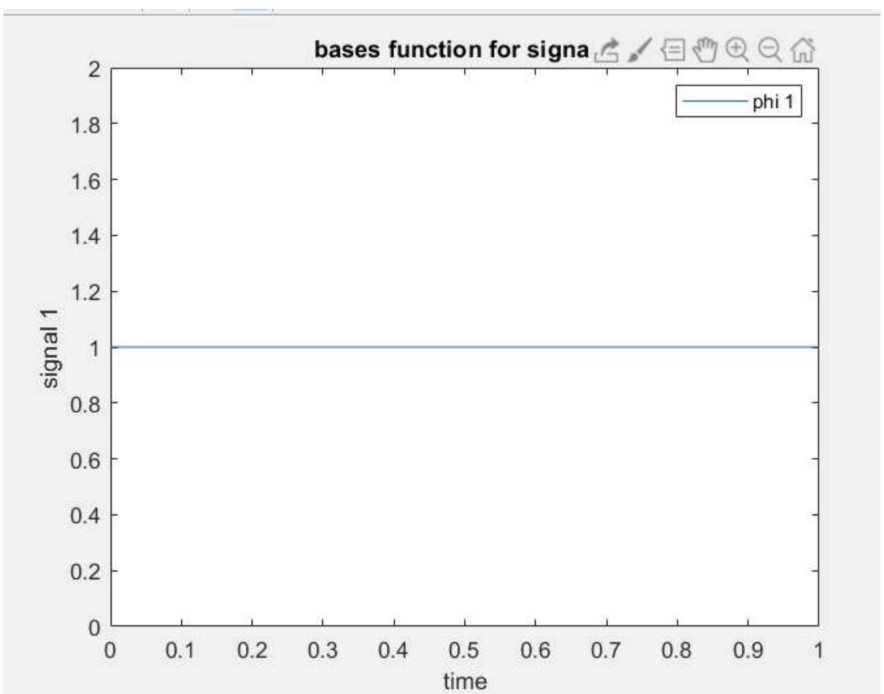

Figure 1 Φ1 VS time after using the GM_Bases function

Phi2:

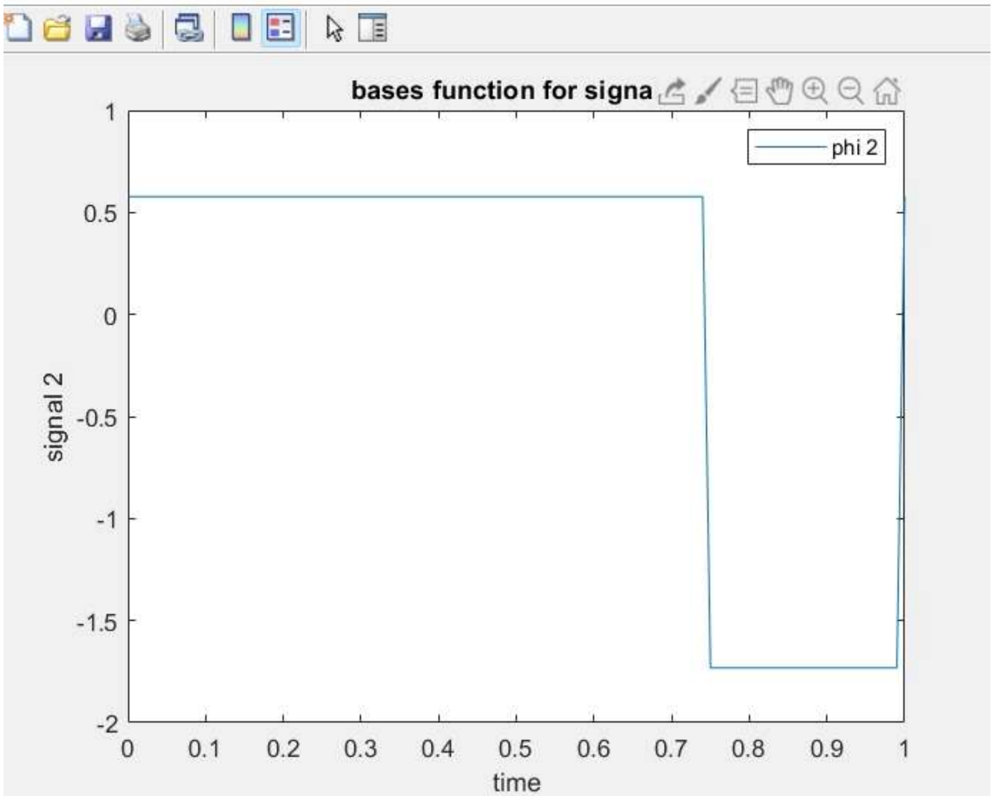

Figure 2 Φ2 VS time after using the GM_Bases function

1.2 Signal Space Representation

Here we represent the signals using the base functions.

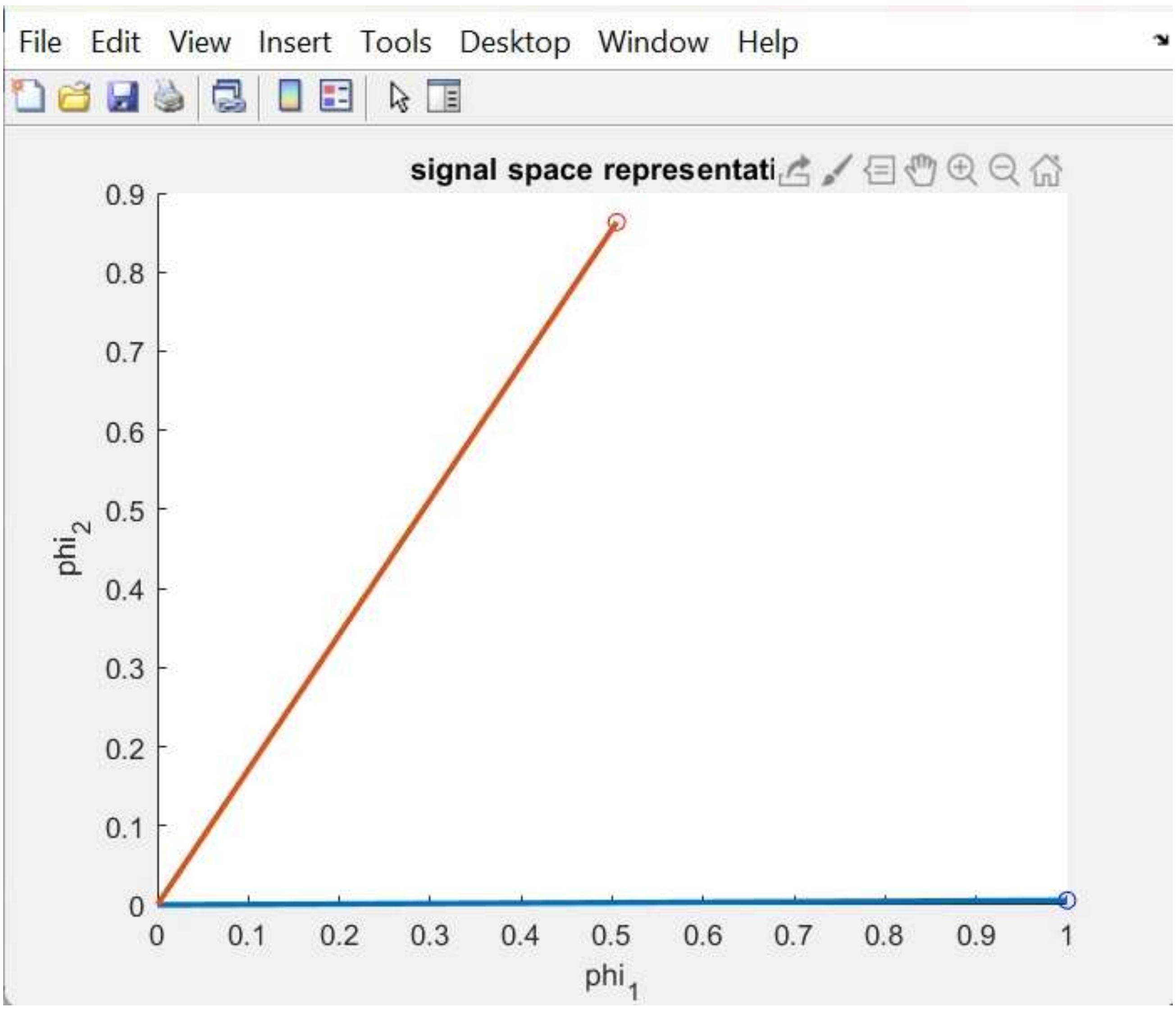

Figure 3 Signal Space representation of signals s1,s2

1.3 Signal Space Representation with adding AWGN

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

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

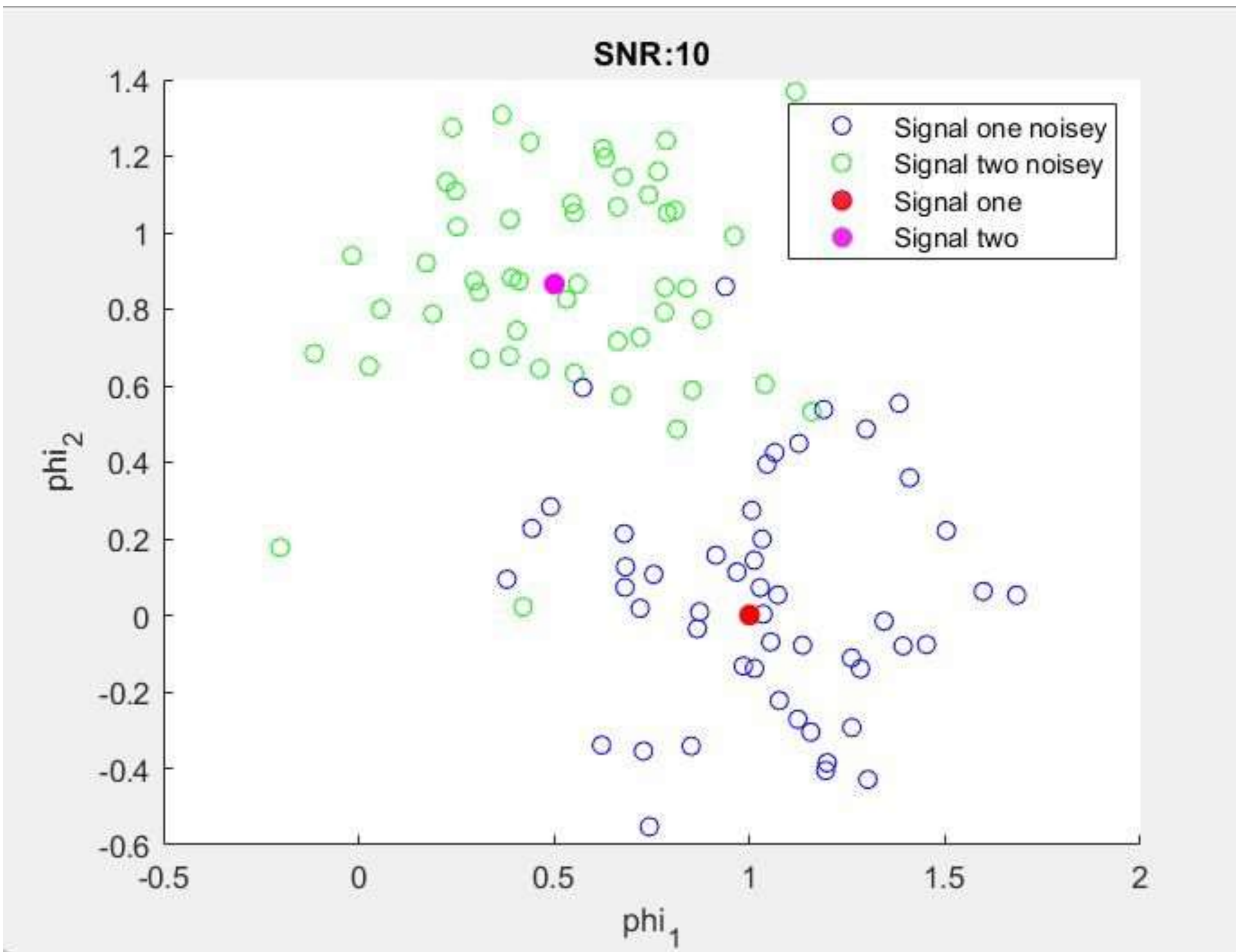

Figure 4 Signal Space representation of signals s1,s2 with E/σ¬2 =10dB

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

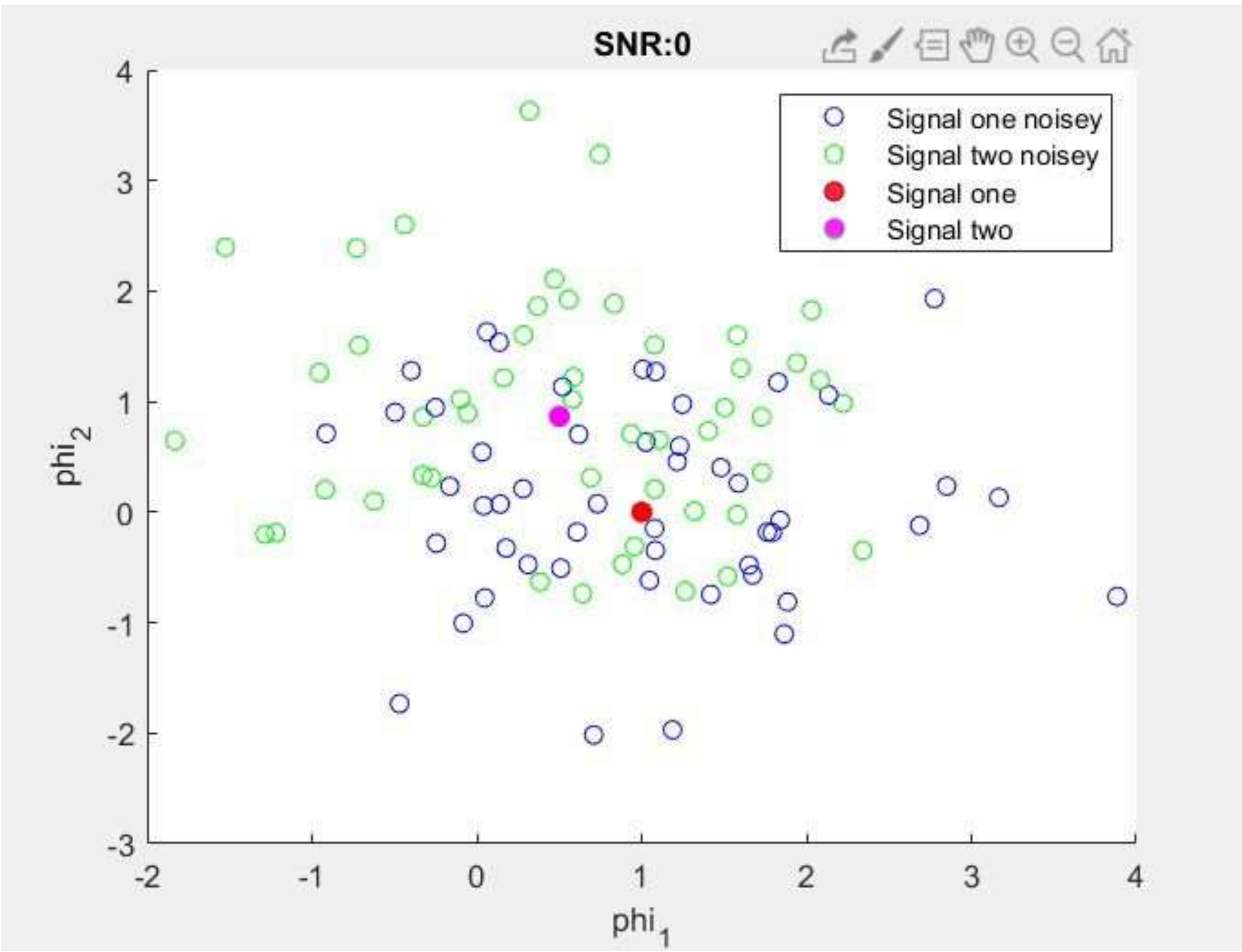

Figure 5 Signal Space representation of signals s1,s2 with E/σ¬2 =0dB

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

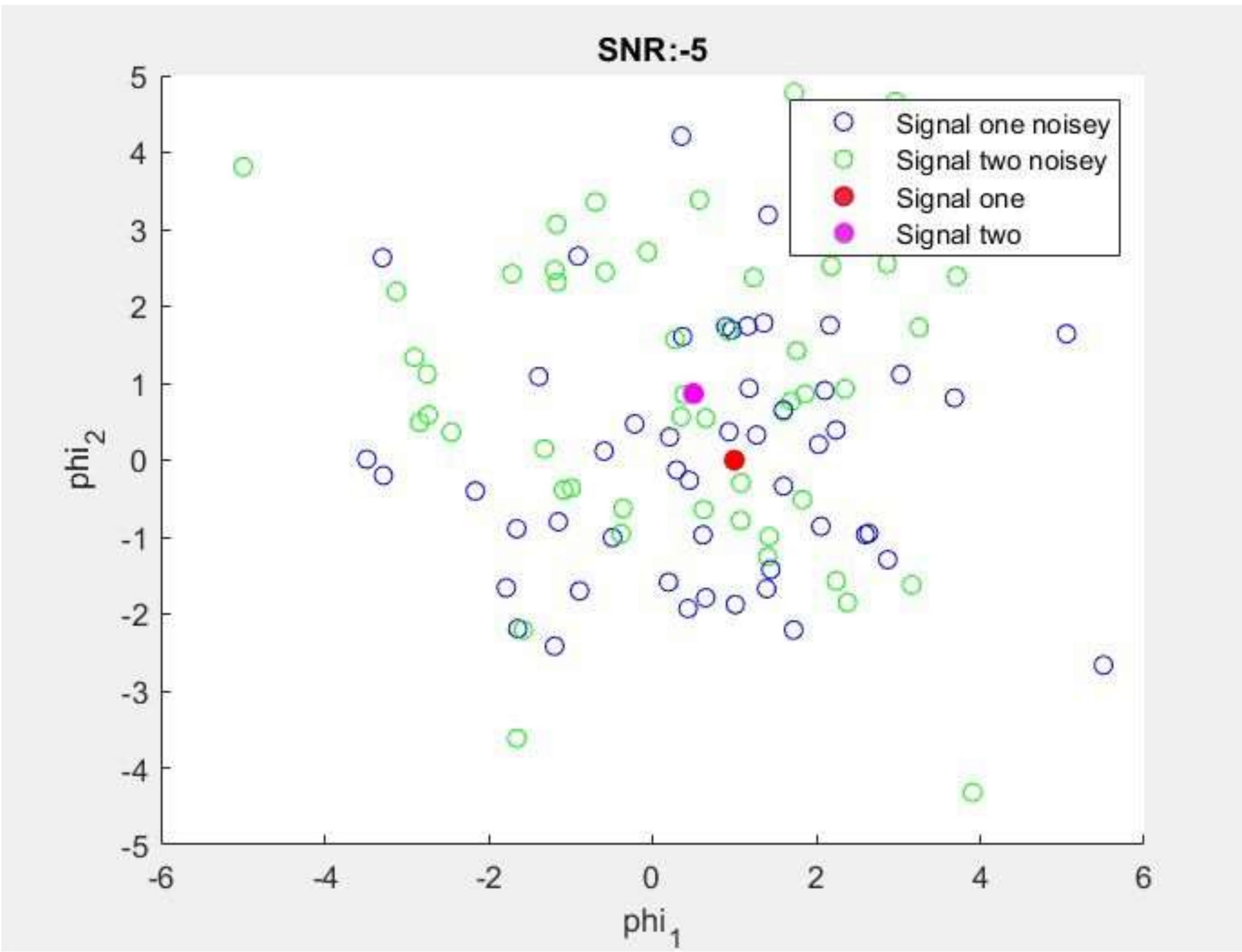

Figure 6 Signal Space representation of signals s1,s2 with E/σ¬2 =-5dB

1.4 Noise Effect on Signal Space

The presence of noise in the signal space can have several effects on the representation of signals:

1. Reduced Signal-to-Noise Ratio (SNR): Noise adds an unwanted component to the signal, which decreases the SNR. As the noise level increases, the ratio of the signal power to the noise power decreases, making it more challenging to distinguish the signal from the noise.

2. Distortion of Signal Components: Noise can introduce random fluctuations or distortions in the signal components within the signal space. It can modify the amplitude, phase, or frequency characteristics of the signal, leading to inaccuracies in the representation.

When the variance σ^2 of the noise increases, it indicates greater variability and amplitude of the noise. This means that the noise component becomes more pronounced relative to the signal component, making it more challenging to extract the desired signal information.

Appendix A: Codes for Part One:

A.1 Code for Gram-Schmidt Orthogonalization

```
%% gat orthonormal bases of signal one and two
function [phi1,phi2] = GM_Bases(s1,s2)
% get integration of signal 1
s11= trapz(0:0.01:1, s1);
% get bases one
phi1 = s1 / s11;

% get s2 in bases one
s21 = phi1.*s2;
% get integration of signal 2
val = trapz(0:0.01:1, s21);
% subtract from siganl two
s2 = s2 - val*phi1;
s2_squared = s2 .* s2;
% get integration of signal 2 after substracted
s22 = trapz(0:0.01:1, s2_squared);
% get bases two
phi2 = s2 / sqrt(s22);
end
```

A.2 Code for Signal Space representation

```
%% get signal space representation of input signal s
function [v1,v2] = signal_space(s,phi1,phi2)
v1 = dot(s, phi1) /length(s);
v2 = dot(s, phi2) /length(s);
end
```

A.3 Code for plotting the bases functions

```
clc;
close all;
clear;
%% Generate signal
% time
t = 0:0.01:1;

duty = 100;
% signal one
s1 = square(2*pi*t, duty);

% Generate square wave with 75% duty cycle
duty = 75;
% signal two
s2 = square(2*pi*t, duty);

% return the Gram-Schmidt orthonormal bases functions
% for both signals s1 and s2
[phi1, phi2] = GM_Bases(s1, s2);

%% plot signal one
Figure
title('bases function for signal 1')
plot(t, phi1)
xlabel('time')
legend('phi 1')

%% plot signal two
Figure
title('bases function for signal 2')
plot(t, phi2)
xlabel('time')
legend('phi 2')
```

A.4 Code for plotting the Signal space Representations

```
%% get signal spaces
% return the projection of s1 and s2 over the
% basis function phi1 and phi2
% get spaces of signal one
[v11, v21] = signal_space(s1, phi1, phi2);
% get spaces of signal two
[v12, v22] = signal_space(s2, phi1, phi2);

%% plot spaces of signal one and two
figure
hold on
title('signal space representation')
scatter(v11, v21, 'blue')
plot([0 v11] , [0 v21], 'LineWidth', 2)
scatter(v12, v22, 'red')
plot([0 v12] , [0 v22], 'LineWidth', 2)
hold off
xlabel('phi_1');
ylabel('phi_2');
```

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

```
%% add noice to signal one and two
function add_noise(s1,s2,phi1,phi2)
% get spaces of signal one
[v11_origin, v21_origin] = signal_space(s1, phi1, phi2);
% get spaces of signal two
[v12_origin, v22_origin] = signal_space(s2, phi1, phi2);
% values of snr
snr = [-5,0,10];
E1 = sum(abs(s1).^2);
E2 = sum(abs(s2).^2);
for i=1:3
    % get snr value
    snr_value=10 ^(snr(i)/10);
    % get variance of signal 1
    variance1 = E1 ./ snr_value;
    % get variance of signal 2
    variance2 = E2 ./ snr_value;
    figure
```

```
    title("signal space with snr = " + snr_value + " dB")
    for j=1:50
        % Add the noise to the original signal
        r1 = s1 +  (sqrt(variance1) * randn(size(s1)));
        % noise on signal two
        r2= s2 +  (sqrt(variance2) * randn(size(s2)));
        % get spaces of signal one noisey
        [v11,v21]=signal_space(r1,phi1,phi2);
        % get spaces of signal two noisey
        [v12,v22]=signal_space(r2,phi1,phi2);

        hold on;
        scatter(v11, v21, 'blue');
        hold on;
        scatter(v12, v22, 'green');
    end
    hold on;
    scatter(v11_origin, v21_origin, 50 , 'red', 'filled');
    hold on;
    scatter(v12_origin, v22_origin, 50 , 'magenta', 'filled');
    xlabel('phi_1');
    ylabel('phi_2');
    title(['SNR:' , num2str(snr(i))]);
    legend( 'Signal one noisey', 'Signal two noisey','Signal one', 'Signal two' );
    hold off;
end
end
```