*Daire2compaire: 18-758 Final Project*

Raewyn Duvall and Emmanuel Aire-Oaihimire

Electrical and Computer Engineering Department

Carnegie Mellon University

Pittsburgh, PA, USA

rduvall@andrew.cmu.edu; eaireoai@andrew.cmu.edu

*Abstract*—This report contains some findings and realizations made while working on the 18-758 Wireless Communications Final Project. The goal of the project was to design a working digital wireless communications system using the principles taught in the course to transmit and receive a message with a BER < 5%. The project was split into a basic design and an advanced design.

# Basic Design

## Overview

The basic design was split into two portions, “Transmit” and “Receive.” The transmission code constructed the frequency preamble, the timing synchronization, and the alternating pilot signals and sub-messages. Those components were combined into a single signal, convolved with a pulse filter, and transmitted over the USRP. Once the signal was received from the USRP, the receiver code applied a timing recovery, applied a matched filter, then sampled the signal at intervals of . After sampling, the signal was equalized using a one-tap channel and the bits were identified using a minimum distance detector.

|  |  |
| --- | --- |
| Parameter | Length (symbol) |
| Frequency Preamble | 300 |
| Timing Synchronization | 50 |
| Pilot | 20 |
| Sub Message | 160 |

Table 1: Specifications for Transmitted Signal.

None of the MATLAB toolboxes were used to implement any design for this project. Instead, the principles taught in the course were the means of creating all the code for the project [1].

## Method

### Timing Recovery

After receiving the transmitted signal () and allowing for the phase lock loop to settle, the timing recovery was applied to the received signal. The timing recovery correlated the received signal with the timing signal and found where the timing signal ended in the received signal.

### Matched Filtering

Following the timing recovery, matched filtering was applied to each arm (in-phase and quadrature) of the received signal to reduce the noise. The matched filter was the mirror of the original square-root raised cosine filter, .

### Sampling

Sampling was performed on the in-phase and quadrature outputs from the matched filtering at intervals of kT, where .

### Equalization – One Tap Channel

A one-tap channel was used to convert the ISI channel into an equivalent channel, similar to the AWGN channel. For every pilot in the signal, a one-tap channel was applied to equalize the message following the pilot signal. The system makes an estimation of the channel gain, , with at each pilot since the pilot is known to both the receiver and the transmitter.

### Bit Detection – Minimum Distance

The values of are precisely guessed using a minimum distance detector , following the creation of the equivalent channel in the one-tap channel. The in this case was a simple classification of the sign of .

### Bit Error Rate (BER) Calculation

For QPSK (or 4-QAM), the BER can be estimated using the following equation.

For the purposes of this project, we were able to detect the actual transmission by directly solving for the ratio of incorrectly detected bits to total bits sent.

## Results

### Timing Recovery

The timing sync recovery found a correlation of 0.8617 which is considered a good recovery as it is greater than 0.8. There is also only a single correlated moment that has this high a correlation, as seen in Figure 1 of Appendix 1.

### Matched Filtering

The matched filtering made the signal clear with definitive high and low segments, as seen in the partial filtered signal in Figure 2 of Appendix 1.

### Sampling

The sampling was accurate and caught the high and low segments of the signal that could be translated back into bits. Figure 3 of Appendix 1 shows the partial sampled signal.

### Equalization

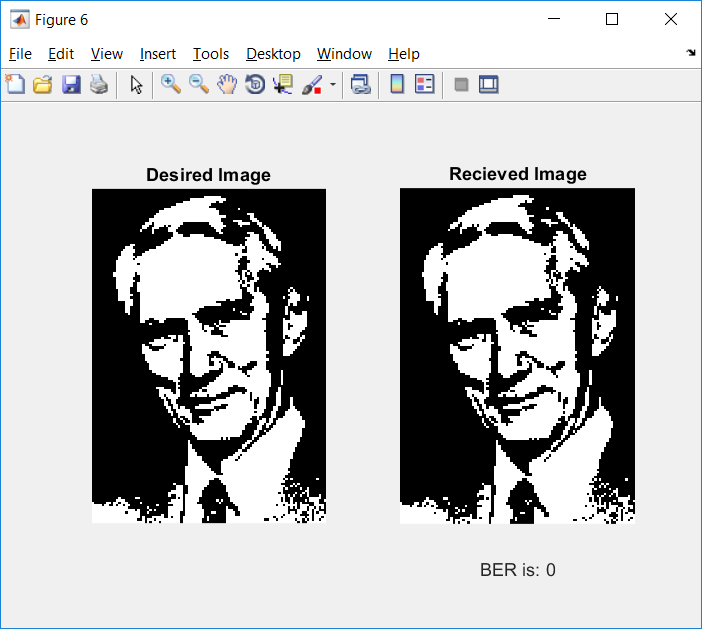
The equalization method eliminated all rotation caused by the transmitter and receiver frequencies being ever so slightly imprecise. The constellation of the final can be seen in Figure 4 of Appendix 1.

### Bit Detection

The bit detection is of such design that it can be proven to act perfectly every time as it does not transform the received data. The determined by equalization, seen in Figure 5 of Appendix 1, is clearly above or below zero.

### Bit Error Rate (BER) Calculation

This design yielded BER = 0 for the 13720-bit image sent over the USRP.



# Advanced Design

## Overview

The advanced design borrowed all fundamentals taught in the course that were used in the basic design, but it was constructed to implement a spread-spectrum system with a spreading gain of 100 and a 3-finger RAKE receiver. Each finger of the RAKE decoded a single multipath component and then the decoded messages from the three fingers were combined to reconstruct the transmission signal [2]. Because the USRP setup does not have multipath, we simulated it by copying the transmit signal 3 times, adding random delay to two, and adding the three signals together to create the signal transmitted over USRP.

Since the signal was to be within the noise floor, spreading was implemented during signal creation by applying a spreading gain number of chips to each transmit symbol. The following parameters were used for the transmission post-spreading.

|  |  |  |
| --- | --- | --- |
| Parameter | Length (symbol) | Length (spread) |
| Frequency Preamble | 4 | 400 |
| Timing Synchronization | 30 | 3000 |
| Pilot | 4 | 400 |
| Message | 106 | 10600 |

Table : Specifications for Transmitted Signal.

## Methods

### Spreading Signal

Spreading the transmit message allows the signal to stay under the noise. Each symbol is multiplied with 100 pseudorandom chips of 1 or -1 value, spreading the power between 100 symbols rather than all the power in 1 symbol.

### Fine Frequency Synchronization

A fine frequency synchronization was used to apply a rotation to the initially received signal, prior to the implementation of the RAKEs. The fine frequency synchronization can be applied to calculate the maximum likelihood (ML) of the frequency offset , since the preamble is known to the receiver. The ML can be computed using the following equations after applying a DTFT to find the peak frequency of the signal.

### Timing Recovery

The timing synchronization was the same as the Basic Design, but the top 3 correlation values were used to obtain the start of each RAKE.

### Matched Filtering

The matched filtering was the same as the Basic Design, repeated for each RAKE.

### Sampling

The sampling was the same as the Basic Design, repeated for each RAKE.

### Equalization

The matched filtering was the same as the Basic Design, repeated for each RAKE.

## Results

### Spreading Signal

As expected, the resulting signal looks like noise as seen in Figure 7 of Appendix 2.

### Fine Frequency Synchronization

The frequency domain of the received signal seen in Figure 8 of Appendix 2 shows how the frequency is not as easily detectable due to being below the noise level. This could contribute to error in the rest of the recovery process.

### Timing Recovery

Despite the frequency being somewhat lost in noise, the timing recovery worked the same as the Basic Design, with three peaks to mark the start of each multipath as seen in Figure 9 of Appendix 2.

### Matched Filtering

The matched filtering worked the same as the Basic Design. Due to the nature of spreading, it looks like noise as seen in Figure 10 of Appendix 2.

### Sampling

The matched filtering worked the same as the Basic Design as seen in Figure 11 of Appendix 2.

# Difficulties in the project

**Basic Design Challenges**

We realized that what we learned in class could be applied directly but the intricacies of MATLAB and its functions were not always evident. For example, when initially building the one-tap channel, we were using the function “*conj”* to find the estimation of channel gain instead of using “*dot”*. This produced many errors, but the issue was eventually resolved. When fixing the code, it was especially useful to create a plot at each step in the process to visualize what errors may have arisen in the design.

We also encountered a problem with a rotating signal (the signal wasn’t settling in the PLL) that couldn’t be properly sampled. As we looked at the signal, we observed two solutions: 1, our frequency preamble was inadequate in length and 2, our symbol period was too large. Upon elongating the frequency preamble, the signal settled; after reducing the symbol period to , we reduced the ISI and could sample properly.

**Advanced Design Challenges**

With our advanced design, we tried to approach the initial construction of the time synchronization, pilot, and message portion of our signal the way we did in the basic design. But we were running into issues with our timing synchronization, pilot detection, and message capturing. When the change was made to drastically increase the size of the time sync to ensure that there was a much higher probability of having an accurate correlation, we also realized that our message size was smaller than the message size from the basic design. We re-distributed the symbols to the timing synchronization, pilot, and message to still fit within the transmission time. This helped to decrease our BER.

# Next steps (desired improvements)

If we were to do this project all over again, or implement additional parts to it, we would like to drastically increase the modularity of the code. We feel that it might help with the presentation and readability of the code, as well as improve the time it takes to run the code. We would also try to run the process on a newer computer with increased performance.

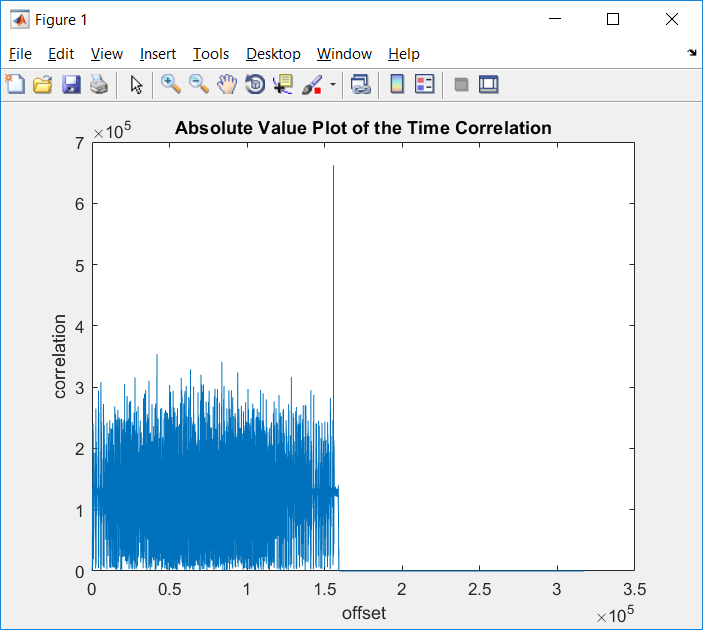
# Acknowledgments

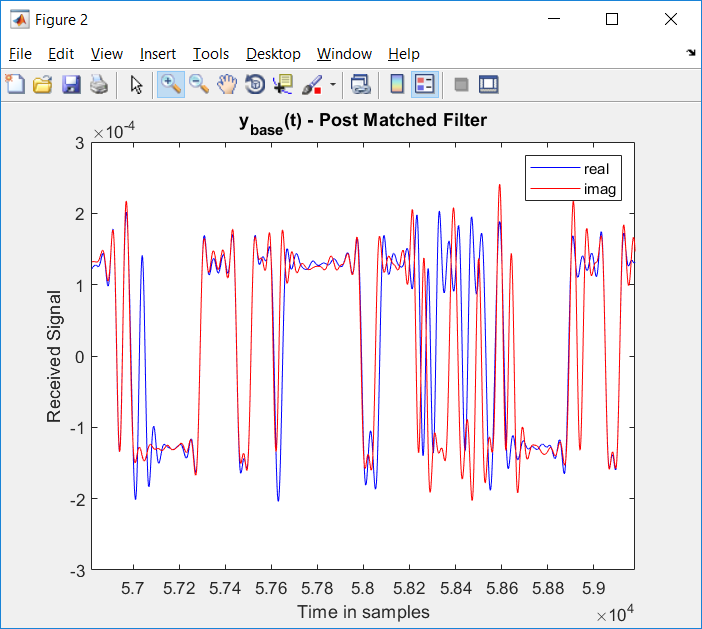
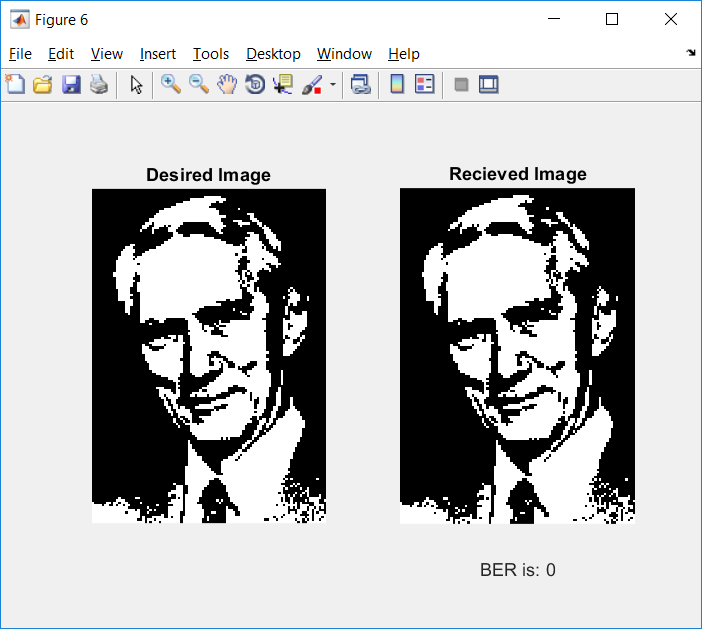
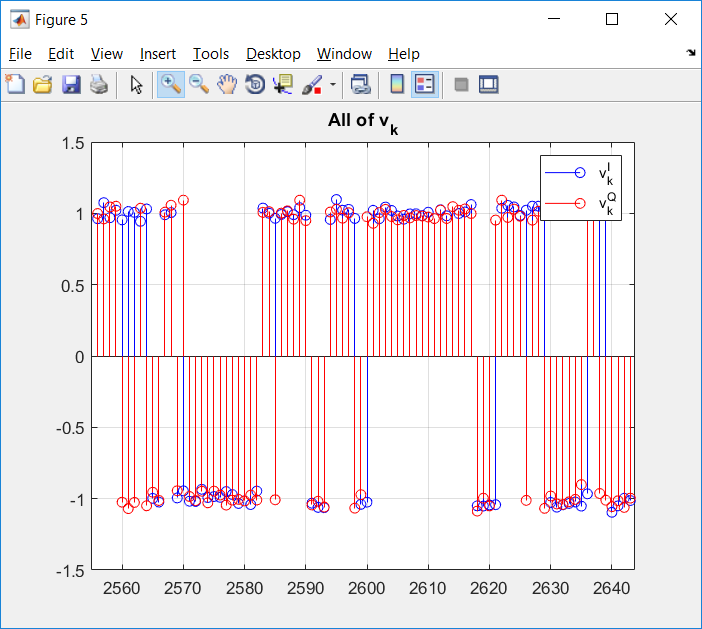
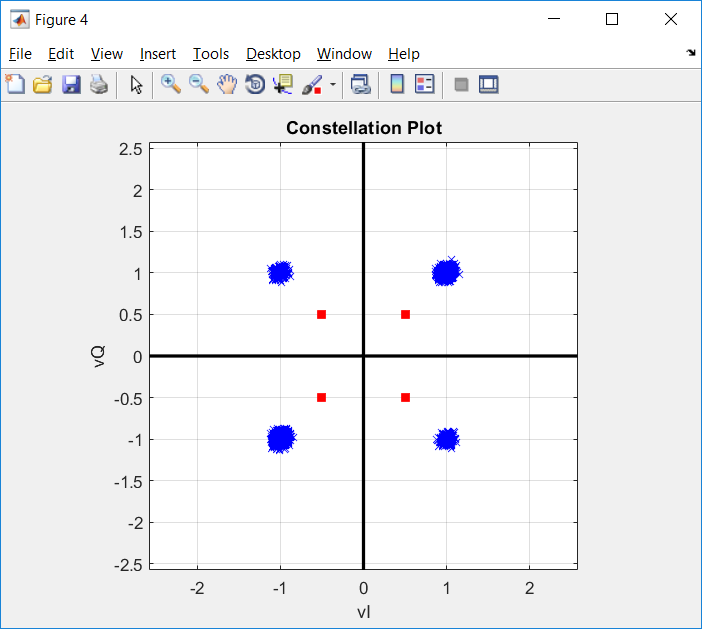
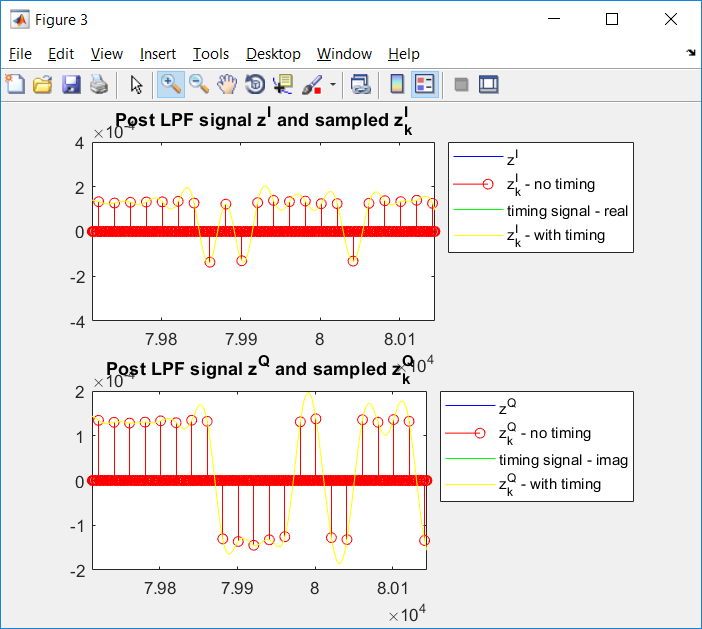
While working on this project we encountered new learning opportunities with each difficulty that we faced. We made efforts to speak with as many different people and consult multiple sources to gain understanding, including: Dr. Rohit Negi, Alireza Chaman Zar, other students in the class (particularly Trevor Rizzolo), and MATLAB documentation.

# References

1. R. Negi, “Chapters 1-15,” in 18-758: Wireless Communications, 2018.
2. Ziemer, R. E.; Tranter, W. H. (August 2001). Principles of Communications: Systems, Modulation, and Noise, 5th Edition. Wiley. ISBN 978-0-471-39253-8.

# Appendix 1 : Basic Design



# Appendix 2 : Advanced Design

