Dept. of Electronics and Electrical Communication Engineering
Indian Institute of Technology Kharagpur

DIGITAL COMMUNICATION LAB (EC39001)

Author: Aryan Satpathy

Roll Number: 20EC10014

Group Number: 10
Experiment Number: 8

TA Instructor:

Experiment Number: 8
Title Of Experiment: QPSK Modulation
Date Of Submission: April 4, 2023

Introduction

This experiment in purely **MATLAB** based. The experiment intends to demonstrate Quadrature Phase Shift Key(QPSK) Modulation, specifically its performance in the presence of Noise.

Key Objectives

For this experiment, out key objectives are:

- Generating a PN Sequence.
- Modulating the PN Sequence using QPSK Modulation Scheme.
- Adding Gaussian Noise on the Signal.
- Demodulating it and comparing the Bit Error Rate with Expected/Analytical Expression.

Results

We begin by first generating the PN Sequence of 6th degree. This can be done by using the built in module in **Communications Toolbox**. We use the following Polynomial for this purpose:

$$\mathrm{x}^6 + \mathrm{x}^5 + 1 = 0$$

Let us have a look at how the PN Sequence looks.

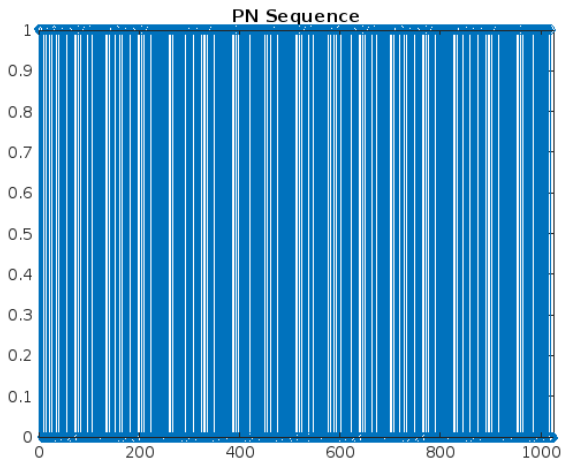


Figure 1: PN Sequence

With the PN Sequence done, we need to take 2 points in the Sequence at a time, and perform a QPSK Modulation.

$$\text{signal}_i = \left(\text{PN Sequence}_{2i}, \text{PN Sequence}_{2i+1}\right)$$

And the signal is Modulated into QPSK(2 Dimensional Signal) as follows:

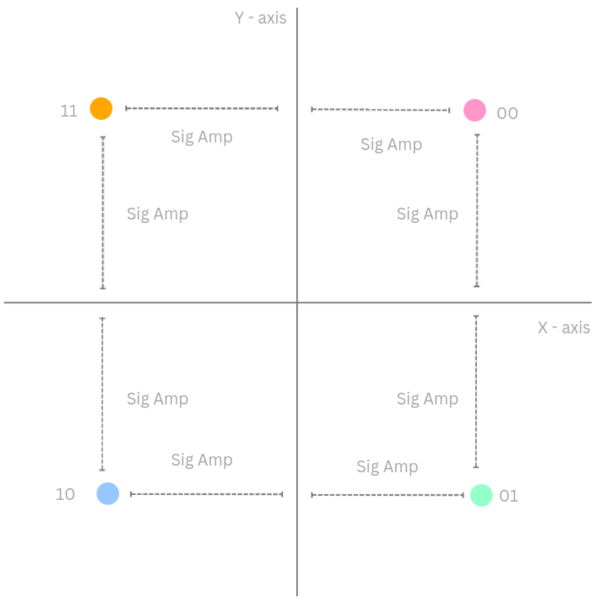
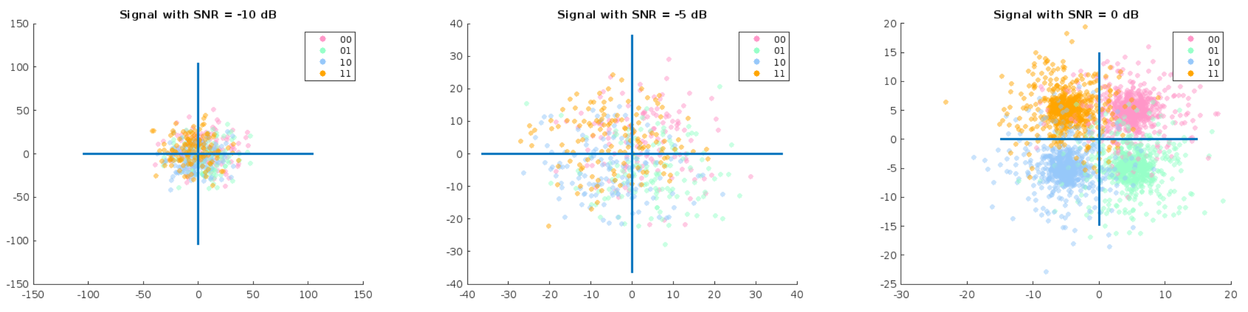


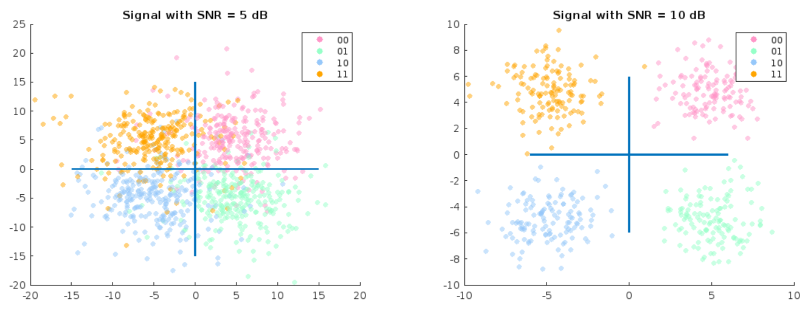
Figure 2: QPSK Modulation

We take *Sig Amp* = 5.

We will now add Gaussian Noise on the Modulated Signal. The Noise will be generated using *randn* function in MATLAB. Let us have a look at Scatter Plots when noise acts on the Modulated Signal.



(a) SNR = -10 dB (b) SNR = -5 dB (c) SNR = 0 dB



(d) SNR = 5 dB (e) SNR = 10 dB

Figure 3: Scatter Plots demonstrating the action of Noise

We can clearly notice how some **Messsages bleed into the Decision Region of other Constellations**(Orange bleeding into the Decision Region of Green for example). Its pretty basic theory that **Decision Regions in case of QPSK Modulation is just the Quadrants**. This is of course under the assumption that the **Constellations are equi-probable**. This is the **MAP Detector Strategy**. This gives us a Qualitative view on how the Error in Demodulation will be related to the Noise(SNR). Let us now have a quantitative look at the Bit Error Rate and its relation with SNR.

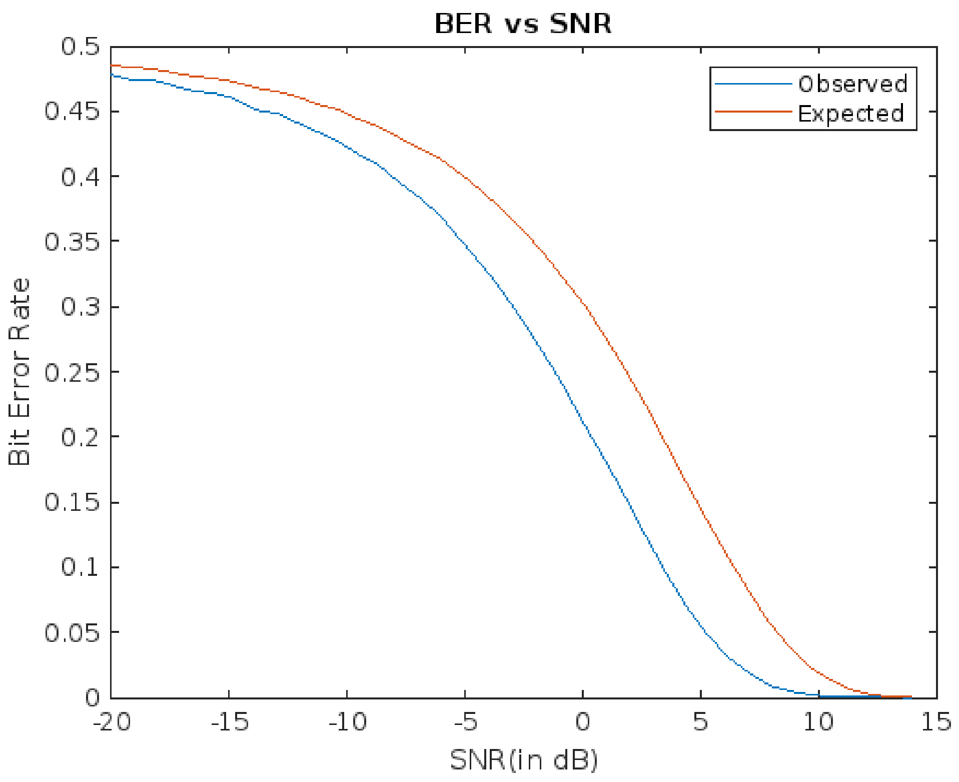


Figure 4: Bit Error Rate vs Signal to Noise Ratio(SNR)

As we can see, the plot of BER v/s SNR more of less matches with the Analytical Expression. The Analytical Expression here is:

$$\text{BER} = \text{Probability of Error} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{-x} e^{-t^2}dt$$

Where x is the square root of SNR.
Let us define Q as Probability of Bit error in the following setting. Consider **n** as a Normal Distribution $N(0,1)$. It will have a

$$\text{Probability Density Function(pdf)} = \frac{1}{\sqrt{2\pi}}e^{-x^2}$$

Now consider a case where you are adding this Normal Distribution on the value *+1*. *+1* will be decoded wrongly as *-1* if it cross the Origin(*0*). In order for that to happen, **n** has to be smaller than *-1*. Probability of that happening is:

$$\text{P(n < -1)} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{-1} e^{-t^2}dt$$

Now if we generalize this for our case, we have to put *variance(σ)* and *Sig Amp* in the expression as follows:

$$\text{P(error)} = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{-\text{Sig Amp}} e^{-\frac{t^2}{\sigma^2}}dt$$

Substitute t $= \frac{t}{\sigma}$ to get the familiar result:

$$\text{Q} = \text{P(error)} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{-\sqrt{SNR}} e^{-t^2}dt$$

As SNR $= \frac{\text{Sig Amp}}{\sigma}$. So

$$\begin{aligned}\text{P(Bit Error)} &= 0.5\Big[\text{P}(\text{n}_x\text{>-Sig Amp}).\text{P}(\text{n}_y\text{<-Sig Amp})\Big]\\ &+ 0.5\Big[\text{P}(\text{n}_x\text{<-Sig Amp}).\text{P}(\text{n}_y\text{<-Sig Amp})\Big]\\ &+ 1\Big[\text{P}(\text{n}_x\text{<-Sig Amp}).\text{P}(\text{n}_y\text{>-Sig Amp})\Big]\end{aligned}$$

The coefficients *0.5 & 1* come from the Hamming Distance. If both n_x and n_y are less than *-Sig Amp*, 00 will be decoded as 11. This means 2 bit errors. So Bit error Probability/Rate = 2/2 = 1. Similar arguments are to be made for 00 to be decoded as 01 and 10 which gives Bit Error rate of 0.5. So the result simplifies as:

$$\text{P(Bit Error)} = 1.5\ Q - Q^2$$

Discussion

Answering the questions given in Lab Manual:

- **Analytical Expression for Bit Error Rate**

  The Analytical Expression for Bit Error Rate has already been presented and derived in detail in the Results section. So I will just mention whether the Multi-Run Sampling helped in smoothing out the curve.

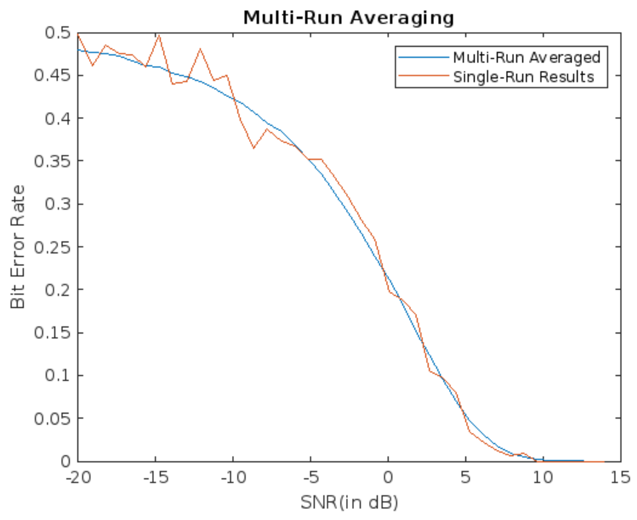
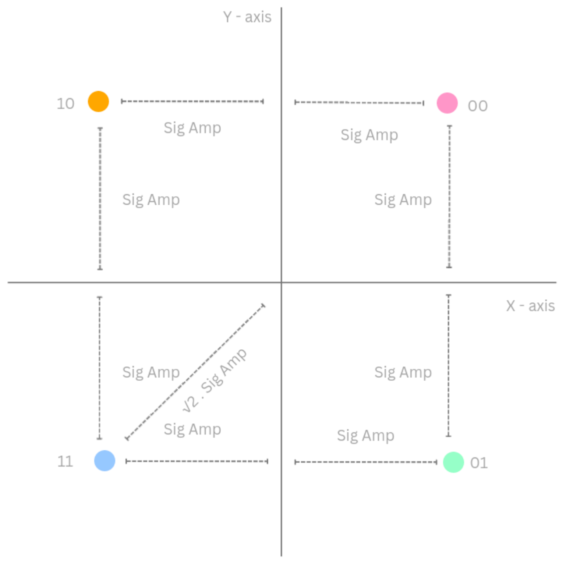


Figure 5: Single-Run and Multi-Run Averaged

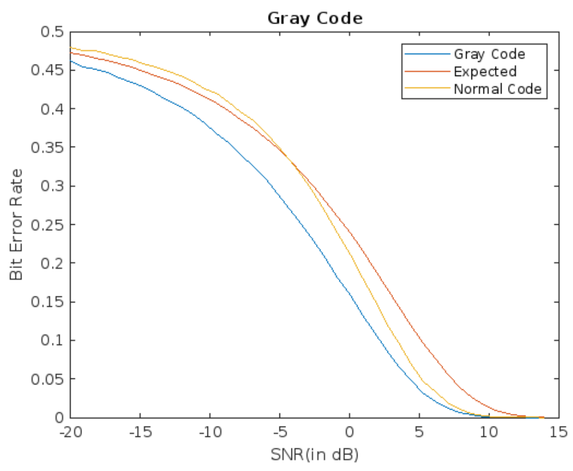
  As we can see, **Single-Run graph** is very noisy and cannot be used to compare with the Analytical Expression. Therefore the **Multi-Run Averaged Graph** does help with smoothing out the curve and making it comparable to other curves.

- **Gray Code vs Normal Code**

  It is intuitive to expect the Gray Code Modulation to perform better. It is because, the Constellations have more Hamming Distance are kept further apart. If they are kept closer, more number of bit flips(or error) can occur with greater probability. It will be clear with the illustration.



(a) Gray Coded Modulation



(b) BER for Gray Code v/s Normal

Figure 6: Gray Code

This is indeed observed in the Experiment(see 6b). As we can see, Gray Code actually is always better than the Analytical Expression for Normal Code(in all the range of SNR plotted in the graph, it is not relevant to look at the plots beyond this range of SNR as the differences become so less that they can be easily mistaken with Chance or even floating point error).

Let us derive the Analyical Expression for Bit Error Rate in case of Gray Code. We will use the similar analogy as we did for Normal Code.

$$\text{P(Bit Error)} = 2 \text{ x } 0.5\Big[\text{P(n}_x\text{<-Sig Amp) - P(n}_x\text{<-Sig Amp)P(n}_y\text{<-Sig Amp)}\Big] \\ + 1 \text{ P(n}_x\text{<-Sig Amp)P(n}_y\text{<-Sig Amp)}$$

$$\text{OR P(Bit Error)} = \text{Q}$$

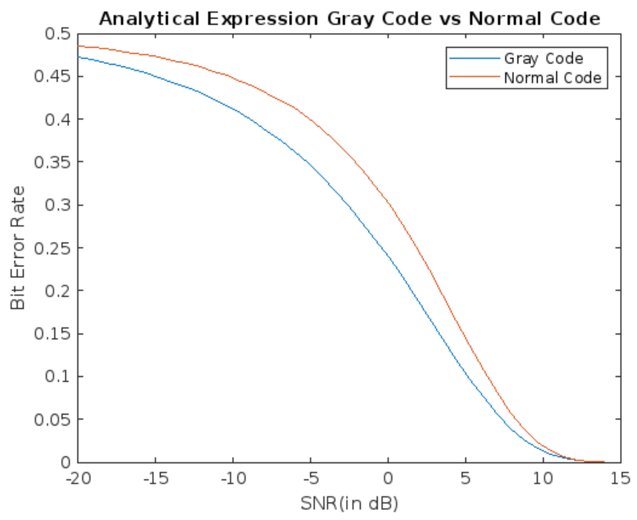


Figure 7: Gray Code v/s Normal Code

As we can tell, the Analytical Expression itself is better for Gray Code. This supports the intuition.

- **An Interesting Observation**

  Its worth noting that for both the cases(Normal and Gray Code), the Observed BERs follow the pattern of Analytical Expressions. However, they are less than the Analytical Expressions. This is because we make a lot of assumptions about a lot of things. First off, we assume that n$_x$ and n$_y$ are independent. They are indeed independent but they are generated by MATLAB's Pseudo Random Generator, which doesn't guarantee **complete randomness**, therefore doesn't guarantee independence either. Once again, the Analytical Expressions are an **Expectation of Bit Error Rates**. Since number of samples are finite, they cannot follow an Expectation perfectly.

- **How To Generate Noise?**

  Noise can be generated in 3 ways:

  - MATLAB's *awgn()* on the 2D Modulated Array
  - MATLAB's *awgn()* on 1D array multiplied with e$^{j\theta}$ where θ is uniform random variable between 0 and 2π. Then adding Real part to first axis of 2D Modulated Array and Imaginary part to the second axis of 2D Modulated Array.
  - Two separate MATLAB's *randn* to generate n$_x$ and n$_y$.

  All the above approaches produce Gaussian Noise however the BERs are very different for all of the approaches. This is due to the fact that these are indeed different distributions. For example, 2nd Approach would add the restriction

  $$\mathrm{n}_x^2 + \mathrm{n}_x^2 = \mathrm{r}^2$$

  Where *r* is a Gaussian Random Variable. As we can tell, n$_x$ and n$_y$ are not independent. However, Approach 1 should have worked. I have no idea why it doesn't work. I suspect its the Seed. MATLAB would use the same Seed to generate noise for both axes which might not make them independent anymore(Rather less independent)? Anyways, I tried all of the above 3 approaches. MATLAB's *randn* approach is the only one that works well. Approach 1 also works but deviates a lot from Analytical Expression. In fact is reaches BER = 0 for SNR values near 0 dB. It is evident that there is less Randomness in this approach. Therefore, I go ahead with Approach 2.

  Please generate all the plots in order to get high resolution view. Link to the code: Drive Link to the Code

Conclusion

In this experiment, we coded Viterbi's Algorithm and observed the Bit Error Rates for Binary Symmetric and Asymmetric Channels.

References

- MATLAB Official Documentation
- Wikipedia
- ChatGPT
- Lab Manual