Cairo University
Faculty of
Engineering

Department of Computer
Engineering

ELC 325B – Spring 2023

Digital Communications

Assignment #3

Submitted to

Dr. Mai

Dr. Hala

Eng. Mohamed Khaled

Submitted by

Name	Sec	BN
Nancy Ayman	2	27
Yara Hisham	2	31

Table of Contents

Table of Figures

Gram-Schmidt Orthogonalization

S1,and S2 after applying Gram Schmidt to the signals to get their bases.

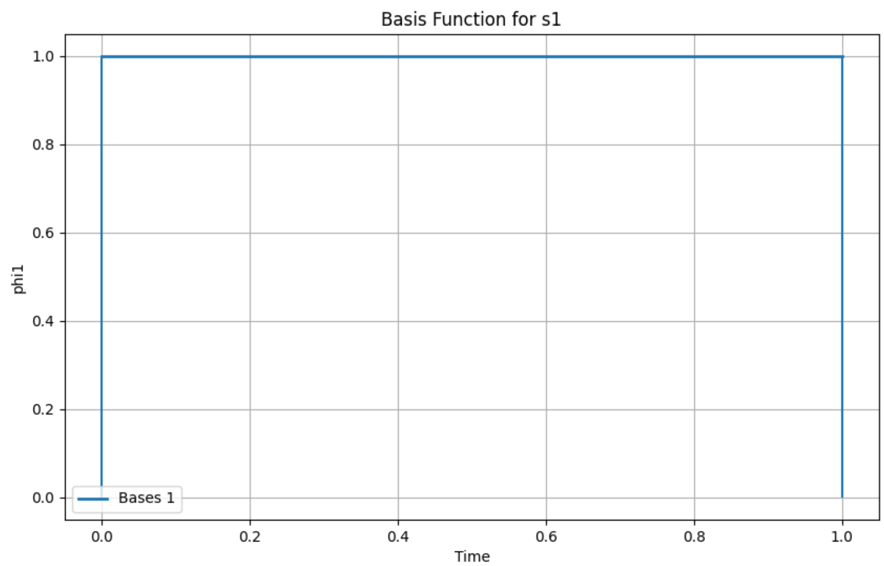

Figure 1 S1 bases plot

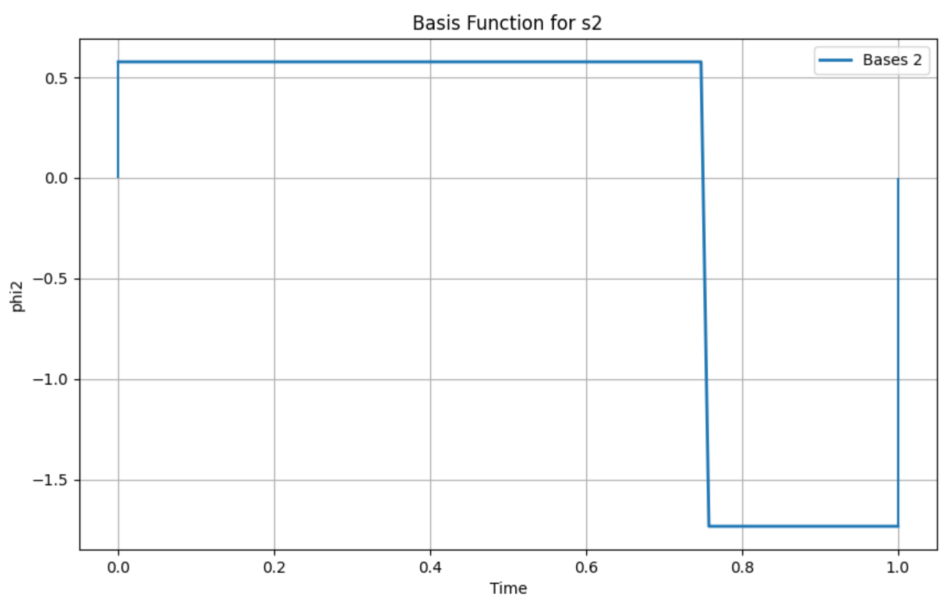

Figure 2 S2 bases plot

Signal Space Representation

The Representation of S1, and S2 using their bases

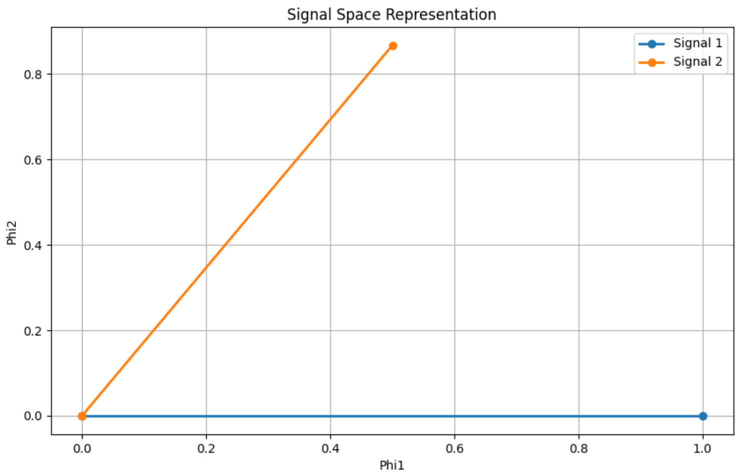

Figure 3 signal space for S1, S2

Signal Space Representation with adding AWGN.

The expected signal received and the actual received signal due to noise.

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

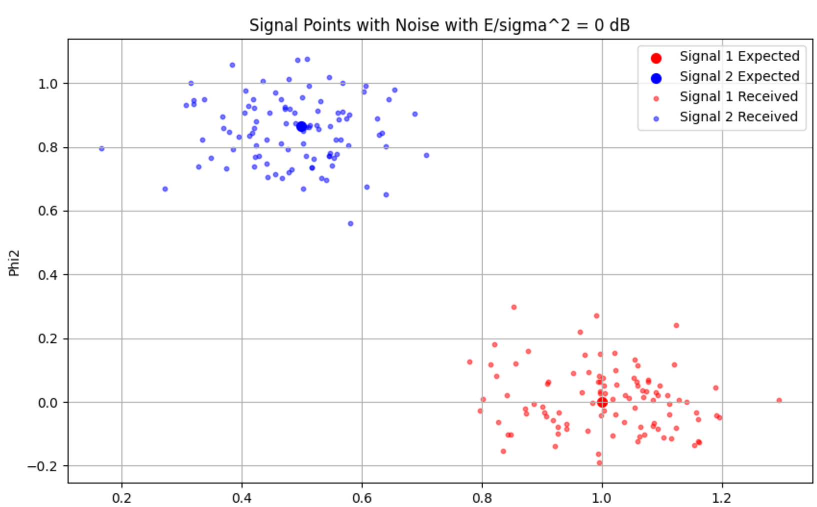

Figure 4 signal points with noise 10 log(E/σ^2) = 0 dB

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

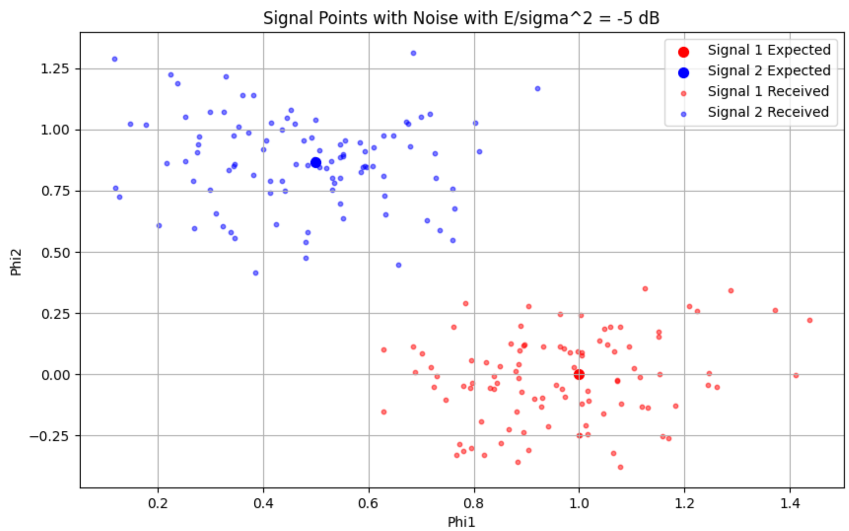

Figure 5 signal points with noise 10 log(E/σ^2) = -5 dB

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

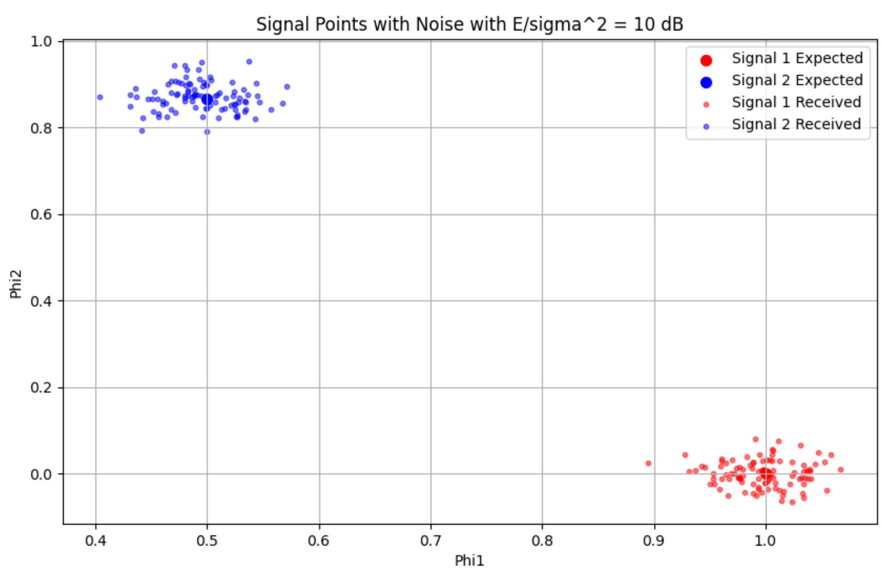

Figure 6 signal points with noise 10 log(E/σ^2) = 10 dB

Code

Code for Gram-Schmidt Orthogonalization

```
def Gram_Schmidt( s1,s2):
    s11=np.sqrt(np.sum(s1**2))
    phi1=s1/s11
    s21=np.sum(s2*phi1)
    gT=s2-s21*phi1
    s22=np.sqrt(np.sum(gT**2))
    phi2=gT/s22
    phi1*=np.sqrt(len(s1))
    phi2*=np.sqrt(len(s2))
    return phi1,phi2
```

Code for Signal Space representation

```
def signal_space(s, phi1,phi2):
    V1 = np.dot(s,phi1)/len(s)
    V2 = np.dot(s,phi2)/len(s)
    return V1,V2
```

Code for plotting the bases functions

```
t = np.linspace(0, 1, SAMPLESNUM)
s1 = np.ones(SAMPLESNUM)
s2 = np.concatenate((np.ones(int(SAMPLESNUM*(75/100))), -
np.ones(int(SAMPLESNUM*(25/100)))))

# Get orthonormal bases of signal one and two
phi1, phi2 = Gram_Schmidt(s1, s2)


# Plot the basis functions
plt.figure(figsize=(10, 6))
plt.plot(t, phi1, linewidth=2, label='Bases 1')
plt.xlabel('Time')
plt.ylabel('phi1')
plt.vlines(x=0, ymin=0, ymax=phi1[0])
plt.vlines(x=1, ymin=0, ymax=phi1[len(phi1)-1])
plt.title('Basis Function for s1')
plt.legend()
plt.grid(True)
plt.savefig('Basis Function for s1')

plt.show()
```

```
plt.figure(figsize=(10, 6))
plt.plot(t, phi2, linewidth=2, label='Bases 2')
plt.xlabel('Time')
plt.ylabel('phi2')
plt.title('Basis Function for s2')
plt.vlines(x=0, ymin=0, ymax=phi2[0])
plt.vlines(x=1, ymin=phi2[len(phi1)-1], ymax=0)
plt.legend()
plt.grid(True)
plt.savefig('Basis Function for s2')
plt.show()
```

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 space representation
plt.figure(figsize=(10, 6))
plt.plot([0, s1_v1], [0, s1_v2], '-o', linewidth=2, label='Signal 1')
plt.plot([0, s2_v1], [0, s2_v2], '-o', linewidth=2, label='Signal 2')
plt.xlabel('Phi1')
plt.ylabel('Phi2')
plt.title('Signal Space Representation')
plt.legend()
plt.grid(True)
plt.savefig('signal_space_representation.png')

plt.show()
```

Code for effect of noise on the Signal space Representations

```
def signal_space_with_noise(s, sigma):
    noise =  np.random.normal(0, np.sqrt(sigma), SAMPLESNUM)
    return s + noise

def plot_signal_with_noise(testCase, s1_v1, s1_v2, s2_v1, s2_v2, s1,
s2, phi1, phi2):
    EoSigma = [-5, 0, 10]
    Es1 = np.dot(s1, s1) / len(s1)
    Es2 = np.dot(s2, s2) / len(s2)
    sigma1 = (Es1 / np.power(10, EoSigma[testCase] / 10))
    sigma2 = (Es2 / np.power(10, EoSigma[testCase] / 10))
```

```python
    plt.figure(figsize=(10, 6))
    plt.scatter(s1_v1, s1_v2, c='r', s=50, label='Signal 1 Expected')
    plt.scatter(s2_v1, s2_v2, c='b', s=50, label='Signal 2 Expected')

    for _ in range(SAMPLESNUM):
        r1 = signal_space_with_noise(s1, sigma1)
        r2 = signal_space_with_noise(s2, sigma2)
        r1_v1, r1_v2 = signal_space(r1, phi1, phi2)
        r2_v1, r2_v2 = signal_space(r2, phi1, phi2)
        plt.scatter(r1_v1, r1_v2, c='r', s=10, alpha=0.5)
        plt.scatter(r2_v1, r2_v2, c='b', s=10, alpha=0.5)

    plt.scatter([], [], c='r', s=10, alpha=0.5, label='Signal 1 
Received')
    plt.scatter([], [], c='b', s=10, alpha=0.5, label='Signal 2 
Received')

    plt.xlabel('Phi1')
    plt.ylabel('Phi2')
    title_string = 'Signal Points with Noise with E/sigma^2 = {} 
dB'.format(EoSigma[testCase])

    plt.title(title_string)
    plt.legend()
    plt.grid(True)
    plt.savefig('{}.png'.format(title_string.replace('/', '_')))

    plt.show()

plot_signal_with_noise(0, s1_v1, s1_v2, s2_v1, s2_v2, s1, s2, phi1, 
phi2)
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)
```

How does the noise affect the signal space? Does the noise effect increase or decrease with increasing $\sigma 2$?

Noise affects the signal space by adding random fluctuations to the signal. Increasing $\sigma 2$ amplifies noise impact on signal space, decreasing signal clarity and SNR, impairing system performance. As shown in the previous graphs as the noise increased the probability of the signal to match the expected value decreased.