**Assignment 3**

**Team Members**

|  |  |  |  |
| --- | --- | --- | --- |
| **Num** | **Full Name in ARABIC** | SEC | BN |
| 1 | اسماء عادل عبدالحميد | 1 | 14 |
| 2 | سماء حازم عبداللطيف | 1 | 32 |

**Table of contents:**

[1. Part One 3](#_Toc135281088)

[1.1 Gram-Schmidt Orthogonalization 3](#_Toc135281089)

[1.2 Signal Space Representation 4](#_Toc135281090)

[1.3 Signal Space Representation with adding AWGN 5](#_Toc135281091)

[1.4 Noise Effect on Signal Space 6](#_Toc135281092)

[2. Appendix A: Codes for Part One: 7](#_Toc135281093)

[A.1 Code for Gram-Schmidt Orthogonalization 7](#_Toc135281094)

[A.2 Code for Signal Space representation 7](#_Toc135281095)

[A.3 Code for plotting the bases functions 7](#_Toc135281096)

[A.4 Code for plotting the Signal space Representations 8](#_Toc135281097)

[A.5 Code for effect of noise on the Signal space Representations 8](#_Toc135281098)

**List of Figures**

[Figure 1 Φ1 VS time after using the GM\_Bases function 5](#_Toc135281077)

[Figure 2 Φ2 VS time after using the GM\_Bases function 5](#_Toc135281078)

[Figure 3 Signal Space representation of signals s1,s2 6](#_Toc135281079)

[Figure 4 Signal Space representation of signals s1,s2 with E/σ¬2 =10dB 7](#_Toc135281080)

[Figure 5 Signal Space representation of signals s1,s2 with E/σ¬2 =0dB 7](#_Toc135281081)

[Figure 6 Signal Space representation of signals s1,s2 with E/σ¬2 =-5dB 8](#_Toc135281082)

# Part One

## 1.1 Gram-Schmidt Orthogonalization

The function calculates the Gram-Schmidt orthonormal bases functions (phi1 & phi 2) for two input signals (s1 & s2)

A picture containing text, screenshot, display, rectangle

Description automatically generated

Figure 1 Φ1 VS time after using the GM\_Bases function

A picture containing text, screenshot, rectangle, diagram

Description automatically generated

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.

A graph with green and yellow dots

Description automatically generated with low confidence

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**:

A chart with green and yellow dots

Description automatically generated with low confidence

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

**Case 2**:

A chart with green and yellow dots

Description automatically generated with low confidence

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

**Case 3**:

A chart with green and yellow dots

Description automatically generated with low confidence

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

## 1.4 Noise Effect on Signal Space

# Appendix A: Codes for Part One:

## A.1 Code for Gram-Schmidt Orthogonalization

def Gram\_Schmidt(S1, S2):

# calculate the energy of the s1(t)

Energy1 = abs(S1)\*\*2

# calculate Gram-Schmidt base function Phi1

Phi1 = S1 / np.sqrt(Energy1)

S21 = np.sum(S2 \* Phi1) / numOfSamples

# calculate Gram-Schmidt second orthogonalized

g2 = S2 - S21 \* Phi1

# calculate the energy of the s2(t)

Energy2 = np.sum(g2 \*\* 2) / numOfSamples

# calculate Gram-Schmidt base function Phi2

Phi2 = g2 / np.sqrt(Energy2)

return Phi1, Phi2

## A.2 Code for Signal Space representation

def Signal\_Space(S, Phi1, Phi2):

# calculate the signal space representation

V1 = np.sum(S \* Phi1)

V2 = np.sum(S \* Phi2)

return V1, V2

## A.3 Code for plotting the bases functions

# get the bases functions of s1(t) & s2(t)

Phi1, Phi2 = Gram\_Schmidt(S1, S2)

# plot Phi1 vs time

plt.figure(3)

plt.plot(timeAxis, Phi1, linewidth=2)

plt.vlines(x=0, ymin=0, ymax=1)

plt.vlines(x=1, ymin=0, ymax=1)

plt.title('Gram-Schmidt For S1(t)')

plt.xlabel("time")

plt.ylabel("Phi1(t)")

plt.legend()

# plot Phi2 vs time

plt.figure(4)

plt.plot(timeAxis, Phi2, linewidth=2)

plt.vlines(x=0, ymin=0, ymax=Phi2[0])

plt.vlines(x=1, ymin=Phi2[numOfSamples-1], ymax=0)

plt.title('Gram-Schmidt For S2(t)')

plt.xlabel("time")

plt.ylabel("Phi2(t)")

plt.legend()

plt.show()

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

# get the signal space representation of s1(t) & s2(t)

V11, V12 = Signal\_Space(S1, Phi1, Phi2)

V21, V22 = Signal\_Space(S2, Phi1, Phi2)

# Plot the signal space representation

plt.figure(5)

plt.scatter(V11,V12, label='S1', c='g')

plt.scatter(V21,V22 , label='S2', c='y')

plt.title('Signal space')

plt.xlabel("Phi1(t)")

plt.ylabel("Phi2(t)")

plt.legend()

plt.show()

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

# Generate E/σ2 array in db

EOverSigma2\_db\_arr = [-5, 0, 10]

E = 1 # as energy for S1, S2 = 1

# for loop for each element in E/σ2 array

for EOverSigma2\_db in EOverSigma2\_db\_arr:

# plot noise signal Phit1 vs Phi2

plt.title('Noise ('+str(EOverSigma2\_db)+')dB')

plt.xlabel("Phi1(t)")

plt.ylabel("Phi2(t)")

# for loop 50 times for random noise samples

for i in range(50):

# calculate standard deviation

standardDev = E/(10\*\*(EOverSigma2\_db/10))

# generate random noise samples

W = np.random.normal(0, np.sqrt(standardDev), numOfSamples)

# add noise for signals

r1 = S1 + W

r2 = S2 + W

# get the signal space representation of r1(t) & r2(t)

V11\_Req\_3, V12\_Req\_3 = Signal\_Space(r1, Phi1, Phi2)

V21\_Req\_3, V22\_Req\_3 = Signal\_Space(r2, Phi1, Phi2)

# plot the signal space representation

plt.scatter(V11\_Req\_3, V12\_Req\_3, c='g')

plt.scatter(V21\_Req\_3, V22\_Req\_3, c='y')

plt.legend(["r1", "r2"])

plt.show()