Cairo University - Faculty Of Engineering
Computer Engineering Department
Digital Communication - Spring 2025

Digital Communications: Assignment 3 (Signal Space)

Submitted to

Dr. Mai Badawi
Dr. Hala
Eng. Mohamed Khaled

Submitted by

Akram Hany Sec 1 BN 14
Amir Anwar Sec 1 BN 15

Contents

List of Figures

1 Gram-Schmidt Orthogonalization:

We apply the Gram-Schmidt process to two input signals $s_1(t)$ and $s_2(t)$.

1.1 Original Signals

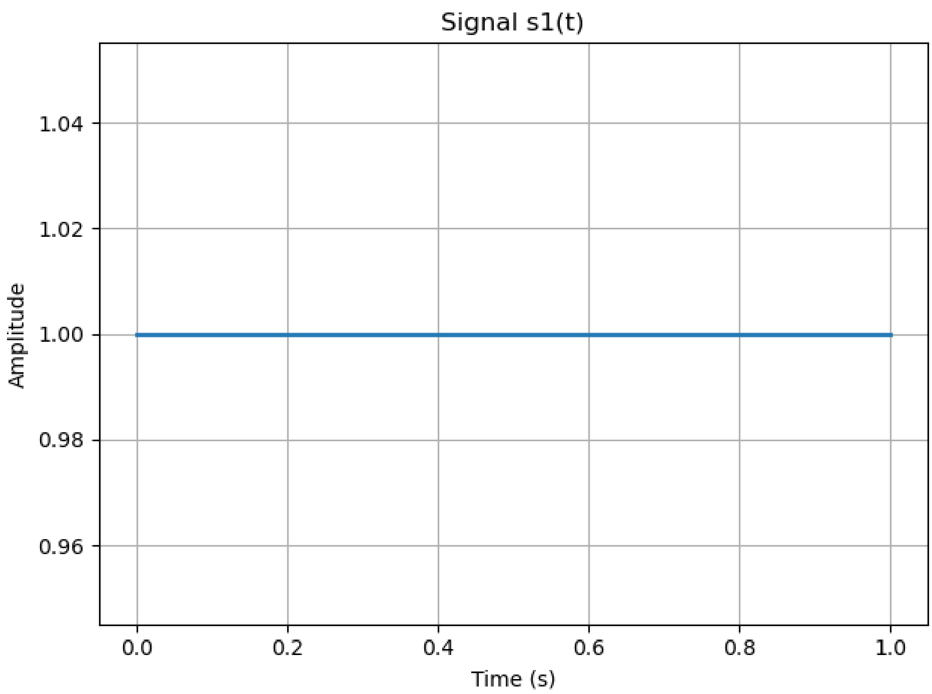

Figure 1: Signal $s_1(t)$

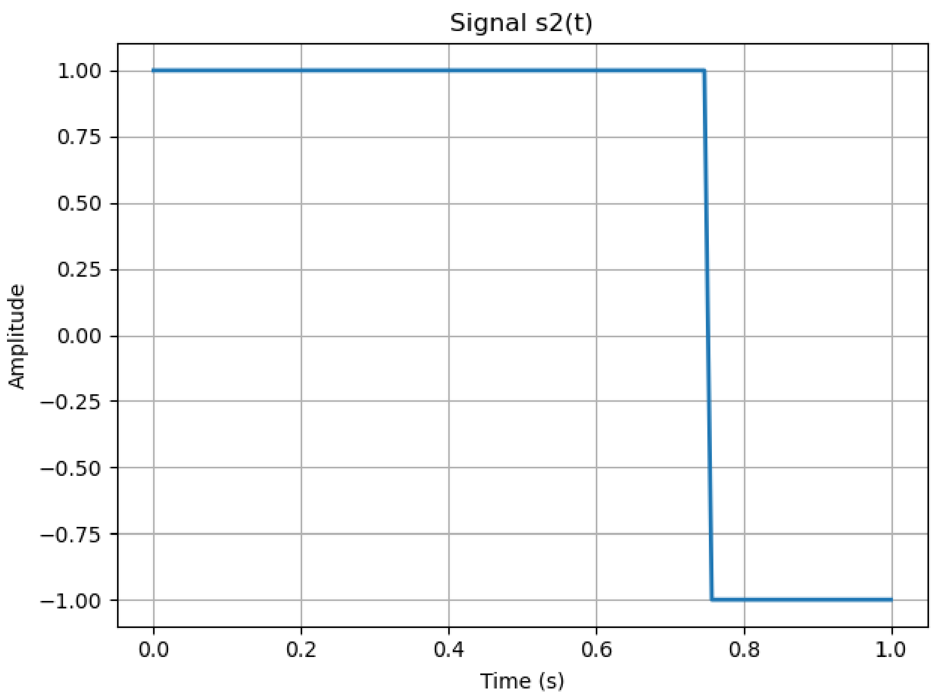

Figure 2: Signal $s_2(t)$

1.2 Basis Functions

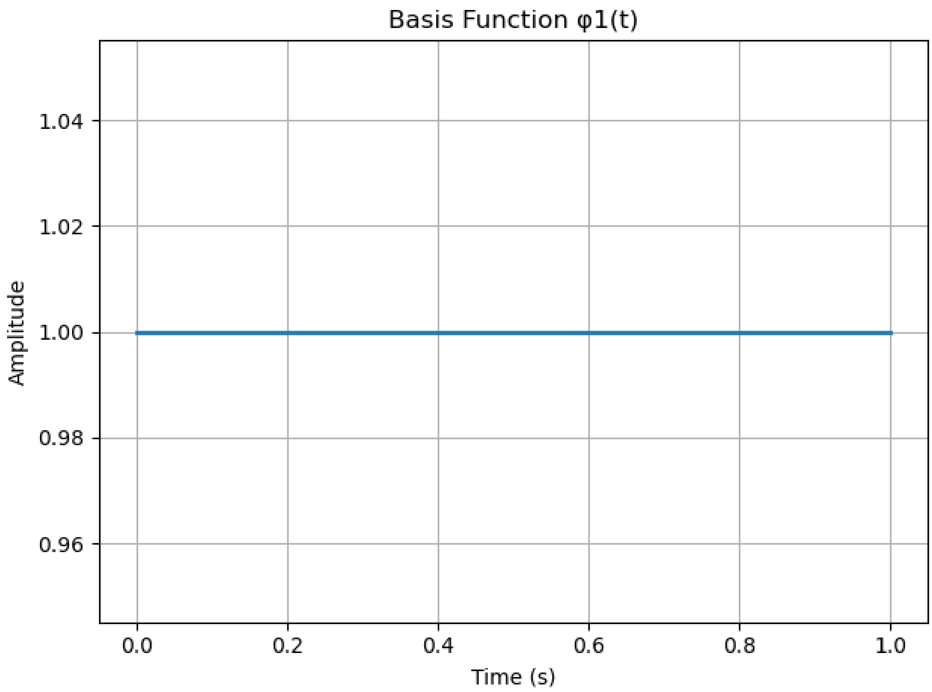

Figure 3: Basis Function $\phi_1(t)$

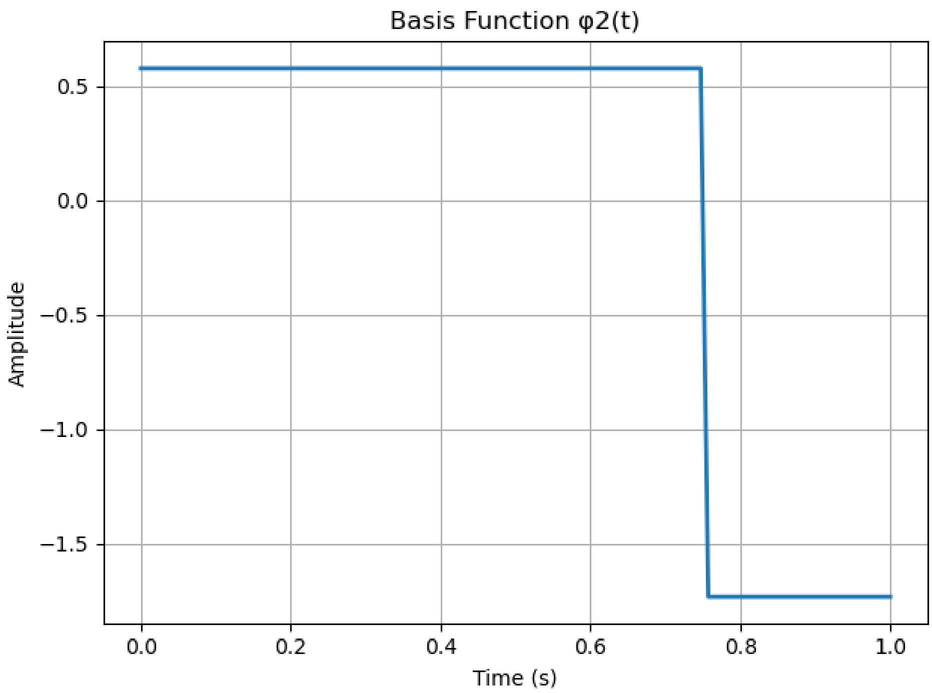

Figure 4: Basis Function $\phi_2(t)$

2 Part 2: Signal Space Representation

Using the orthonormal basis, we project $s_1(t)$ and $s_2(t)$ into the signal space.

- s_1: $(v_1, v_2) \approx (1, 0)$
- s_2: $(v_1, v_2) \approx (0.5, 0.8)$

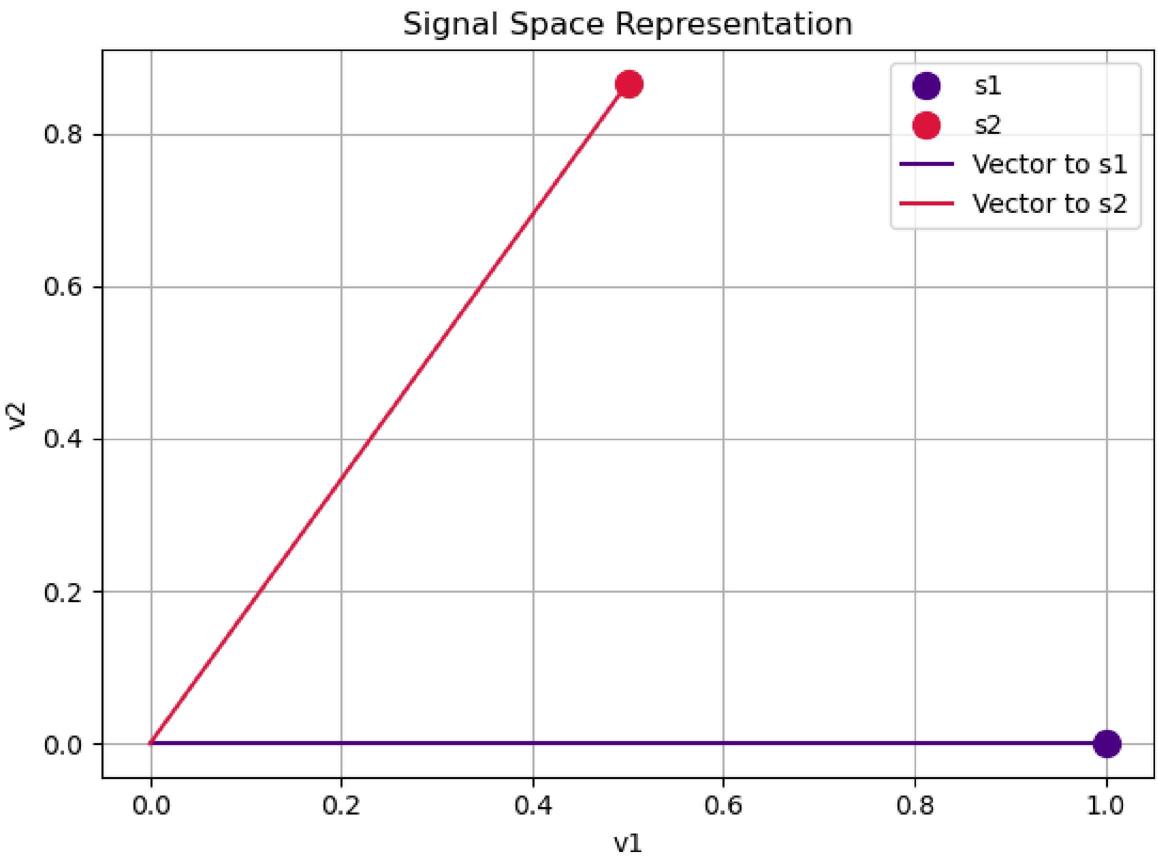

Figure 5: Signal space projection of $s_1(t)$ and $s_2(t)$

2.1 Comment on the signal space representation

As it shows S_1 is aligned with basis that it looks like it completely with no projection on the other axis on the contrary S_2 has projection on the 2 basis.

3 Part 3: Effect of AWGN on Signal Space

We add AWGN (additive white gaussian noise) to both signals and project 100 noisy versions each to signal space.

3.1 SNR = -5 dB

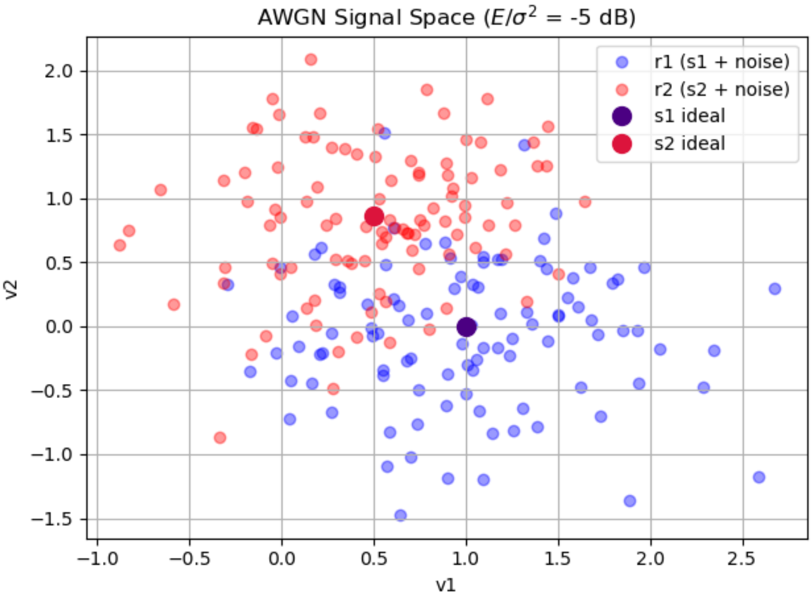

Figure 6: AWGN Scatter Plot for $E/\sigma^2 = -5$ dB

3.2 SNR = 0 dB

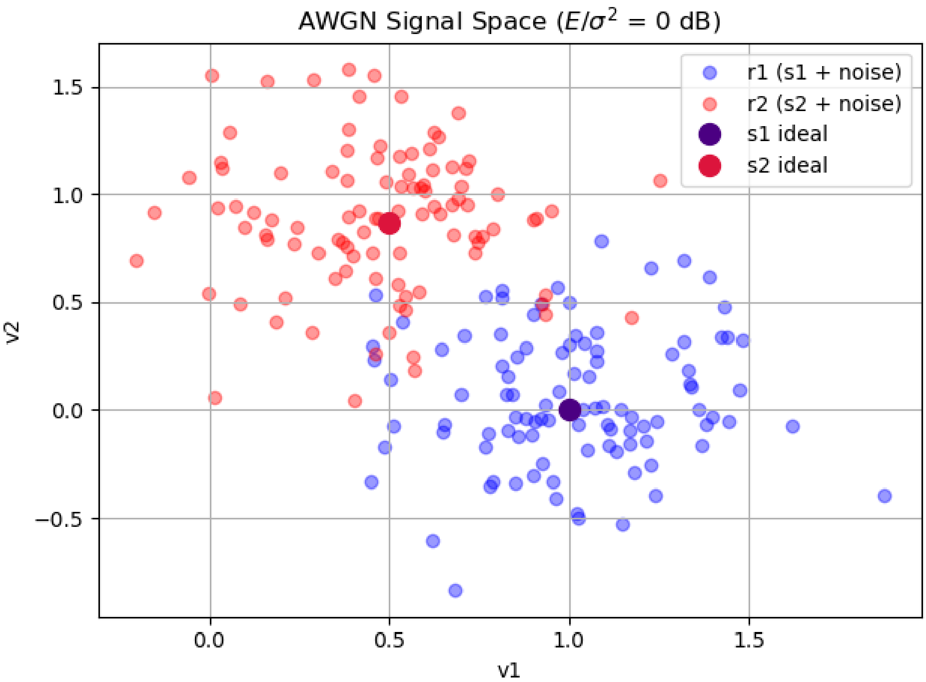

Figure 7: AWGN Scatter Plot for $E/\sigma^2 = 0$ dB

3.3 SNR = 10 dB

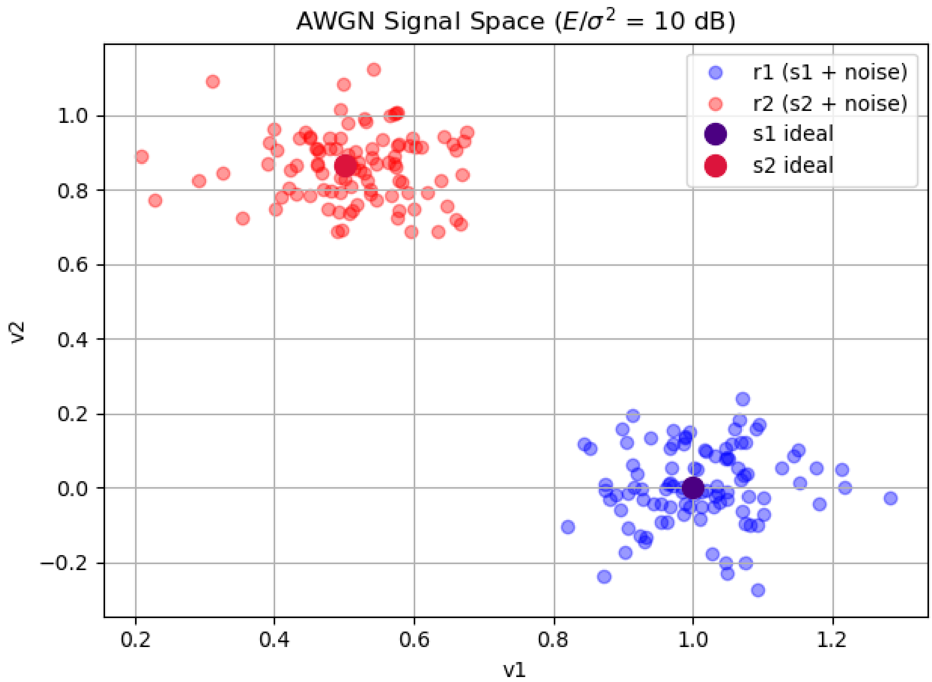

Figure 8: AWGN Scatter Plot for $E/\sigma^2 = 10$ dB

3.4 SNR = 15 dB

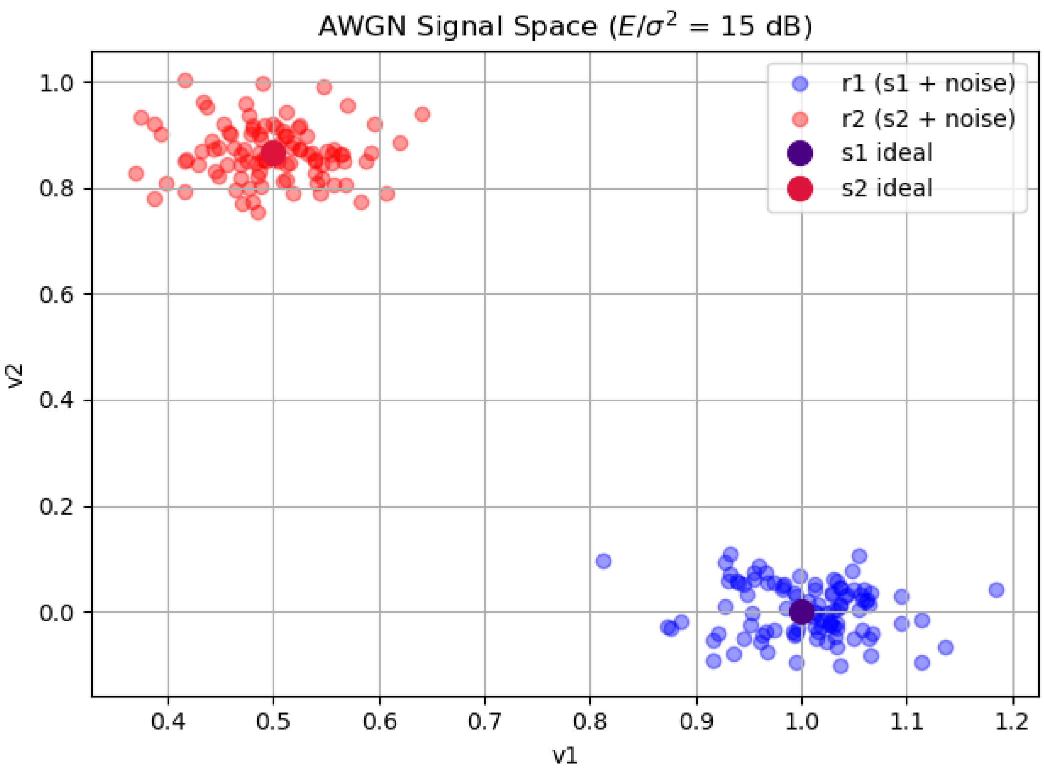

Figure 9: AWGN Scatter Plot for $E/\sigma^2 = 15$ dB

Note I added SNR = 15 dB because 10 dB was not enough to show the separation.

4 Part 4: Discussion

How does noise affect the signal space?

Noise introduces variation around the signal space points. The clusters of noisy projections around s_1 and s_2 become wider as noise increases.

Does noise effect increase or decrease with increasing σ^2?

As σ^2 increases (i.e., lower SNR), noise spreads the points more, making signals less distinguishable and increasing error probability. So, **the noise effect increases with increasing σ^2.**

5 Appendix

5.1 Python Code

```python
import matplotlib.pyplot as plt
import numpy as np
```

Listing 1: Imports and external dependecies

```python
def gm_bases(s1, s2):
    # applying 2 steps of Gram-Schmidt process
    norm_s1 = np.linalg.norm(s1)
    phi1 = s1 / norm_s1

    proj = np.dot(s2, phi1) * phi1
    ortho = s2 - proj

    norm_ortho = np.linalg.norm(ortho)
    phi2 = ortho / norm_ortho if norm_ortho != 0 else np.zeros_like(s1)

    return phi1, phi2
```

Listing 2: Gram-Schmidt Orthogonalization (part 1)

```python
def signal_space(s, phi1, phi2):
    # NOTE: Simply project into phi1, phi2 axes (linear algebra)
    v1 = np.dot(s, phi1) / NUMBER_SAMPLES
    v2 = np.dot(s, phi2) / NUMBER_SAMPLES
    return v1, v2
```

Listing 3: Signal Space representation: (part 2)

```python
def add_awgn(signal, sigma2):
    # NOTE: awgn: Additive White Gaussian Noise
    noise = np.random.normal(0, np.sqrt(sigma2), signal.shape)
    return signal + noise
```

Listing 4: AWGN: (part 3)

```python
def plot_signal(t, signal, title, filename):
    # Simple utility to not repeat this 4 times below :)
    plt.figure()
    plt.plot(t, signal, linewidth=2)
    plt.title(title)
    plt.xlabel("Time (s)")
    plt.ylabel("Amplitude")
    plt.grid(True)
    plt.tight_layout()
    plt.savefig(filename)
    plt.close()
```

Listing 5: Utility function to plot a signal

```python
def run_awgn_experiment(
    s1, s2, phi1, phi2, num_samples=100, eb_sigma2_db_list=[-5, 0, 10, 15]
):
    # NOTE: s1, s2 are the signals, phi1, phi2 are the basis functions
    # NOTE: eb_sigma2_db_list is the list of SNR values in dB (E_b/N)

    Es = np.sum(s1**2)
    Es /= np.sqrt(NUMBER_SAMPLES)

    v1_s1, v2_s1 = signal_space(s1, phi1, phi2)
    v1_s2, v2_s2 = signal_space(s2, phi1, phi2)
    results = {}

    for db in eb_sigma2_db_list:
        sigma2 = Es / (10 ** (db / 10))
        r1_points = []
        r2_points = []

        for _ in range(num_samples):
```

```
            r1 = add_awgn(s1, sigma2)
            r2 = add_awgn(s2, sigma2)
            v1_r1, v2_r1 = signal_space(r1, phi1, phi2)
            v1_r2, v2_r2 = signal_space(r2, phi1, phi2)
            r1_points.append((v1_r1, v2_r1))
            r2_points.append((v1_r2, v2_r2))

        results[db] = (np.array(r1_points), np.array(r2_points))

        # Plot
        plt.figure()
        # Plotting the noisy signal space
        plt.scatter(*zip(*r1_points), label="r1 (s1 + noise)", alpha=0.7, color="blue")
        plt.scatter(*zip(*r2_points), label="r2 (s2 + noise)", alpha=0.7, color="red")
        # Plotting the ideal signal space
        plt.plot(v1_s1, v2_s1, "o", label="s1 ideal", markersize=10, color="indigo")
        plt.plot(v1_s2, v2_s2, "o", label="s2 ideal", markersize=10, color="crimson")

        plt.xlabel("v1")
        plt.ylabel("v2")
        plt.title(f"AWGN Signal Space ($E/\\sigma^2$ = {db} dB)")
        plt.grid(True)
        plt.legend()
        plt.tight_layout()
        plt.savefig(f"noisy_signal_space_{db}dB.png")
        plt.close()

    return results
```

Listing 6: Runs AWGN as described above in the begining of part 3

```
if __name__ == "__main__":
    # NOTE: our time space 1 is 100 samples for simplicity

    t = np.linspace(0, 1, NUMBER_SAMPLES)

    # signal one is all ones
    s1 = np.ones(NUMBER_SAMPLES)

    # signal two is 75 ones and 25 -1
    s2 = np.concatenate(
        [np.ones(int(0.75 * NUMBER_SAMPLES)), -1 * np.ones(int(0.25 * NUMBER_SAMPLES))]
    )


    plot_signal(t, s1, "Signal s1(t)", "signal_s1.png")
    plot_signal(t, s2, "Signal s2(t)", "signal_s2.png")

    phi1, phi2 = gm_bases(s1, s2)
    phi1 = phi1 * scale
    phi2 = phi2 * scale

    plot_signal(t, phi1, "Basis Function  1 (t)", "basis_phi1.png")
    plot_signal(t, phi2, "Basis Function  2 (t)", "basis_phi2.png")

    # Signal space representation
    v1_s1, v2_s1 = signal_space(s1, phi1, phi2)
    v1_s2, v2_s2 = signal_space(s2, phi1, phi2)
    plt.figure()

    plt.plot(v1_s1, v2_s1, "o", label="s1", markersize=10, color="indigo")
    plt.plot(v1_s2, v2_s2, "o", label="s2", markersize=10, color="crimson")

    plt.plot([0, v1_s1], [0, v2_s1], "-", color="indigo", label="Vector to s1")
    plt.plot([0, v1_s2], [0, v2_s2], "-", color="crimson", label="Vector to s2")


    plt.xlabel("v1")
    plt.ylabel("v2")
    plt.title("Signal Space Representation")
    plt.grid(True)
    plt.legend()
    plt.tight_layout()
    plt.savefig("signal_space_clean.png")
```

```
    plt.close()

    run_awgn_experiment(s1, s2, phi1, phi2)
```

Listing 7: Main functions calles function above and plot the signales