**Assignment 3**

**Team Members**

|  |  |  |  |
| --- | --- | --- | --- |
| **Num** | **Full Name in ARABIC** | SEC | BN |
| 1 | سارة بشير فكري بشير | 1 | 18 |
| 2 | منه محمد عبدالباسط علي ابراهيم | 2 | 26 |

**Table of contents:**

[1. Part One 3](#_Toc165818042)

[1.1 Gram-Schmidt Orthogonalization 3](#_Toc165818043)

[1.2 Signal Space Representation 5](#_Toc165818044)

[1.3 Signal Space Representation with adding AWGN 6](#_Toc165818045)

[1.4 Noise Effect on Signal Space 8](#_Toc165818046)

[2. Appendix A: Codes for Part One: 9](#_Toc165818047)

[A.1 Code for Gram-Schmidt Orthogonalization 9](#_Toc165818048)

[A.2 Code for Signal Space representation 9](#_Toc165818049)

[A.3 Code for plotting the bases functions 9](#_Toc165818050)

[A.4 Code for plotting the Signal space Representations 10](#_Toc165818051)

[A.5 Code for effect of noise on the Signal space Representations 10](#_Toc165818052)

**List of Figures**

[Figure 1 Φ1 VS time after using the GM\_Bases function 3](#_Toc165818059)

[Figure 2 Φ2 VS time after using the GM\_Bases function 4](#_Toc165818060)

[Figure 3 Signal Space representation of signals s1,s2 5](#_Toc165818061)

[Figure 4 Signal Space representation of signals s1,s2 with E/σ¬2 =10dB 6](#_Toc165818062)

[Figure 5 Signal Space representation of signals s1,s2 with E/σ¬2 =0dB 7](#_Toc165818063)

[Figure 6 Signal Space representation of signals s1,s2 with E/σ¬2 =-5dB 8](#_Toc165818064)

# Part One

## 1.1 Gram-Schmidt Orthogonalization

We have plotted Φ1(t) after calculating it using Gram Schmidt as

Φ1(t) is s1(t) divided by the normalization of s1(t)

And then calculate Φ2(t) which is s2(t) but after the removal of the component in Φ1(t) and then normalizing it

To have Φ1, Φ2 orthonormal to each other

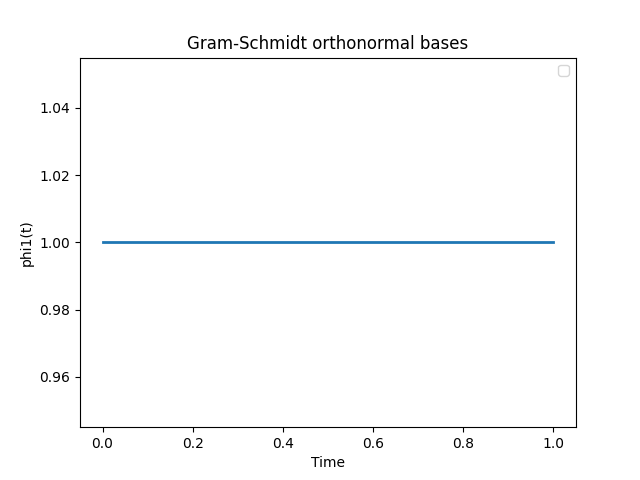


Figure Φ1 VS time after using the GM\_Bases function

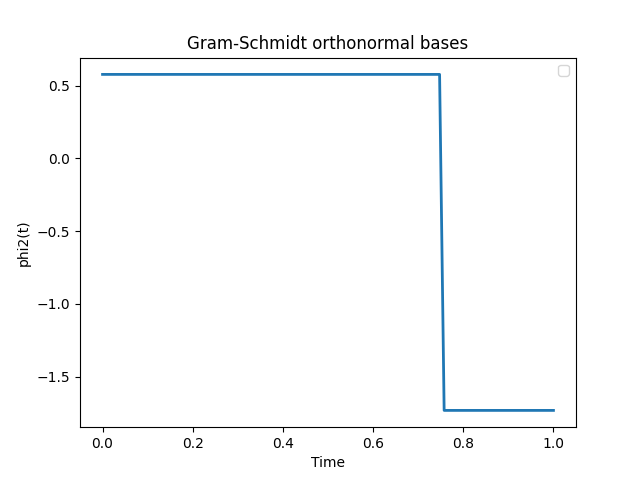


Figure Φ2 VS time after using the GM\_Bases function

## 1.2 Signal Space Representation

Here we represent the signals using the base functions.

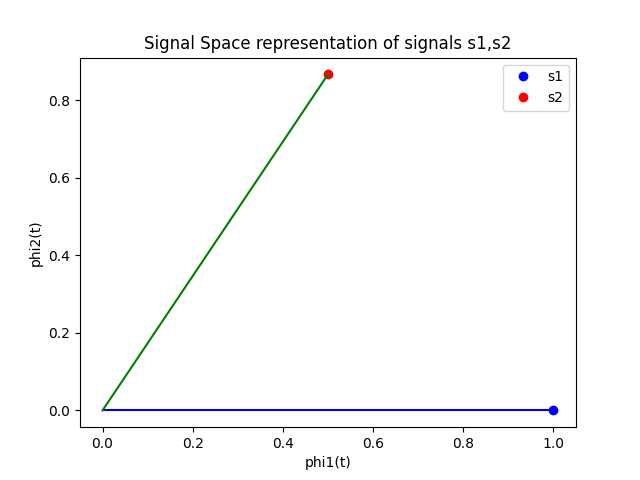


Figure Signal Space representation of signals s1,s2

## 1.3 Signal Space Representation with adding AWGN

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

**Case 1**:

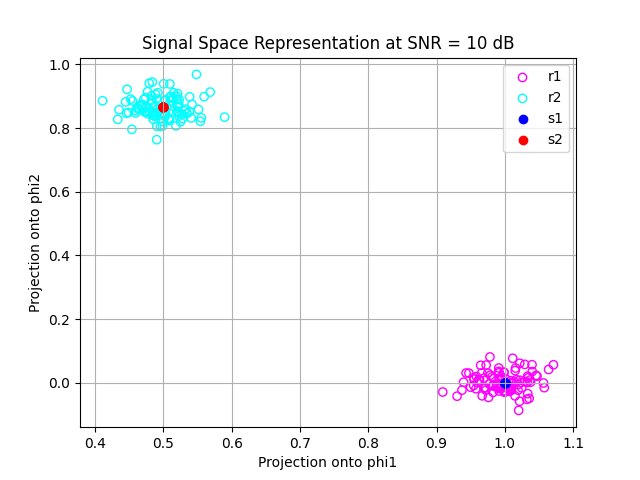


Figure Signal Space representation of signals s1,s2 with E/σ¬2 =10dB

**Case 2**:

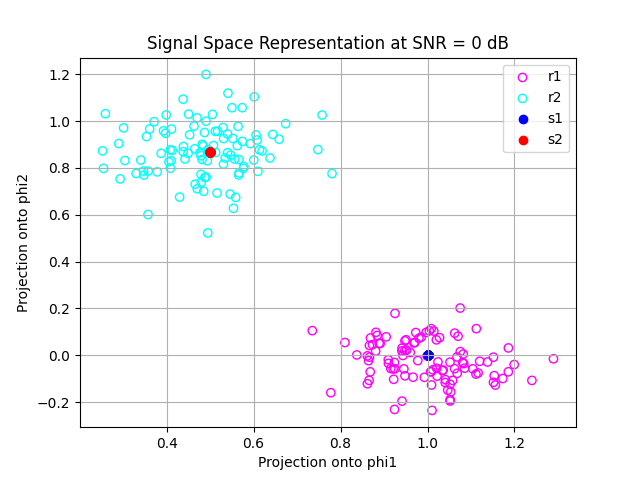


Figure Signal Space representation of signals s1,s2 with E/σ¬2 =0dB

**Case 3**:

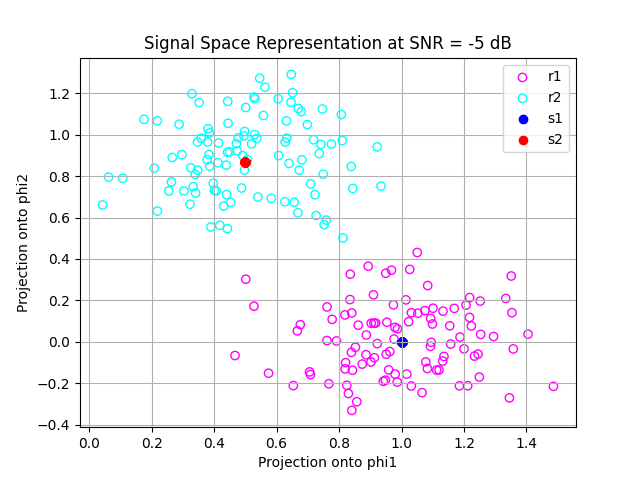


Figure Signal Space representation of signals s1,s2 with E/σ¬2 =-5dB

**Comment**:

From the previous three plots, It is shown that when the value of SNR increases the points get close more to the intended point that it should belong to as probability of error decreases.

## 1.4 Noise Effect on Signal Space

As SNR increases the noise effect decreases as sigma decreases

so it enhances the difference between the two signals in the signal space and minimizes the probability of error

# Appendix A: Codes for Part One:

## A.1 Code for Gram-Schmidt Orthogonalization

def mag(s):

    return np.sqrt(np.dot(s,s)/sample\_num)

def GM\_Bases(s1, s2):

    phi1 = s1 / mag(s1)

    phi2 = s2 -((np.dot(phi1, s2) \* phi1)/sample\_num)

    phi2 = phi2 / mag(phi2)

    return phi1, phi2

## A.2 Code for Signal Space representation

def signal\_space(s, phi1, phi2):

    v1 = np.dot(s, phi1)/sample\_num

    v2 = np.dot(s, phi2)/sample\_num

    return v1, v2

## A.3 Code for plotting the bases functions

phi1,phi2=GM\_Bases(s1,s2)

plt.plot(t,phi1,linewidth=2)

plt.title("Gram-Schmidt orthonormal bases")

plt.xlabel("Time")

plt.ylabel("phi1(t)")

plt.legend()

plt.show()

plt.plot(t,phi2,linewidth=2)

plt.title("Gram-Schmidt orthonormal bases")

plt.xlabel("Time")

plt.ylabel("phi2(t)")

plt.legend()

plt.show()

## 

## A.4 Code for plotting the Signal space Representations

v1,v2=signal\_space(s1,phi1,phi2)

v3,v4=signal\_space(s2,phi1,phi2)

# Plot the signal space representation for the two signals

plt.plot(v1,v2,'bo',label='s1')

plt.plot(v3,v4,'ro',label='s2')

plt.plot([0, v1], [0, v2], 'b')

plt.plot([0, v3], [0, v4], 'g')

plt.title("Signal Space representation of signals s1,s2")

plt.xlabel("phi1(t)")

plt.ylabel("phi2(t)")

plt.legend()

plt.show()

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

# calculate the Energy of each signal

e1 = np.dot([v1,v2],[v1,v2])

e2 = np.dot([v3,v4],[v3,v4])

# Function to add noise to a signal based on SNR in dB

def add\_noise(s, SNR\_dB,E):

    sigma2 = E \* 10\*\*(-SNR\_dB/10)

    noise = np.random.normal(0, np.sqrt(sigma2), sample\_num)

    return s + noise

# SNRs to test

SNRs = [-5, 0, 10]

for SNR in SNRs:

    # Signal space projection

    v1\_r1\_org, v2\_r1\_org = signal\_space(s1, phi1, phi2)

    v1\_r2\_org, v2\_r2\_org = signal\_space(s2, phi1, phi2)

    plt.figure()

    plt.title('Signal Space Representation at SNR = {} dB'.format(SNR))

    for i in range(sample\_num):

        r1 = add\_noise(s1, SNR,e1)

        r2 = add\_noise(s2, SNR,e2)

        v1\_r1, v2\_r1 = signal\_space(r1, phi1, phi2)

        v1\_r2, v2\_r2 = signal\_space(r2, phi1, phi2)

        # Plotting

        plt.scatter(v1\_r1, v2\_r1, color='magenta',facecolors='none', marker='o',label='r1')

        plt.scatter(v1\_r2, v2\_r2, color='cyan',facecolors='none',marker='o',label='r2')

        plt.scatter(v1\_r1\_org, v2\_r1\_org, color='blue', label='s1')

        plt.scatter(v1\_r2\_org, v2\_r2\_org, color='red', label='s2')

    plt.legend(['r1','r2','s1','s2'])

    plt.grid(True)

    plt.xlabel('Projection onto phi1')

    plt.ylabel('Projection onto phi2')

    plt.show()