C Tech Information Technologies Inc.

Summer Internship

24/06/2024 - 28/06/2024

Week 1 Report:

28/06/2024

Fatih Mehmet Çetin

Purpose:

The purpose of this document is to provide an overview of my first week's activities during my internship. I aim to document the tasks I was responsible with, the skills I learned, and my experiences in my first week in the company.

Introduction:

I work with the “Communication System Design” department within the company. The department consists of 4 engineers and myself as an intern. This week mainly passed with adaptation to the office environment and getting to know what my department was doing.

Work Done:

The very first task I was assigned to was to study some fundamental concepts related to signals and systems. My supervisor tasked me with learning each step of the following block diagram **(Figure 1.1)** of a typical digital communication system.

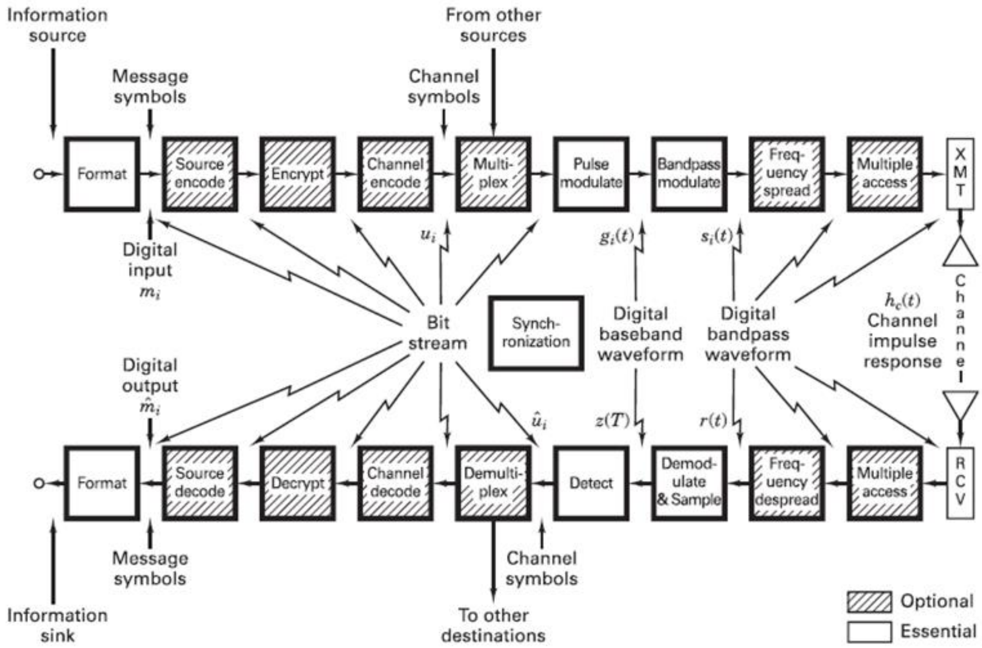

Figure 1.1: Generalized block diagram of a digital communication system.

After I learned each step of the diagram, I was assigned to write some Matlab codes (bpsk.m, qpsk.m, opsk.m, spsk.m) regarding M-ary Phase Shift Keying techniques **(Appendices 1 to 5)**. I did this task by taking the bit info and the phase offset as inputs and giving a complex number as an output. This task helped me to understand better about different M-ary PSK types.

After coding 4 different modulation types, I was assigned with creating demodulator modules in Matlab for each one (debpsk.m, deqpsk.m, deobpsk.m, despsk.m)**(Appendices 6 to 10).** I achieved this task by taking the phase offset and a complex number as an input and giving the bits as output. After writing 4 modulator and 4 demodulator modules in matlab. I tested whether the code I've written can fully recover the bits I gave as input to the modulator. After testing, I found out that my code can successfully modulate bits and then recover them with 100% accuracy.

After creating 4 different modulator-demodulator couples, I was assigned with creating BER (Bit-Error-Rate) - SNR (Signal to Noise Ratio) graphs for noised signals. In order to accomplish this task, I first modulated signals and then by using a built-in function in Matlab, I added white Gaussian noise to the modulated signals by changing the SNR each time. Then I used my already existing demodulators to demodulate the noisy signals. Here you can see the relationship between different M-ary PSK types and their BER/SNR relation **(Figure 1.2) (Appendix 11)**.

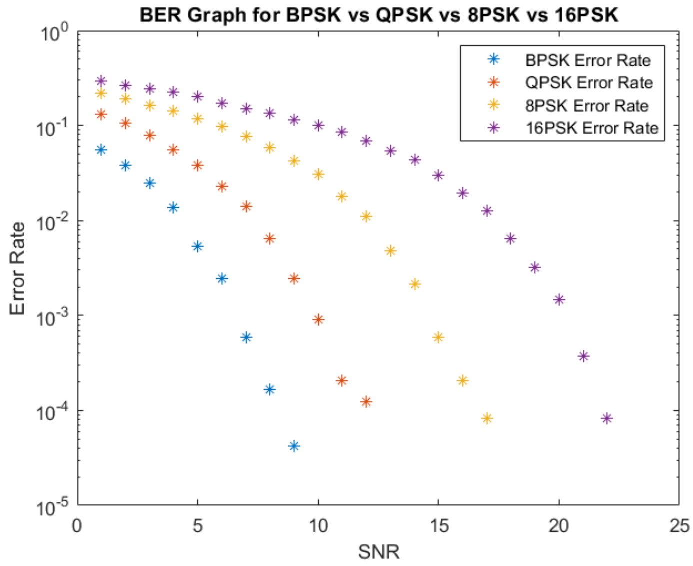

Figure 1.2: Different M-ary PSK types and their BER/SNR ratio in a logarithmic y-axis

Then, my supervisor tasked me with learning what Eb/n0 ratio (energy per bit to noise power spectral density ratio) was and to transform the graph in **Figure 1.2** to a BER/(Eb/n0) graph. I then successfully transformed the same process but this time instead of SNR, I plotted bit-error-rate to Eb/n0. Here you can see my results **(Figure 1.3) (Appendix 12)**:

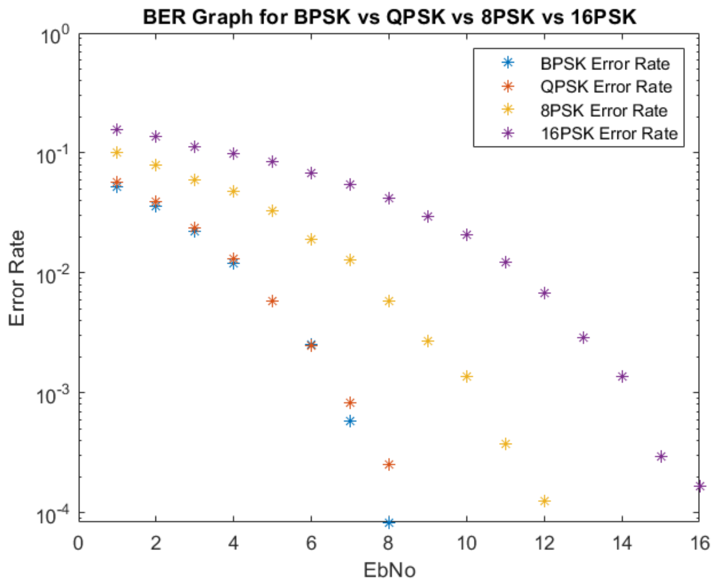

Figure 1.3: Different M-ary PSK types and their BER/(Eb/n0) ratio in a logarithmic y-axis

From **Figure 1.2,** we can state that using a lower M value in M-ary PSK, results in a more accurate transmission and less susceptibility to bit errors. Nevertheless, the bit rate you can transmit is limited drastically since you can transmit less bits per symbol. From **Figure**

1.3, we can conclude that a higher M value generally results in faster but more disruptable communication. However, what was different from the BER/SNR graph was that this time BPSK and QPSK almost got the same BER per Eb/n0 ratio. This means that even though bit error rate is smaller in BPSK per SNR, it requires more energy to transmit a slow transmission, therefore if you scale the bit-error-rate per Eb/n0, BPSK and QPSK would almost do the same job.

The next task I was committed was to learn what repetition code was and to simulate the same BER/SNR and BER/Ebn0 graphs for repetition codes with different repetition factors. To do this task, I've first written 3 modules in Matlab that repeats the bits with different repetition factors (3, 5 and 7**) (Appendices 13 to 15).** After this, I created 3 other modules that this time returns the original bits when given a repetitive code as input-reverse repeaters **(Appendices 16 to 18)**. Then I've written a test module to simulate the BER/SNR relation by only using QPSK but with different repetition factors **(Appendix 19)**. Here was my results **(Figure 1.4)**:

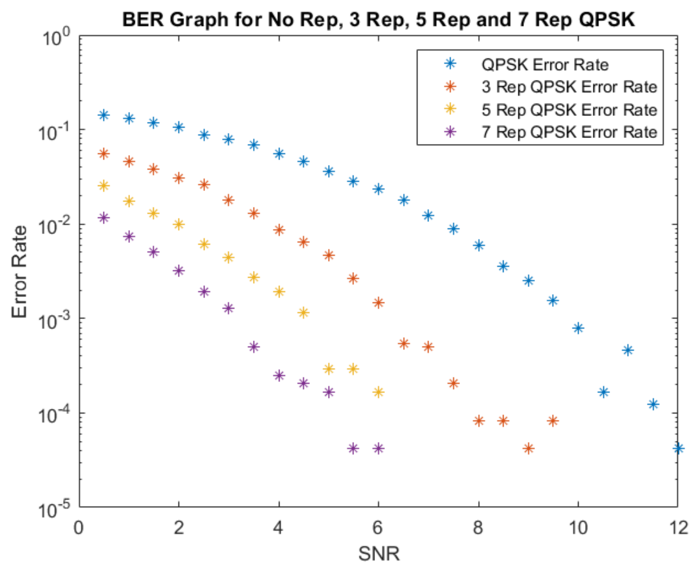

Figure 1.4: BER/SNR relation for different repetition factors by using QPSK and deQPSK

After obtaining **Figure 1.4,** I then did the same operation but this time I plotted the bit-error-rate to Eb/n0 **(Appendix 20)**. Here you can see my results **(Figure 1.5)**:

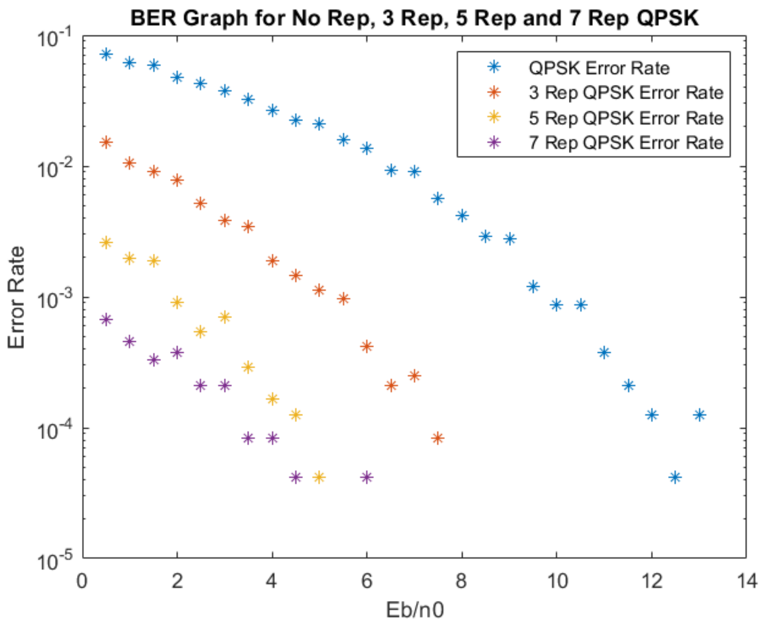

Figure 1.5: BER/(Eb/n0) relation for different repetition factors by using QPSK and deQPSK

From **Figure 1.4,** we can conclude that using a higher repetition factor has significantly reduced the bit-error-rate, but this meant that we would transmit the same signal in a longer time interval because we had 7 times more data to transfer. From **Figure 1.5,** we can similarly say that reducing the repetition factor led to a faster communication but a more inaccurate one.

At this point of the work I've done, I also committed myself to do a simulation of something I wondered. During the very beginning of the week, I coded modulator and demodulators for M-ary PSK signalling. One thing my supervisor advised me to do was to map the bits with gray-mapping technique. This technique would lead to a more complex mapping but resulted with a less error rate. I then coded a modulator-demodulator couple for QPSK but with binary mapping **(Appendices 21&22)**. Then I simulated the BER/SNR value for two QPSK models with different mapping techniques **(Appendix 23)**. Here are my results **(Figure 1.6)**:

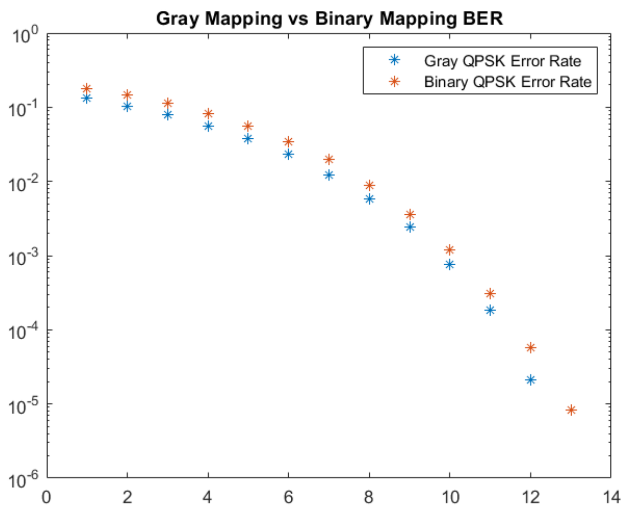

Figure 1.6: BER/SNR relation for two QPSK models with different mapping styles

From **Figures 1.4&1.5**, we had some information on the effects of repetition ratio and the M value in M-ary PSK types on the BER graphs. I then wondered whether the repetition ratio or the M value in M-ary PSK types mattered more in BER graphs. Then I tested the same previous procedure but this time I compared 16 different quantized graphs (BPSK, QPSK, 8PSK, 16PSK) –4 different M values & 4 different repetition ratios (1,3,5,7). Here are my code **(Appendices 24&25)** and my results **(Figures 1.7&1.8)**.

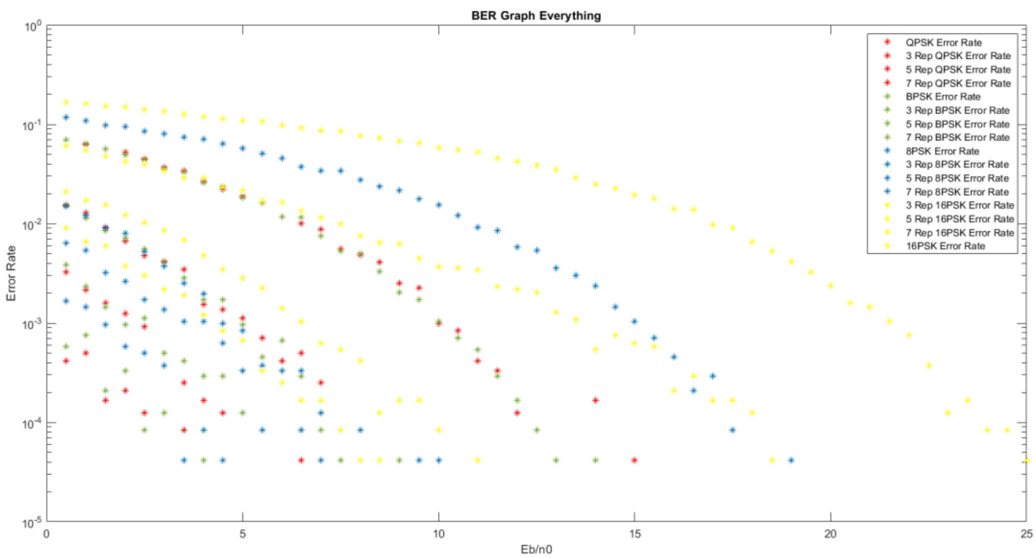

Figure 1.7: BER/(Eb/n0) relation for 4 PSK models with 4 different repetition ratios

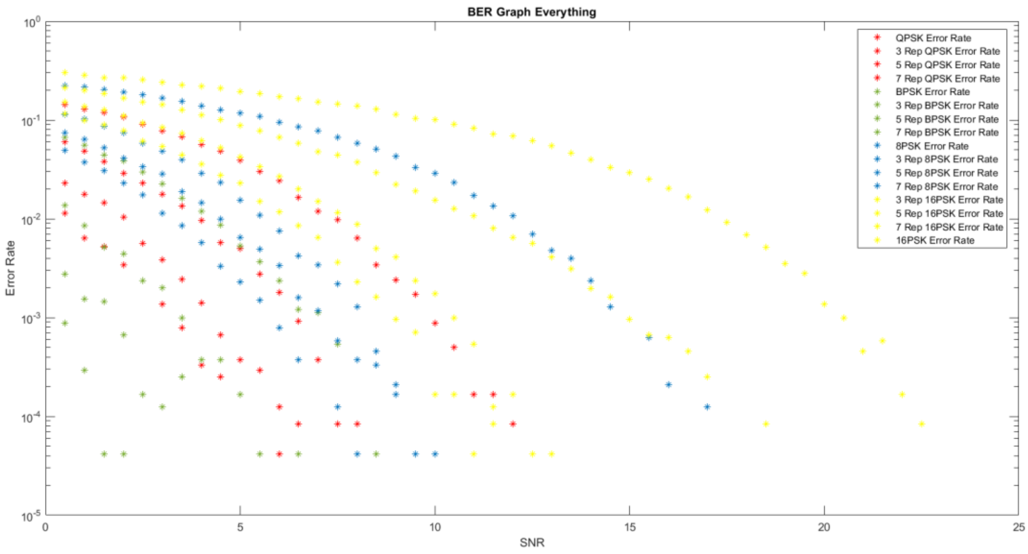

Figure 1.7: BER/SNR relation for 4 PSK models with 4 different repetition ratios

We can conclude that there was no significant dominance in terms of effecting the BER ratio from different M values and different repetition ratios. Nevertheless, we can observe the foundings we did earlier such as the similarity of QPSK and BPSK in Eb/n0 graphs; the relationship between PSK types and bit-error-rate in both graphs.

Conclusion:

My first week at the company was mainly focused on the essentials and fundamentals of digital communication systems and signals. I got used to the office environment and also improved my Matlab skills. I feel more confident in testing out or simulating data in Matlab now. For the upcoming weeks, I aim to dig deeper into the digital communication systems and software simulation tools.

Appendices:

1. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/bpsk.m
2. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/qpsk.m
3. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/opsk.m
4. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/spsk.m
5. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/debpsk.m
6. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/modulation.m
7. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/deqpsk.m
8. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/deopsk.m

9. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/despsk.m
10. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/demodulation.m
11. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/snr.m
12. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/Ebn0.m
13. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/modrep3.m
14. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/modrep5.m
15. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/modrep7.m
16. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/demodrep3.m
17. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/demodrep5.m
18. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/demodrep7.m
19. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/snrrepetition.m
20. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/enbn0repetition.m
21. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/qqpsk.m
22. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/deqqpsk.m
23. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/binaryvsgrayqpsk.m
24. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/bersnr16.m
25. https://github.com/fmcetin7/CTech-Internship/blob/main/matlab%20codes/berebn016.m