1 Introduction

For this project we set out to build and simulated end-to-end wireless communication system based on 802.11n standard. This standard is commonly used today in wireless short to medium range networks. However, the techniques used here such as OFDM, AGC, Channel Estimation, Frequency Offset correction and Phase Tracking are used in many other systems already in use and that are currently in design (UWB).

This project entailed more procedures and simulation that will be described here, but since some of them turned out to be extraneous for the final operation of the system we decided to leave them out. Let us start out with the most important part of any wireless communication simulation - channel modeling.

2 Channel Model

In wireless communications one of the biggest issues is the signal power loss due to transmission. Under free space conditions received power, \( P_r \), could be calculated given transmission power \( P_t \), antennae gains \( G_r, G_t \), distance \( d \) and the wavelength of the transmitted signal \( \lambda \) using Friis’ formula:

\[
P_r = P_t G_t G_r \left( \frac{\lambda}{4\pi d} \right)^2
\]

Although equation (1) gives you a very rough idea about how to build your system it is dangerously oversimplified for a robust real-world system. To help alleviate some of the inadequacies of Friis’ formula, we use Simple Path Loss Model.

2.1 Simple Path-Loss Model

Simple path-loss model builds upon Friis’ model by taking into account different environments in which wireless systems operate from open fields to dense urban macrocells.

\[
P_r = P_t \left( \frac{\lambda}{4\pi d_0} \right)^2 \left[ \frac{d_0}{d} \right]^\gamma
\]

As can be seen from equation (2) that it is actually similar to Friis’ except for the exponent \( \gamma \) and the reference distance, \( d_0 \), which is usually between 1 and 10 meters. The exponent has already been empirically computed for different environments and could be looked up - in our simulations we use \( \gamma \sim 3 \) for a low density urban microcell, since 802.11 is primarily used in shorter range communications. It is important to note that simple path-loss model assumes line of sight between transmitter and the receiver.
2.2 Rayleigh Fading Model

In many wireless scenarios, and especially in indoor wireless networks, there is no line of sight. It is then assumed that the signal scatters and is received from all directions - this is referred to as Rayleigh fading model. This phenomena is modeled via a complex gaussian scaling of the received power, and is usually considered the worst possible scenario. However, there are several techniques that could be used to improve the performance of systems in Rayleigh channels.

2.3 AWGN

Independent of which channel is available to you there is always noise present at the receiver antenna. Most commonly this noise is modeled as Additive(complex) White Gaussian Noise (AWGN). Signal-to-Noise ratio (SNR) at the receiver is the decisive factor in system performance. In LoS setup noise is simply added to the received signal, thus maintaining more-or-less a constant SNR. In Rayleigh systems the noise remains constant and is still added to the received signal, however due to the effects of the Rayleigh the SNR varies and is worse than in similar LoS setup.

2.4 Frequency Selective Channel Model

Presence of strong multipath components present at the received has an undesired effect on the channel making it a frequency selective channel. Such channel fades differently across different frequencies, which is devastating for an OFDM system since some of the carriers could fall into deep fade and thus be indistinguishable from noise.

3 802.11 Packet

The packet itself has several fields that need to be correctly build for the proper operation of the system (mixed mode), the fields are listed in chronological order:

- L-STF – Legacy Short Training Field is primarily used for coarse frequency detection, AGC calibration and rough channel estimation. With duration of 8μs, L-STF is attached in front of every packet.

- L-LTF – Legacy Long Training Field comes into the picture once the receiver used the information contained in L-STF to do preliminary packet and channel analysis and is used for a higher precision estimates of the channel and frequency offsets. Duration is also 8μs.

- L-SIG – Legacy Signal Field is used to related legacy information about the data to come, like data length. L-SIG has the duration of 4μs.
• **HT-SIG** – High Throughput Signal Field is used to relate relevant payload information, like payload length, encoding used, bandwidth used and the number of HT-LTFs to come. This field is of extreme importance at the MAC layers of packet reception and decoding. With duration of $8\,\mu s$ it also contains a CRC. If this field fails the cyclic redundancy check that the whole packet will be discarded.

• **HT-STF** – High Throughput Short Training Field is used to improve AGC operation in multi-transmit and multi-receive systems.

• **HT-LTF** – High Throughput Long Training Field is used to estimate the channel between different spatial streams in operational modes other than 1x1. The number of HT-LTF fields determines the number of spatial streams.

Upon building the packet in frequency domain, we perform and IFFT on the whole packet to produce different carrier signals for the OFDM transmission scheme. NOTE: Most of the fields are not needed to be implemented for PHY level simulations.

### 3.1 LTF and STF Fields

To better understand the overall process of packet reception we must dig deeper into the structure of the legacy STF and LTF, since they are the ones used for the primary system calibration at the arrival of every packet. At first inspection we notice that L-STF in time domain is a repetition of the same pattern 10 times. This repetition benefits us in several ways: block boundary detection, Automatic gain control (AGC) and coarse frequency offset estimation all of which will be discussed later in the report. Similarly, LTF field has also repeating pattern in the time domain, but it repeats only twice with 64 symbol interval. LTF will be used primarily in our fine frequency offset estimation.

### 3.2 Packet Construction

Interesting thing about 802.11 mixed mode communications is that packets are constructed in frequency domain. Thus for each of the following fields mentioned above we create a length 64 vector to represent the data, the actual fields only contain 56 channels to represent the 56 carriers used for data and pilots. The remaining 8 carriers or channels are used as guard bands. There are 7 guard bands total at the edges and there is one very important guard band at DC. The DC guard bands is mostly placed to protect the system as a whole from DC operations.

The final field in the packet is the payload or data. Our payload size is 256 bytes, which is padded in front by a service field. The service field is 16 0’s and is used to recover the initialization state for the of the scrambler, since the scrambler’s register always starts of with a pseudo-random value.

Once the whole packet has been built according to specification we go through blocks of 64
symbols and insert pilots into -27, -7, 7, 21 \textsuperscript{st} positions before we take an IFFT and send it over the channel. Pilots contain lots of valuable information, like what is the spatial configuration used in the system and also provide help with phase offset tracking and correction.

3.3 Alternative Approaches & Discussion

The implementation had to go according to the 802.11n standard, so there is no room for different approaches.

The packet construction is pretty extensive and did required investigation. The part here that’s important to note that most of the fields deal with MAC layer information; on the other hand, we’re doing PHY layer simulation and that information is completely useless. And although, building extraneous parts, like the CRC, wasn’t that hard it was tedious and time consuming.

4 AGC

For Step 6 of the project and this final report, we had to alter the Automatic Gain Controller (AGC) method that we had in Step 3, to suit the new purpose. Previously, in Step 3, the purpose of the AGC was to eliminate the slow fading and path loss affects of the incoming signal, by scaling the input stream between +1V and -1V, with a parameter that limited the probability of saturating the stream to either 1\% and 10\%. Using probabilistic methods, we had approached this problem. As expected, for more strict saturation criteria (e.g. 1\%), the stream output from the AGC was lower in amplitude than that using a more lenient saturation criteria. This directly lead to errors in demodulating the bits from the data fields, as the signal energies were changed.

Therefore, for Step 6, in which the primary concern is to achieve the symbol demodulation with least number of errors as possible, the AGC has to adjust the signal power so that the incoming data fit the decision regions (e.g. for 16-QAM demodulation). Using the same algorithm as in the previous AGC, we calculate the signal energy of the L-STF (specifically, of the first 4 blocks, corresponding to 3.2 \mu s), but instead of adjusting the gain by a saturation-dependent parameter we do so to make the signal energy equal to 1, so that the bits can be decoded optimally, using the correct decision boundaries. Symbolically:

\[
\text{Gain} = \frac{1}{\sqrt{\frac{1}{L} \sum_{k=0}^{L-1} |x_k|^2 \times \frac{64}{52} \times 64}} \quad (3)
\]

\[
y_n = x_n \times \text{Gain} \quad (4)
\]

where the factor of \frac{64}{52} normalizes the gain, taking into account that there are 52 used subcarriers (48 for data, 4 for pilots) out of a total of 64 subcarriers, and the factor of 64 handles the scaling factor of the 64-point FFT.
4.1 Alternative Methods

In a real life implementation, AGC should still serve its original purpose by adjusting the gain so that the data does not saturate with a given probability. However, to make sure that the symbol demodulation is done using the correct decision boundaries, another block has to be used, that scales either the input or the decision boundaries -in the digital domain- based on the input stream energy. We have learned that, for real-life implementations, the packets include a training data field with a known structure, helping the receiver adjust its gain for symbol demodulation.

In this project, since we do not have this training data field and we are not dealing with A/D converters, we decided to keep things simpler by changing the AGC to fit the second purpose. The saturation criteria is no longer checked, although the demodulation is carried out correctly, as indicated by the BER curves.

5 Carrier Frequency and Phase Offsets

Carrier frequency offset is a phenomenon that is caused by the manufacturing variability of the crystal oscillators, and is measured in parts per million (ppm). Since the offset in ppm is inversely proportional to the price of the crystal oscillator most communication standards allow for carrier frequency offset, keeping in mind that it must be corrected by the receiver. The main reason why correcting carrier frequency is so important is because it causes the modulating constellations to rotate at the receiver, causing a severe drop in BER performance of the system.

For this part of the project we had to develop coarse frequency estimation procedure, then use the coarse estimate that coarse estimate to acquire and fine frequency estimate and finally correct the carrier frequency offset. We used the last 3 L-STF blocks for the coarse frequency estimation procedure, since the first 7 L-STF blocks are used for AGC and packet detection. Then we utilized that estimate and L-LTF section to do fine frequency estimation.

5.1 Coarse Frequency Estimation

For coarse frequency estimation we decided to use a time domain approach over the last 3 L-STF symbols. We were able to do that since L-STF is periodic every 16 symbols\((D = 16)\).

\[
z = \sum_{n=0}^{L-1} r_n r_{n+D}^* \tag{5}\]

Where \(r_n\) is a received symbol for an L-STF block, \(r_{n+D}\) is the same symbol from the next L-STF block, \(L\) is the number of symbols available and \(z\) is just an intermediate step in the calculation. Frequency estimate is given by, where \(T_s\) is the symbol interval:

\[
\hat{f}_\Delta = -\frac{\angle z}{2\pi DT_s} \tag{6}\]
To show that equation (6) does in fact estimate the frequency offset, \( f_\Delta \), with the help of equation (5). Let’s us consider received signal \( r_n \), where \( f_\Delta = f_{tx} - f_{rx} \):

\[
\begin{align*}
 r_n &= s_n e^{j2\pi f_\Delta n T_s} \\
 z &= \sum_{n=0}^{L-1} s_n e^{j2\pi f_\Delta n T_s} \left( s_n e^{j2\pi f_\Delta (n+D) T_s} \right) \\
 &= e^{-j2\pi f_\Delta T_s} \sum_{n=0}^{L-1} |s_n|^2 
\end{align*}
\]

(7)

Thus it can be seen from equation (7) that the angle of \( z \) is equal to \(-2\pi f_\Delta D T_s\). The algorithm seems to work fairly well even under low SNR conditions like 5dB, as can be seen in figure 1.

![Figure 1: Coarse Frequency Estimation Performance](image)

Once we find \( \hat{f}_\Delta \) we could now correct the carrier frequency offset. Since phase is defined as change of frequency over time, or in our case symbols, we can just correct the phase of each symbol by an appropriate amount which linearly grows with each successive symbols. This could be implemented in the following fashion:

\[
y_n = r_n e^{-j2\pi n T_s \hat{f}_\Delta}
\]

(8)
5.2 Fine Frequency Estimation

After observing outstanding performance of the time domain frequency estimation algorithm used in the previous section and its simplicity we decided to implement it for fine frequency estimation. However, we do correct the frequency offset by coarse frequency estimate before applying fine frequency estimate. It is very important that we do that, since time domain estimation algorithm has bounds on initial frequency offset that it can estimate correctly. These bounds are given by:

\[ |f\Delta| \leq \frac{1}{2DT_s} \]
\[ \therefore f_{\Delta_{\text{max}}} = \frac{1}{2DT_s} \]  \hspace{1cm} (9)

Since we know that the sample time \( T_s = 50\,\text{ns} \) and \( D = 16 \) for short training sequence, we can calculate this bound from equation (9) \( f_{\Delta_{\text{STF}}}^{\text{max}} = 625\,\text{kHz} \). However, long training sequence is only periodic in \( D = 64 \) giving us a bound of \( f_{\Delta_{\text{LTF}}}^{\text{max}} = 156.25\,\text{kHz} \). Therefore, if we just use LTF for frequency estimation we will not get a correct estimate if the \( f_{\Delta} > f_{\Delta_{\text{LTF}}}^{\text{max}} \). Also, if we refer back to (5) we can observe that the estimate would improve with larger size \( L \), since that would average out the AWGN component of the signal. Due to these factors it is clear why we must use a two stage frequency offset estimation. The performance of coarse and fine frequency estimations can be seen below in figure 2.

5.3 Coarse & Fine Frequency Estimation Performance

<table>
<thead>
<tr>
<th>SNR</th>
<th>( \mu )</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5dB</td>
<td>40Hz</td>
<td>1700Hz</td>
</tr>
<tr>
<td>15dB</td>
<td>18Hz</td>
<td>550Hz</td>
</tr>
<tr>
<td>25dB</td>
<td>-11Hz</td>
<td>164Hz</td>
</tr>
</tbody>
</table>

Table 1: Cumulative Performance of Both Offset Estimations, \( f_{\Delta}^{\Delta} \)

Our metric for coarse and fine frequency estimation performance was the estimate error, given by \( f_{\Delta} = f_{\Delta} - \hat{f}_{\Delta} \). Therefore after correction the new carrier offset is equal to \( f_{\Delta} \). As can be seen from the results in table 1, the mean of the estimate errors is close to 0Hz. We can access our results with the use of proportionality [2]:

\[ \sigma_{f_{\Delta}}^2 \sim \frac{1}{L \times SNR} \]  \hspace{1cm} (10)

As can be seen from equation (10) the variance for the resulting frequency offset estimate is inversely proportional to the SNR, and our results maintain that proportionality. However, with lower SNR we still have high variability in results. Even though we it is possible to improve the performance of the overall system frequency offset estimate using higher number
of symbols, the benefits would be marginal in proportion to the added training sequence length. Small variability in the final offset estimate is fine, because the remaining offset error can be corrected by the carrier phase correcting procedure. Rotation from symbol to symbol can be computed by:

\[ \theta = 360 \times f_\Delta T_s = 360(1.2 \times 10^5)(50 \times 10^{-9}) = 2.16 \]  
\[ \theta = 360 \times f_\Delta T_s = 360(1.7 \times 10^3)(50 \times 10^{-9}) \approx 0.03 \]  

From equation (11) we can see that the rotation speed was 2.16°/symbol, which is too fast for the carrier phase tracking to correct. However, after we corrected the carrier frequency offset the rotation speed was only 0.03°/symbol, as seen in equation (12), which is two orders of magnitude smaller. Regardless of the fact that frequency offset estimation is not as precise as desired, it is good enough to make sure the phase can be now corrected by the carrier phase tracker.

5.4 Phase Tracking

Since the 802.11n standard has previously discussed pilots signals, we can use them to our advantage in implementing data-aided carrier phase tracking.

\[ R_{nk} = H_k P_{nk} e^{j2\pi n f_\Delta} \]  

Figure 2: Coarse + Fine Frequency Estimation + Phase Tracking Performance
With $R_{nk}$ is the n-th received pilot, with $H_k$ being the channel frequency response and $P_{nk}$ known transmitted pilot. As the symbols are rotated by the frequency error and phase offset knowing the information in equation (13) we can estimate the carrier phase in the following fashion:

$$\hat{\phi}_n = \angle \left[ \sum_{k=1}^{N_p} R_{n,k} \left( \hat{H}_k P_{n,k} \right)^* \right]$$

$$= \angle \left[ e^{j2\pi f \Delta} \sum_{k=1}^{N_p} |H_k|^2 \right]$$

(14)

However this process also will have error introduced into it by imperfections in the channel estimate. As can be seen in figure 2 the overall system performance results in carrier frequency error at around 0 Hz, which is negligible over the length of one packet. We can see a significant improvement in performance summarized in table 2.

<table>
<thead>
<tr>
<th>SNR</th>
<th>$\mu$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5dB</td>
<td>10Hz</td>
<td>268Hz</td>
</tr>
<tr>
<td>15dB</td>
<td>-2.6Hz</td>
<td>50Hz</td>
</tr>
<tr>
<td>25dB</td>
<td>0.5Hz</td>
<td>15Hz</td>
</tr>
</tbody>
</table>

Table 2: Updated Performance w/ Phase Tracking

5.5 Alternative Approaches & Discussion

In our project we proposed and implemented time-domain approach to carrier frequency estimation; however, it is possible to do frequency-domain carrier frequency estimation. This approach relies primarily on inter-carrier-interference that is introduced by a shift in carrier frequency. Since the complexity of implementation for this algorithm is higher, yet the performance is comparable to time-domain approach we decided to implement the time-domain frequency estimation [1].

There is also a different approach that we considered for carrier phase tracking - non-data-aided carrier recovery. This approach centers on the fact that with out knowing what was sent it is possible to just make a hard decision on the received data and use that estimate in the similar fashion to equation (14). There will be incorrect hard decisions, but over larger sample size of the data the errors would averaged out. As this approach might be fairly reliable for BPSK or QPSK, there will be a much higher error rate associated with 16/64-QAM constellations. However, the standard does provide us with known information - Pilots - this eliminated any incorrect decodings that would be seen with hard decision.

The actual carrier frequency estimation and correction was not too difficult to implement, but we discovered the issue of the residual frequency error. After some though we
decided to use phase correction to fix this problem, it was a little more extensive in coding since the symbols had to be first decoded then the phase was estimated and corrected on the original un-decoded data.

6 Channel Estimation

To do basic channel estimation we use the two LTF sequences. We compute the frequency-domain representation of two identical LTF sequences by taking 2 64-point FFTs, then average them together. Each bin in the FFT corresponds to the channel effect on a particular carrier. To eliminate the effects of the channel on our data we divide out each carrier by the scaling factor computed above. Nothing fancy in this step.

7 MIMO

The 1x2 configuration means receiver diversity, as the same signal is detected using two receivers, where the two paths are ideally independent of each other. To realize this configuration, we implemented Maximal Ratio Combining (MRC), the optimum method for receiver diversity. Using the channel information, which is basically the gains of the two channels, as indicated by the input signal energies detected by the AGC blocks, we can combine the 2 input streams by weighting each branch inversely to its respective signal energy. For the 2x2 configuration, we decided to send the same signal over both channels and use Zero Forcing (ZF) Decoder to do the MIMO decoding.

\[
Y = HX + N
\]

\[
H^{-1}Y = \tilde{Y} = X + H^{-1}N
\]

\[
\hat{Y} = (\tilde{y}_1 + \tilde{y}_2)/2
\]

7.1 Alternative Methods

For 1x2 configuration, we could also have easily implemented Equal Ratio Combining (ERC) or Selection Combining (SC), which are both sub-optimum methods leading to higher bit error rates. Therefore we claim that we have implemented the best algorithm possible, which show a 3 dB improvement under AWGN compared to the 1x1 configuration, which is parallel with the theory, as shown by the BER graphs in Step 6.

For the 2x2 configuration, other choices could include Minimum Mean Squared Error (MMSE) solution, non-linear methods, Sphere Decoding and Maximum Likelihood (ML) decoding, which would all perform better than the ZF method which we have implemented. The reason we chose ZF over the other methods was due to its simplicity, implementation-wise and computationally. As expected, the performance of our 2x2 is not significantly better
than the 2x1 system, which is because of the noise enhancement and coloring that is present in ZF decoders. Yet, the important point was to understand how to make use of the channel information to build a working 2x2 MIMO system, which we have successfully achieved, as indicated by the BER curves, given in Step 6.

Furthermore, for the 2x2 configuration, we could have gone for different combinations of multiplexing and diversity gains whose product would be 4. We decided to use full diversity gain, and no multiplexing gain, because this would show the most improvement on the BER curves, and help us compare the 2x2 configuration with the 2x1 using the same data rate, resulting in a fairer comparison. The best way to achieve full diversity gain with a 2x2 system would be to use Alamouti Codes, which is a type of space-time coding. However, implementing a space-time code would introduce further complexity to our system and we felt that this was beyond the scope of this project. Therefore we have transmitted the same signal from the two transmitters, and weighted the signals \((\tilde{y}_1 + \tilde{y}_2)\) with MRC to have the full diversity gain.

### 8 Results

#### 8.1 AWGN

![Figure 3: AWGN, 1x1 configuration, 1-ray model](image)

Theoretical
Ideal
AGC On
Channel Estimate On
All On

\[ 16\text{-QAM Performance, AWGN, 1-Ray Model, 1x1} \]

\[ 10^{-4} - 10^{0} \]

\[ 0 - 18 \text{ (dB)} \]

Bit Error Rate

\[ E/N_0 \text{ (dB)} \]

Figure 3: AWGN, 1x1 configuration, 1-ray model
After successful completion of all the steps the following observations were made for AWGN 1x1 spatial configuration:

- **AGC** – At high SNR, AGC works near close to theory; however, at lower SNR (1-6 dB) we can see the imperfections that result from non-ideal AGC. Due to larger noise the dynamic range is inaccurately estimated which leads to higher BER.

- **Channel Estimation** – As can be seen from figure 3 performance degradation associated with imperfect channel estimate is on the order of $2\,dB$.

- **Frequency Estimation** – As previously discussed, our frequency estimation alongside carrier phase tracking produces excellent results (summarized in table 2). However since channel estimation introduces noise into the system before the frequency estimation, this produces another $1\,dB$ of degradation.

With the total loss in SNR of around $3\,dB$ that our system performs fairly close to theory. It can also be observed that AWGN with 4-Ray Model has very similar results, as seen in figure 4; however, one must consider that the multipath becomes a fairly strong noise component resulting in additional $2\,dB$ SNR loss.
8.2 Rayleigh

Rayleigh is considered the worst case scenario for wireless channels, and as can be seen from figures 5 and 6 the performance is significantly less than that of the AWGN channel. For 1x1 spatial configuration Rayleigh channel we can see that with all things at ideal our system performs close to theory. In figure 5, it’s important to note that the Theory line is actually an asymptote and not the exact curve; furthermore, it’s more accurate for high SNR. From the plots we observe that once again AGC performance doesn’t induce a significant performance degradation. The channel estimate on the other hand introduces the highest loss in SNR out of all components, at around 8dB loss it’s much worse than in AWGN case. Frequency estimation performs relatively well, and only contributes additional 2dB loss to the overall system SNR.

For 1x2 spatial configuration we implemented diversity with Maximal-Ratio-Combining. From figure 6 we can see that MRC gains us around 8dB SNR with respect to theory.
Figure 6: Rayleigh, 1x2 configuration, 1-ray model

References
