C Tech Information Technologies Inc.

Summer Internship

01/07/2024 - 05/07/2024

Week 2 Report:

05/07/2024

Fatih Mehmet Çetin

Purpose:

The purpose of this document is to provide an overview of my second 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 second week in the company.

Introduction:

I intern at the “Communications System Design” department within the company. The department consists of 4 engineers and myself as an intern. My second week has passed with learning and implementing more improved digital communication terms such as different FEC types, filtering, sampling and CRC algorithms in Matlab.

Work Done:

The very first task I committed myself during my second week was getting to know what Forward Error Correction (FEC) and its types were. I already had some initial ideas about FEC from my first week where I coded repetition algorithms. I then coded several FEC algorithms as well as the algorithms that solve the FEC **(Appendices 1 to 16)**. These algorithms include convolution encoding, cyclic encoding, linear encoding and hamming encoding and their relevant decoding algorithms.

I then coded a 24-bit CRC algorithm and a decoder for that algorithm **(Appendices 17&18)**. I would use these two modules to check the error rate of some bit packages I simulated to compare different FEC types. My goal was to split long databits into packages and by adding a header file which contained information about the modulation and FEC type that package contained. I used CRC algorithm instead of adding the –xor result of each transmitted and received bit because I wanted to simulate the real-life situation where the receiver does not access the transmitted bits.

After some work I finally managed to create a test module **(Appendix 20)** that would give an errorrate as output, by taking the databits, modulation&FEC type, package number and SNR value tolerance and SNR maximum value as input. What this module did was basically to give an errorrate vector by changing SNR values. This module provided me a simulation environment where I was able to simulate and compare results of errorrates of different modulation and FEC types.

There are more graphs I acquired in my github page **(Appendix 19)**, I will put only some of them to this report in order to both give the reader a good sense of understanding of the work I conducted as well as not making this report full of repeated data. For instance, here you can see the package error rate for QPSK modulated symbols which one of them had a convolutional encoder and the other had a repetition encoder with same code rates **(Figure 2.1)**. **Figure 2.2** contains the same graph with more FEC types involved. Nevertheless, it is important to note that **Figure 2.1** is more meaningful because the two compared FEC's share the same code rate.

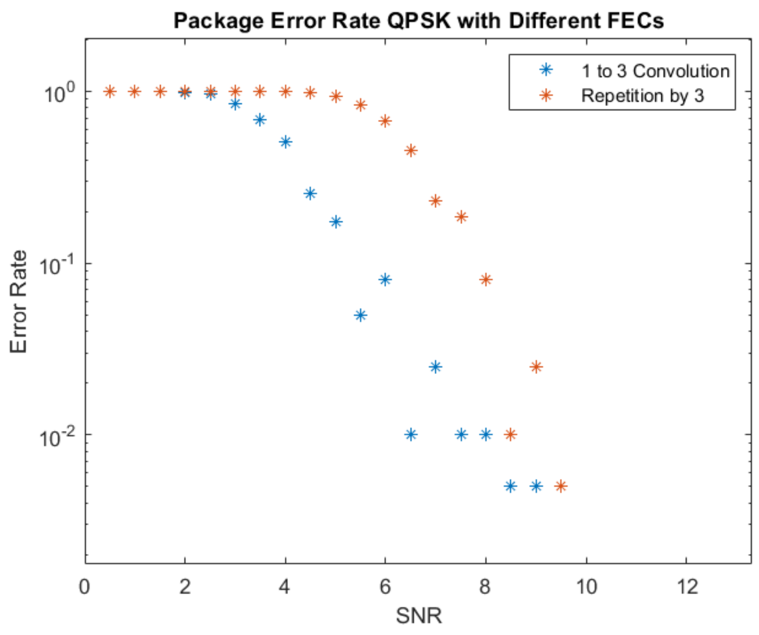

Figure 2.1: Package Error Rate per SNR for QPSK with 2 different FEC Types

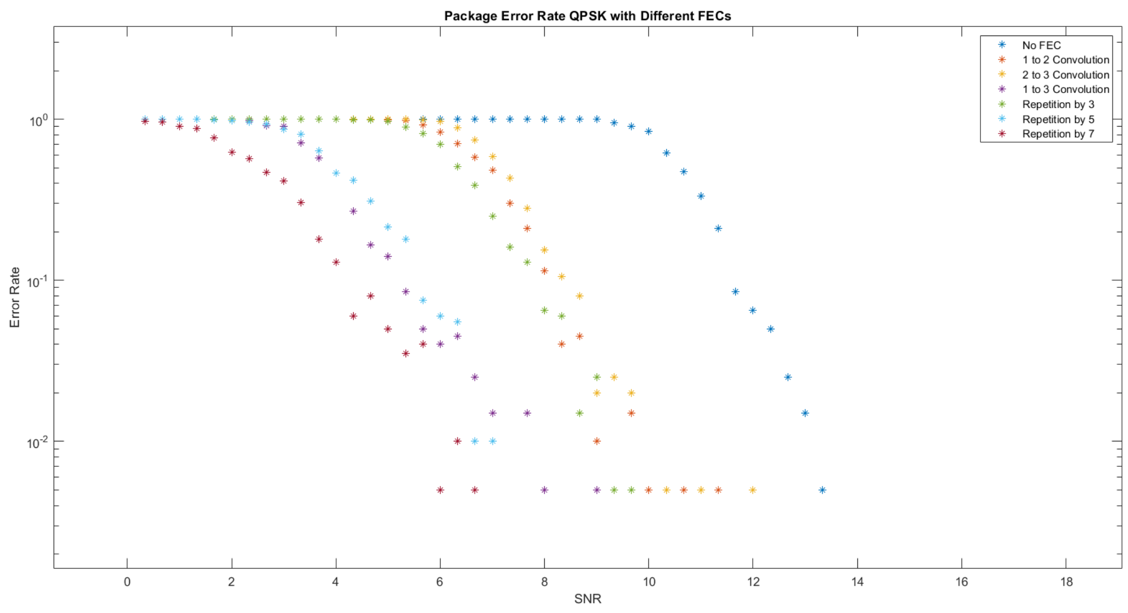

Figure 2.2: Package Error Rate per SNR for QPSK with different FEC Types

Then I learned how to fit curves into discrete datasets in Matlab. I used curveFitter to fit a curve to the plot where I compared 1:3 convolution with 1:3 repetition. You can see the fitted curve graph both in normal scale and logarithmic scale for y-axis **(Appendix 21); (Figures 2.3 & 2.4)**.

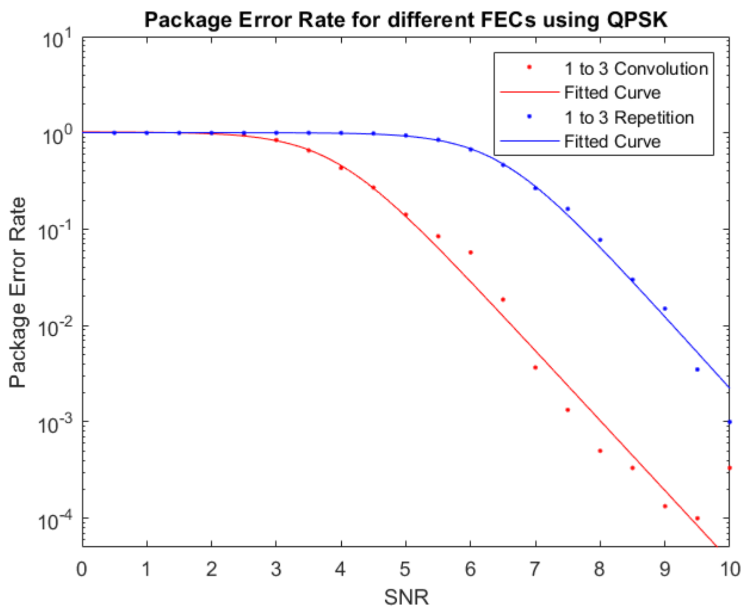

Figure 2.3: Package Error Rate per SNR for QPSK with 2 different FEC Types with a Fitted Curve and a logarithmic scale on y-axis

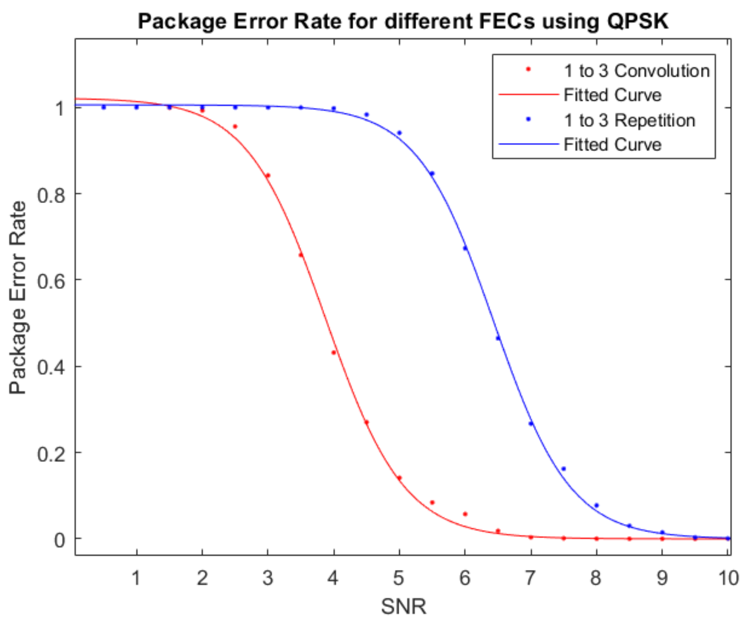

Figure 2.4: Package Error Rate per SNR for QPSK with 2 different FEC Types with a Fitted Curve and a normal scale on y-axis

After this, I learned about RRC filtering, matched filters, upsampling and downsampling. I studied several books and watched videos on these topics. Then I tried to learn how to implement these filters on Matlab. I used Matlab tools to understand more about these filters and the parameters they involved. I spent almost 2 days on testing out the functions and trying to understand the parameters.

Then I created a plot where I compared different SNR values and different SPS (Sample per Symbol) values **(Figure 2.5)**.

In the plot; yellow dots correspond to upsampled QPSK symbols, orange dots are basically the yellow dots with some additive white Gaussian noise. Blue dots are the output when orange dots are passed through a matched filter. Finally, the purple dots correspond to the downsampled blue dots, also they are the input symbols of the final demodulation that will take place. The block diagram for the entire process looks like this:

Modulate – Upsample – Matched Filter – AWGN – Matched Filter – Downsample –Demodulate

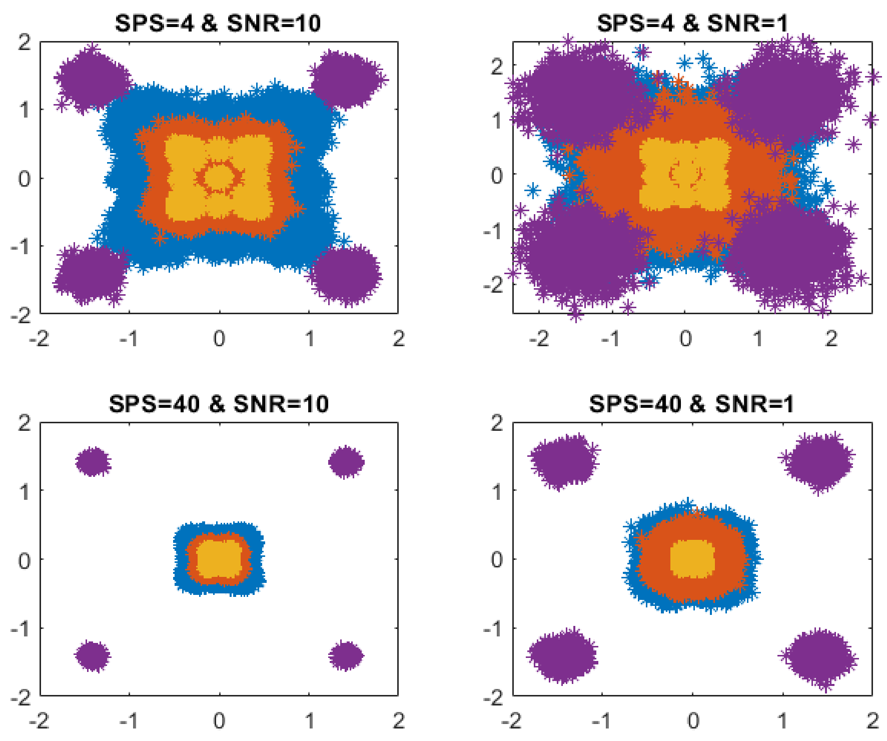

Figure 2.4: The Plot which compares different SNR & SPS values

From the graph we can say that as SPS goes up, two things happen. First, the yellow dots close down on a smaller area which means the same QPSK symbols that undergo an upsampling operation gets transformed into a more similar data. Secondly, the difference between blue and orange dots gets significantly down. This means the outputs of the matched filter is also transformed into a more similar dataset.

On the other hand, when SNR goes up, this affects the difference between the orange and yellow dots. This makes sense since the AWGN function takes SNR as an input, the noised signal becomes affected from the SNR.

Conclusion:

The second week at the company had passed with learning and implementing more improved digital communication terms such as different FEC types, filtering, sampling and CRC algorithms in Matlab. I believe I got better at Matlab if I compare my skills to the previous week. This week did not only contribute to my software simulation tool skills, but also, I learned significant theoretical information on signals and systems which will help me when I take the course in Bilkent.

Appendices:

1. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/conv1to2.m
2. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/conv1to3.m
3. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/conv2to3.m
4. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/convolution.m
5. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/cyclicenc.m
6. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/hamming.m
7. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/linearenc.m
8. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/repetition.m
9. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/deconv1to2.m
10. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/deconv1to3.m
11. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/deconv2to3.m
12. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/deconvolution.m
13. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/decyclic.m
14. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/dehamming.m
15. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/delinearenc.m
16. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/derepetition.m
17. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/crc.m
18. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/checkcrc24.m
19. https://github.com/fmcetin7/CTech-Internship/tree/main/week2graphs
20. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/test.m
21. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/FittedCurve.m
22. https://github.com/fmcetin7/CTech-Internship/blob/main/week2matlabcodes/SNRvSPS.m