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.**

I. Basic Design

The basic design was split into two portions, "Transmit" and "Receive." The transmission code constructed the frequency preamble, the timing synchronization, the pilot signals and the 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 $t = kT$. 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
Message	160

Table 1: Specifications for Transmitted Signal. There were 86 messages and 87 pilots.

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].

A. Timing Recovery

After receiving the transmitted signal ($y_{base}(t)$) 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 $y(t)$ with the timing signal and found where the timing signal ended in the received signal.

$$C(T,\tau) = \int_{-pT/2}^{pT/2} \left(\sum_{k=-\frac{p}{2}}^{\frac{p}{2}} \overline{x_k p(t-kT)} \right) y(t+\tau)dt$$

B. 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, $p(t)$.

$$z^I(t) = p(-t) * y^I(t)$$
$$z^Q(t) = p(-t) * y^Q(t)$$

C. Sampling

Sampling was performed on the in-phase and quadrature outputs from the matched filtering at intervals of kT, where $kT = f_s * T_\Delta$.

$$z_k = z^I(kT) + j * z^Q(kT)$$

D. 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, h_0, with $\hat{h}_0$ at each pilot since the pilot is known to both the receiver and the transmitter.

$$z_k = h_0 x_k + n_k$$
$$\hat{h}_0 = \frac{\boldsymbol{x}^H \boldsymbol{z}}{\|\boldsymbol{x}\|^2}$$
$$v_k = \frac{1}{h_0}(h_0 x_k + n_k)$$

E. Bit Detection – Minimum Distance

The values of $\hat{x}_k$ are precisely guessed using a minimum distance detector $G(\cdot)$, following the creation of the equivalent channel in the one-tap channel.

$$\hat{x}_k = G(v_k)$$

F. Bit Error Rate (BER) Calculation

For QPSK (or 4-QAM) the following BER equation is used.

$$BER \leq 2Q\left(\frac{1}{\sigma_n\sqrt{2}}\right) = 2Q(\sqrt{SNR})$$

II. ADVANCED DESIGN

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].

A fine frequency synchronization was used to apply a rotation to the initially received signal, prior to the implementation of the RAKEs. The timing synchronization was also applied before the RAKEs, using the newly rotated signal from the fine frequency synchronization.

A. Fine Frequency Synchronization

The fine frequency synchronization can be applied to calculate the maximum likelihood (ML) of the frequency offset Δ, 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.

$$\hat{\Delta} = \frac{1}{T} \underset{-0.5<f<0.5}{\operatorname{argmax}} \left|Z(e^{-j2\pi f})\right|$$

$$\hat{\theta} = -\angle\left[Z\left(e^{-j2\pi\hat{\Delta}T}\right)\right]$$

Since the signal was to be within the noise floor, spreading was implemented during signal creation by applying 100 pseudorandom bits to each message bit. De-spreading was implemented after equalization, but before the bit detection.

Parameter	Length (symbol)
Frequency Preamble	300
Timing Synchronization	10,000
Pilot	20
Message	100

Table 2: Specifications for Transmitted Signal. There was one message and two pilots.

III. 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 $0.1\mu s$, 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 $800\mu s$ transmission time. This helped to decrease our BER.

IV. 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.