Orthogonal Frequency-Division Multiplexing on the iPad using C++

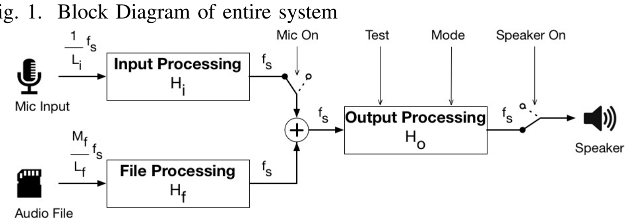
Joseph Loy

***Abstract*—Orthogonal frequency-division multiplexing (OFDM) is a method in which multiple data streams are transmitted over multiple frequency sub-carriers simultaneously. Quadrature amplitude modulation (QAM) is used to take a data stream and modulate it to a complex constellation which is then mapped to one of the sub-carriers by using an inverse fast Fourier transform algorithm. A cyclic prefix consisting of prior output samples of the IFFT unit is prep-ended to the signal before being transmitted over a channel. In order to decode the message, we remove the cyclic prefix of the transmitted signal and extract sub-carrier constellations using an FFT unit. Frequency domain equalization (FEQ) is performed in order to correct for errors introduced by the channel and the signal is QAM demodulated to a bit stream. The goal of the lab was to decode a secret message using the OFDM transceiver.**

I. Introduction

OFDM systems are communication systems that have good spectral efficiency, are robust to inter-symbol interference, and have relatively simple mechanisms for countering the effects of a channel on a transmitted signal. On the other hand, OFDM systems are more susceptible to Doppler shifts, have a relatively high peak to average power ratio, and have other trade offs that make OFDM systems unsuited for some applications. Despite some of the shortcomings of OFDM systems, they are an extremely useful systems that form the backbone of some of the crucial communication technologies that are used today like Digital Subscriber Lines and 4GHz/5GHz wireless networks. Thus, the OFDM project is a great combination of a practical application that is useful and interesting, while still serving as a educational project that incorporates fundamental DSP concepts.The purpose of the OFDM lab is to build a practical, functional digital signal processing (DSP) project that runs on an embedded platform. A variety of DSP concepts such as multi-rate sampling, filtering, and frequency domain processing were incorporated in the project to perform processing on audio file data to later be combined with microphone data and further processed. . A block diagram of the entire system is shown below.

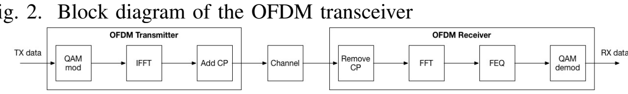
Fig. 1. Block Diagram of entire system



II. Methodology

An OFDM system is comprised of a transmitter and a receiver. The transmission system, consisting of a QAM modulator, IFFT unit, and CP unit, is responsible for encoding a bit stream into a constellation of orthogonal frequency constellations to be transmitted over a channel. This project can transmit data using a wired cable or wireless using a speaker. The receiver consists of a CP handler unit, an FFT unit, an FEQ unit, and a QAM demodulator. A block diagram of the system is shown below.

Fig. 2. Block diagram of the OFDM transceiver



A. QAM modulator

The QAM modulator maps binary data to complex constellations to be later assigned to a sub-carrier. For this lab, we used QAM4, meaning that we assigned the 2 bit input binary data to one of four complex constellations. A table of the mappings is showed below.

Binary	Constellation
00	$-\frac{3}{4}+\frac{3}{4}i$
01	$-\frac{-3}{4}-\frac{3}{4}i$
10	$\frac{3}{4}+\frac{3}{4}i$
11	$\frac{3}{4}-\frac{3}{4}i$

B. IFFT Unit

The IFFT unit modulates the complex data elements to the appropriate carrier frequency before being transmitted.

C. CP Unit

This section adds a cyclic prefix to the data in order to prevent intersymbol interference. If the transmission channel has an impulse response less than v, the length of the cyclic prefix, then we know that there will not be any intersymbol interference. The cyclic prefix unit works by appending v prior output samples of the IFFT unit to the data before transmitting the current symbol.

iOS forces buffers to be a size that is a power of 2, so the basic version of the lab will not have a cyclic prefix, but one can set the cyclic prefix v = 2N, so that we can guarantee no intersymbol interference at the expense of the throughput of the system. This basic version allows for some intersymbol interference.

D. CP Handler Unit

This section removes the cyclic prefix from the transmitted data. For the basic system, this unit is not present. If one were to set v = 2N, this unit removes the first v samples of the data.

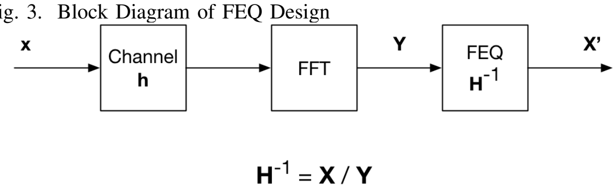
E. FFT Unit

This section performs an FFT on the transmitted data to extract the data from the different sub-carriers.

F. FEQ Unit

The FEQ unit eliminates undesired affects in the channel by estimating the inverse of the channel and applying the inverse to the transmitted data.

Fig. 3. Block Diagram of FEQ Design

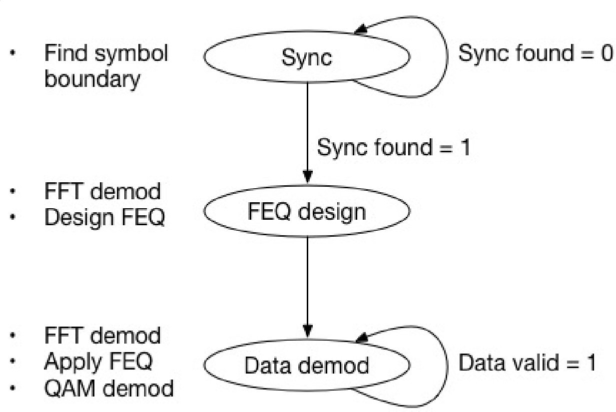


G. QAM demodulator

The QAM demodulator demodulates the the frequency equalized data, which recovers the original data stream that was transmitted by the system. This was accomplished by mapping complex constellations in the transmitted data to their respective bit streams.

The OFDM transceiver operates as a finite state machine (FSM) that is continuously looking for a sync symbol. Once the sync symbol is found, the system transitions into a state where a FEQ equalizer is created using a known FEQ symbol. Finally, FEQ followed by QAM demodulation is applied to the rest of the blocks of the message. A diagram of the FSM is shown below

Fig. 4. State machine for OFDM

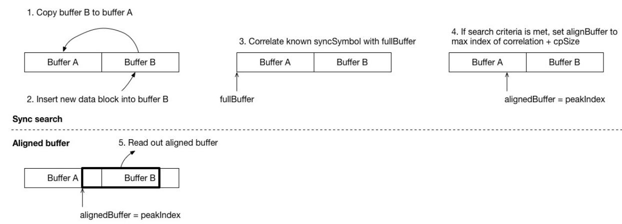


The sync symbol is found by calculating a cross correlation between the received data and the sync symbol which is determined a priori to transmission. By computing the peak to average ratio of the cross correlation, we can find the beginning of the symbol boundary. The peak to average ratio is computed as

$$\text{PAR(x)} = \frac{\max(|x[n]|}{\sqrt{\frac{1}{N}\sum_0^{N-1}|x[n]|^2}}$$

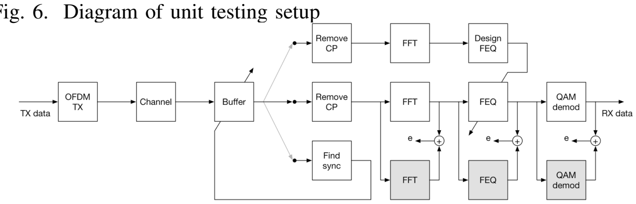
A buffering scheme was used in order to load and store time-domain symbols. Once the symbol boundary was calculated using the PAR of the cross-correlation of the transmitted signal and the sync symbol, the data was read out from the aligned buffer for FEQ estimation and QAM demodulation.

Fig. 5. Linear SyncBuffer alignment setup



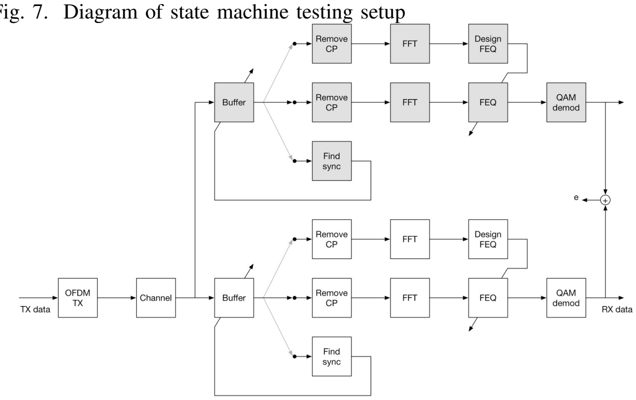
The system is being built and tested in a modular fashion by comparing the output of the OFDM receiver with reference outputs provided by a Matlab implementation of an OFDM transceiver. This was done for each major functional unit in the OFDM receiver. Figure 5 is shown below, where the module tests are depicted where white squares represent units implemented in Matlab and grey squares represent units that run on the iPad.

Fig. 6. Diagram of unit testing setup



The system is then put through state testing in order to ensure that the system transitions from synchronization state to equalize and demodulation state.

Fig. 7. Diagram of state machine testing setup



After passing all of the unit tests and state tests, the system is deployed on the IPad for additional testing and demonstration purposes.

III. Results and Discussion

I was unable to get the lab working completely. Apple's Xcode editor allows one to check CPU usage statistics and memory statistics in order to evaluate the performance of the app on an embedded platform. Furthermore, one could output the data from the embedded platform to a text file in order to look at the power spectral density of the received message, the frequency response of the FEQ matrix, transmit spectral density, and other properties of the signals transmitted/received. The Matlab implementation of the OFDM shows that the spectral density of the received signal is only a few DB smaller than the spectral density of the transmitted signal for the low pass filter channel.

Fig. 8. Spectral density of transmitted signal

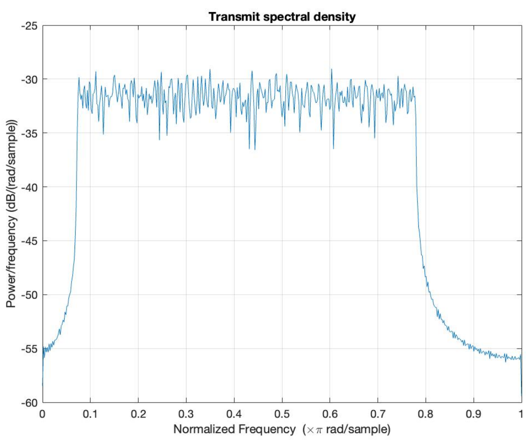
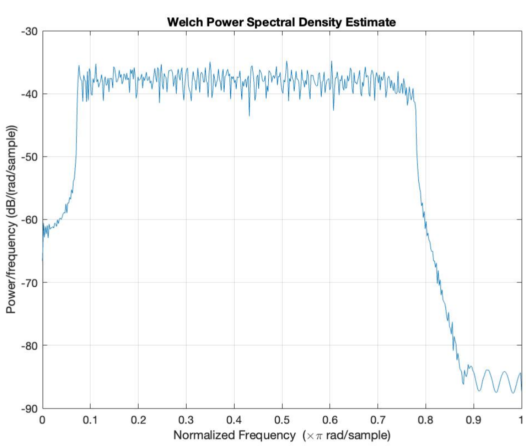


Fig. 9. Spectral density of received signal through low pass filter channel



Notice that the low pass filter is a relatively simple channel. For the random channel, the spectral density estimate is noticeably different than the transmitted spectral density.

A great aspect of the OFDM is the simple solution for countering variance in the channel. Listed in figures 11 and 12 are the different FEQ frequency responses for the low pass filter and the random channel, respectively.

The OFDM is able to perfectly reconstruct the message transmitted through a low pass channel, but the message

Fig. 10. Spectral density of received signal through random channel

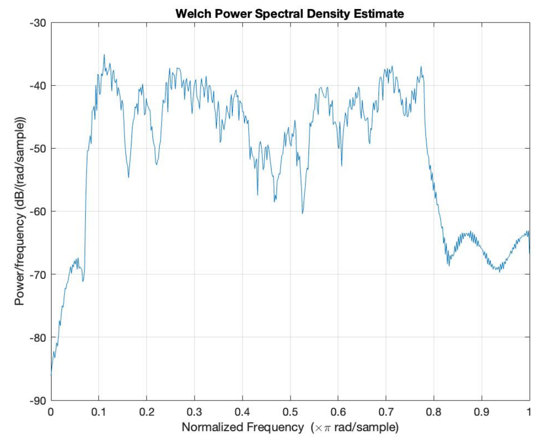


Fig. 11. FEQ applied to transmitted data through low pass filter channel

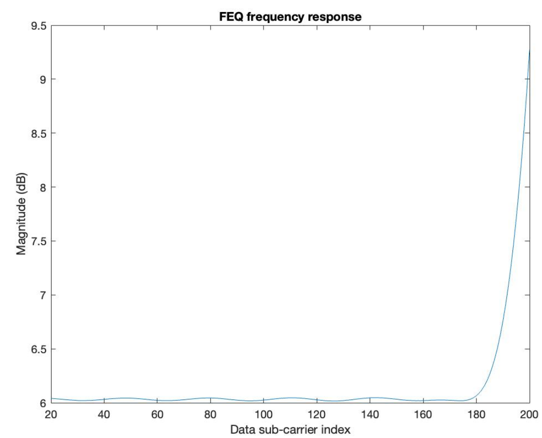
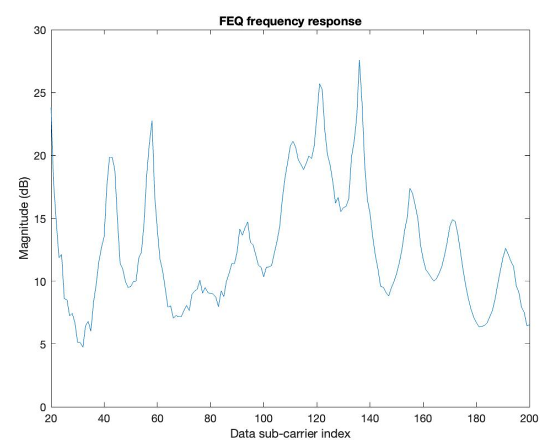


Fig. 12. FEQ applied to transmitted data through random channel



through the random channel has a small degree of error. Fortunately, a space character was turned into a percent sign, so the message was still recoverable. However, it should be noted that there are limitations in the OFDM's ability to mitigate the effects of the random channel. Pictured below are the QAM modulated received signals from the low pass filter and random channel, respectively. Figures 13 and 14 show the

Fig. 13. QAM constellation points for data received through low pass filter channel

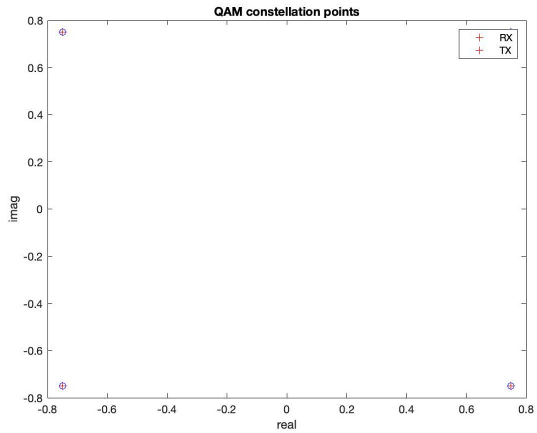
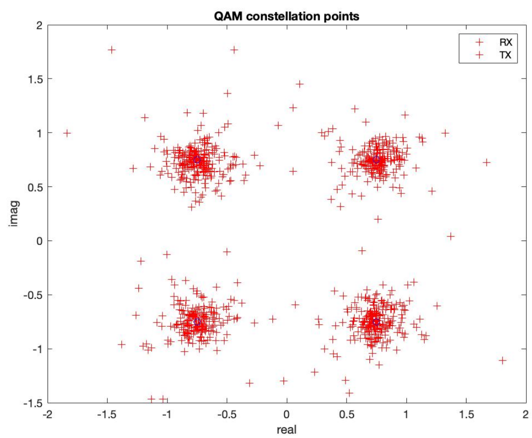


Fig. 14. QAM constellation points for data received through random channel



constellation points of the data received through each the low pass filter channel and random channel.

IV. Conclusion

The OFDM implemented above has good spectral efficiency, a simple equalization mechanism, and can reliably send and receive messages with little error in a variety of channels. The system implemented is as simple as possible, but a variety of extensions could improve the performance. First and foremost, we can set the cyclic prefix to 2N as suggested in the project demo write-up in order to eliminate inter-symbol interference. Another useful extension that was recommended was the circular buffering scheme, which eliminates ISI without affecting throughput. For the sake of providing an original extension, it would be cool to add some sort of speech to text unit so that microphone data can be recorded, transformed into a message, and transmitted over the OFDM. Although the implementation is not completely working at this point, this project has demonstrated a lot of interesting DSP concepts such as modulation, channel equalization, and many more.

V. References Section

References

[1] F Mujica and R Scahfer, *Audio OFDM Project (iOS)*.
[2] F Mujica and R Scahfer, *Audio OFDM Matlab (iOS)*.
[3] Wikipedia contributors. (2022, March 5). *Orthogonal frequency-division multiplexing*.
[4] Oppenheim, A. V., Schafer, R. W., Buck, J. R. (1999). *Discrete-time signal processing*. Upper Saddle River, N.J: Prentice Hall.