ADC Final Project: Visible Light Communications

Mark Goldwater, Vienna Scheyer, Grace Montagnino, Christina Wang

April 2019

1 Introduction

To implement a visible light communication system, we used a software defined radio (SDR) platform to transmit and receive signals. To make this work, we amplified the signal out of an SDR and applied a DC offset and sent that signal into a light bulb. On the receiving end, we used a photo-detector to detect variations in the light levels then removed the offset and decoded the transmitted data.

1.1 Background/Context

Visible light communication (VLC) dates back to when the Romans would polish metallic plates to reflect sunlight in various ways to make different signals over a span of distance [2]. Following this, VLC was next prominent in the 19th and 20th centuries when the heliograph was used to reflect sunlight with a mirror to transmit Morse code [2].

Visible light communication has sometimes been referred to as "the communication system of the future" because it is harmless to the environment and to humans [1]. Similarly, because of its reliance on LEDs, it is considered a "green" technology. LEDs end up actually being ideal for visible light communications (VLC) because LEDs can be modulated much quicker than a standard light bulb [1]. Then on the receiving end, photo-diodes are inexpensive, but powerful enough to detect the intensity changes.

One of the noted benefits of VLC is the security improvement. VLC is confined by opaque walls which essentially confine the signal. However, direct line of sight is not required to make VLC work; it can work with considerable obstruction [1]. However, it does not interfere with RF signals. It is considered a fairly practical alternative to RF in environments such as hospitals, mines, and airplanes [1] because it is an affordable fix to the radio frequency scarcity [2]. The availability of LEDs already in spaces everywhere, makes this idea even more possible and affordable. Other applications of VLC being explored include indoor GPS and near field communications [1].

1.2 System Design

Figure 1 is a block diagram outlining how the signal is transmitted and received. This set-up can support both BPSK (Binary Phase Shift Keying) and QPSK (Quadrature Phase Shift Keying), the only difference being how the imaginary signal is used. Walking through the diagram as would be used for QPSK, x[k] is the input bits, and y[k] is the received data. In the block diagram, the first loop and the second loop are essentially black boxed by the USRPs, where the first loop is the transmitter (Tx) and the last loop is the receiver (Rx).

After the USRP has the signal that it needs to transmit, it breaks it into its real and imaginary components and converts them into analog signals. At this point in the diagram, the signal is a square wave, where ups and downs represents bits. The next step is to multiply the real part of the data with the carrier signal of $\cos(2\pi f_c t)$ and the imaginary part with -$\sin(2\pi f_c t)$. Finally the real and imaginary parts are added together before the USRP transmits. From here, the signal is run through an amplifier and then a Bias Tee. The Bias Tee adds an offset to the signal. We have to do this because before this our signal is centered around zero, but in order for the LED to be powered, the signal needs to be centered more around 22 volts. Now that the

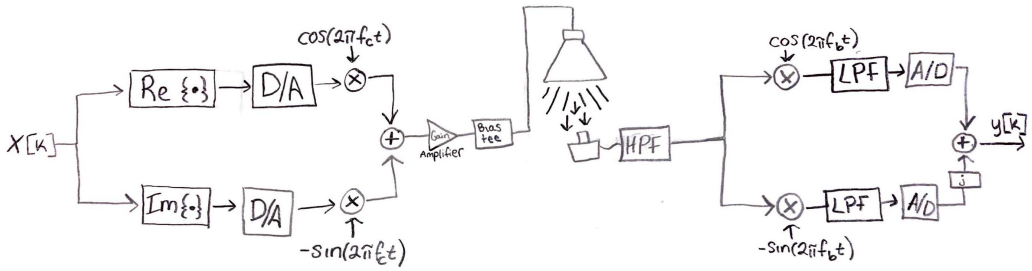

Figure 1: Visible Light Communication Block Diagram using two USRPs.

signal is at the right level to turn on the light and have enough wiggle room on either side of its set level to communicate 1s and 0s.

The data is then initially received by the photo-diode, but the signal at this point is still offset due to noise as well as the light present in the room, so it needs to run through a high-pass filter to recenter the signal around zero. Now the signal starts being received by the second USRP. The USRP starts by removing the carrier frequencies by multiplying the real and imaginary components by their respective carriers. Next the signals are passed through a low pass filter to clean up the signal and are converted from analog back to digital signals. Finally, to make the QPSK signal, the imaginary signal needs to be multiplied by j and then added together with the real components. This combination is y[k], which is what we receive on the computer before we start to process the data.

2 Binary Phase Shift Keying

We started out by implementing BPSK, which is a two phase modulation scheme in which the 0 and 1 bits are represented by the phase of the carrier wave. Figure 3 shows a visual of this phase shift keying method.

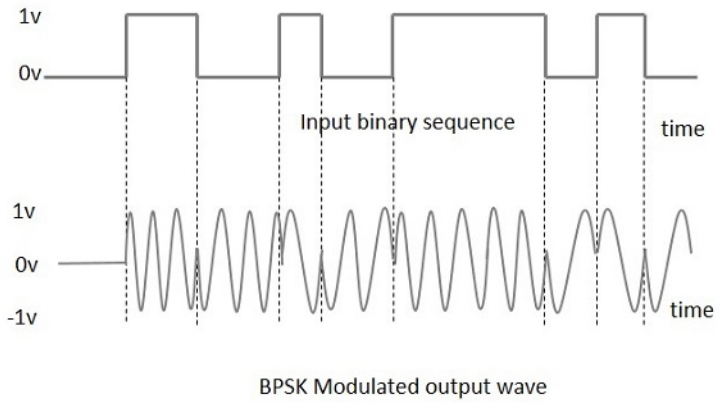

Figure 2: BPSK descriptive image

In BPSK, we send only real bits. Referring to our system diagram in Figure 1, we are only modulating bits by a cosine wave.

On the receiving end, the first operation we need to perform is finding the lag between the sent and received signals. We have a header file that is 100 bits long, and we have this same header on the sending and receiving ends. To find the lag, we take the maximum of the cross-correlation of the sent and received

header file because this is where the files are best aligned. In other words, we take the maximum of Equation 1.

$$R_{yx} = E(Y(t+\tau)X(t)] \tag{1}$$

We then used the maximum value of our cross correlation to figure out the start time of our data in our received signal. Using this information we cropped in on our data as shown in Figure 3.

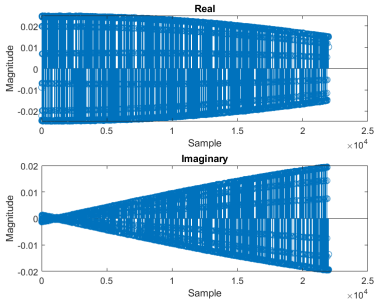

Figure 3: BPSK descriptive image

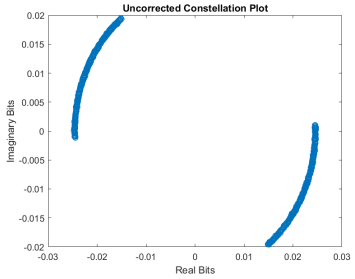

Figure 4: Constellation plot of received signal.

Although we transmitted only over the real channel, some signal shows up in the imaginary channel because there is some amount of frequency and phase offset. Assuming the noise is about zero (n[m] $\approx$ 0) we can model our system with the following equation:

$$y[m] = (x \star h[m])e^{j(f_\Delta m+\theta)} \tag{2}$$

In this equation we have extracted the system into the channel h[m] which, when convolved with x[m], as long as the clocks on the modulation signals are the same, gives the output signal y[m]. However, the frequency and phase offset can be accounted for in the exponential coefficient in Equation 2. In order to find and correct for the frequency and phase offset, we plotted the FFT of the squared received signal as shown in Figure 5.

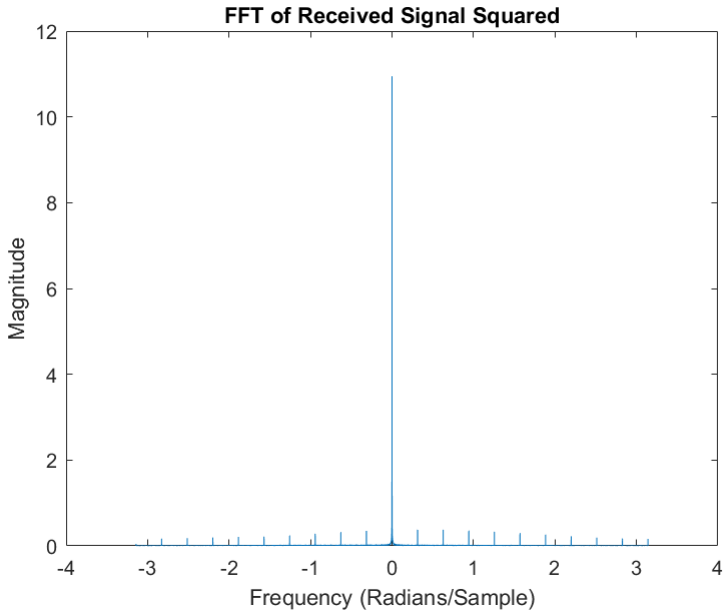

Figure 5: FFT of received signal squared

The frequency of the largest peak is twice the frequency and its angle is twice the offset angle as shown in the following equations.

$$2f_\Delta = f_{peak} \tag{3}$$

$$2\theta = arctan(\frac{imag(2f_\Delta}{real(2f_\Delta}) \tag{4}$$

We solved for f_Δ and θ, and then we divided the signal by the exponential from Equation 2 to get the signal convolved with the channel. Using the signal convolved with just h[m] is sufficient because the channel is relatively clean and small in magnitude in comparison to x[m]. After correcting the frequency and phase offset, the signal is still rotated by some factor of π. To correct for this, we try rotating the header file by π and check to see if the rotated or un-rotated received header matches the original header better. Once we figure out whether a rotation is necessary, we apply the rotation if needed. This produces our final signal as shown in Figure 6

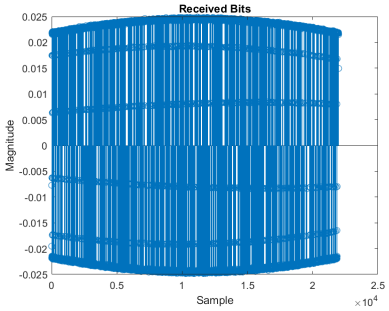

Figure 6: BPSK received bits

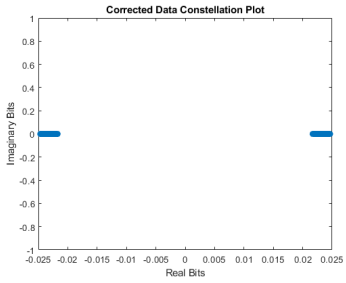

Figure 7: Constellation plot of received signal.

Finally, we calculated our bitrate with the following equation:

$$bitrate = 250k\frac{samples}{second} \cdot \frac{1}{20}\frac{symbols}{sample} \cdot 1\frac{bits}{symbol} = 12,500\frac{bits}{second}$$

This gives us a bitrate of 12,500 bits per second

3 Implementing 4-QAM

Our next step was to implement 4-QAM in our visible light communication system. In comparison to BPSK, we are now sending 1s and 0s over both the real and imaginary "channels." This allows us to send information at a rate of 2 bits per symbol. In order to implement this the process of sending and receiving has changed slightly. First, rather than simply sending a string of 1s and 0s we now send a string of complex numbers of the form $\pm 1 \pm 1j$. These bits were then convolved by a square pulse of 20 samples, padded the beginning and the end of our data vector with a large array of zeros, and sent using a USRP1.

After sending and receiving the signal at a sample rate 250 kHz, we once again used the cross correlation between our new complex header and our complex received signal (using Equation 1) in order to zoom in on our received data.

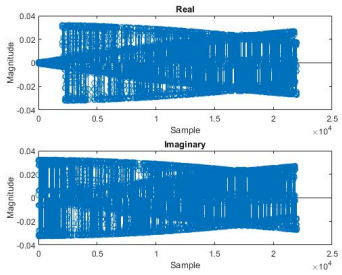

Figure 8: Real and Imaginary Components of Received Signal.

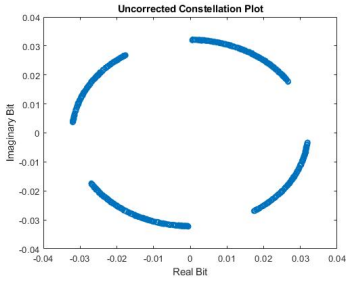

Figure 9: Constellation plot of received signal.

The misshapen nature of the imaginary and real components of the received signal is due, once again, to there being a frequency and phase offset in the sine and cosine waves used to modulate and demodulate the signals. Moreover, one can see that each of the four bit combinations shown in the constellation plot are skewed and have a fairly large variance. Ideally, we should see four clusters in each of the four corners of the plot. In order to correct for this, we used the abstracted model of our system which, assuming the noise is about zero (n[m] $\approx$ 0) we can model our system with the following equation:

$$y[m] = (x \star h[m])e^{j(f_\Delta m+\theta)} \tag{5}$$

In this equation we have extracted the whole 4-QAM system into the channel h[m] which when convolved with x[m], as long as the clocks on the modulation signals are the same, gives the output signal y[m]. However, the frequency and phase offset can be accounted for in the exponential coefficient in Equation 5.

In order to find the phase and frequency offset to correct for it, we plotted the FFT of the fourth power of our received signal as shown below in Figure 10.

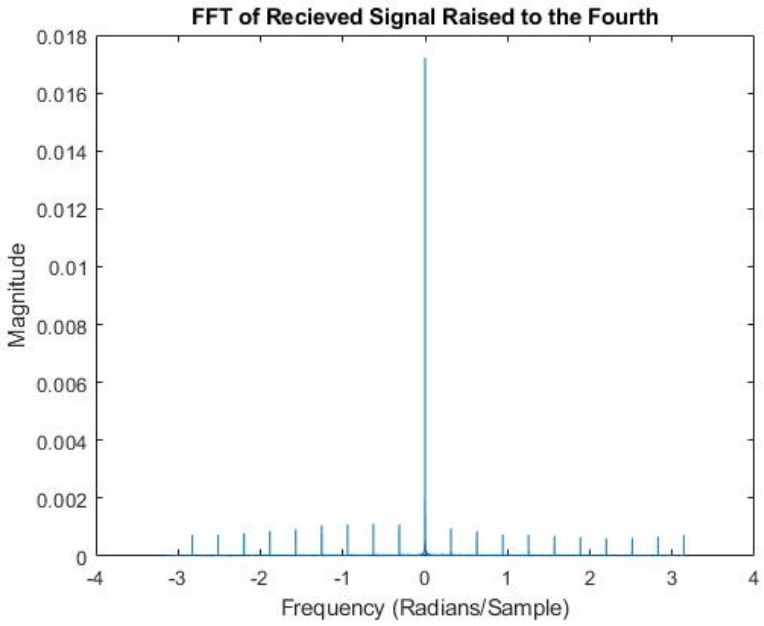

Figure 10: This figure shows the FFT of the received signal raised to the fourth power.

We the found that the frequency value of the tallest peak is equal to four times the frequency and its angle is four times the offset angle as shown be the equations below:

$$4f_\Delta = f_{peak} \tag{6}$$

$$4\theta = arctan(\frac{imag(2f_\Delta}{real(2f_\Delta}) \tag{7}$$

We used these relations to solve for both f_Δ and θ and then divided the received signal by the exponential in Equation 5 to yield the bit signal convolved by the channel as shown by the following equation.

$$\frac{y[m]}{e^{j(f_\Delta m+\theta)}} = (x \star h[m]) \tag{8}$$

Having $x \star h[m]$ is sufficient enough for us to be able to see the bits we received, for the message signal is a lot stronger than our fairly clean channel. However, after applying this phase and frequency correction, our phase correction appeared to be off by a factor of $\frac{\pi}{4}$ as shown below in Figure 11.

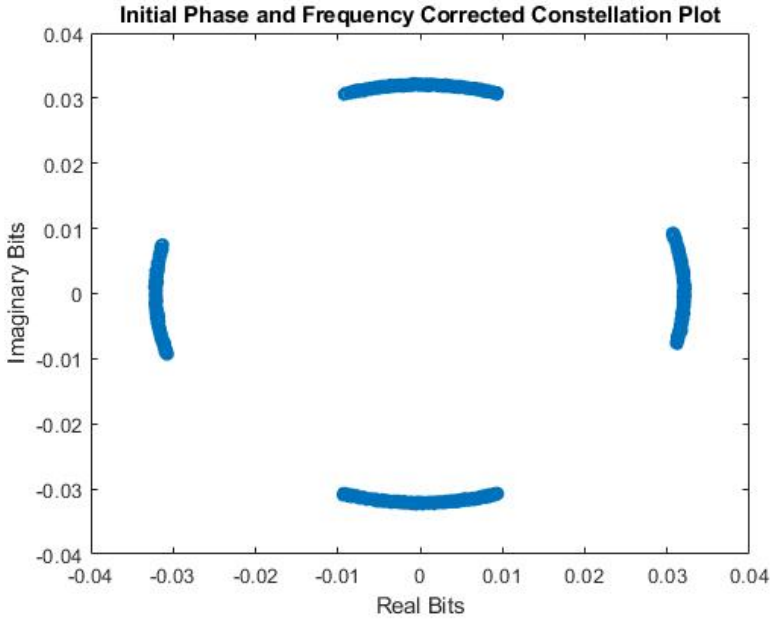

Figure 11: This figure shows constellation plot of our received signal after an initial phase and frequency offset correction. As one can see there seems to still be a phase offset of $\frac{\pi}{4}$

This extra factor comes from when we raised our received signal to the fourth power. Our ideal received signal can be represented by $e^{j\frac{\pi}{4}}$ where the coefficient of j can be any multiple of $\frac{\pi}{4}$ which will change the signs within the complex number which represents the bits. When this is raised to the fourth power we get $e^{j\pi}$. One can then deduce that the peak in Figure 10, can be represented by $e^{j\pi}e^{j(4f_\Delta+4\theta)}$ which can be re-written as $e^{j(4f_\Delta)}e^{j(\pi+4\theta)}$ which, once the calculated phase offset is divided by four, shows where the extra phase offset of $\frac{\pi}{4}$. We corrected for this by multiplying the corrected-once received signal by the complex exponential $e^{-j\frac{\pi}{4}}$ and obtained the corrected constellation plot shown below in Figure 12.

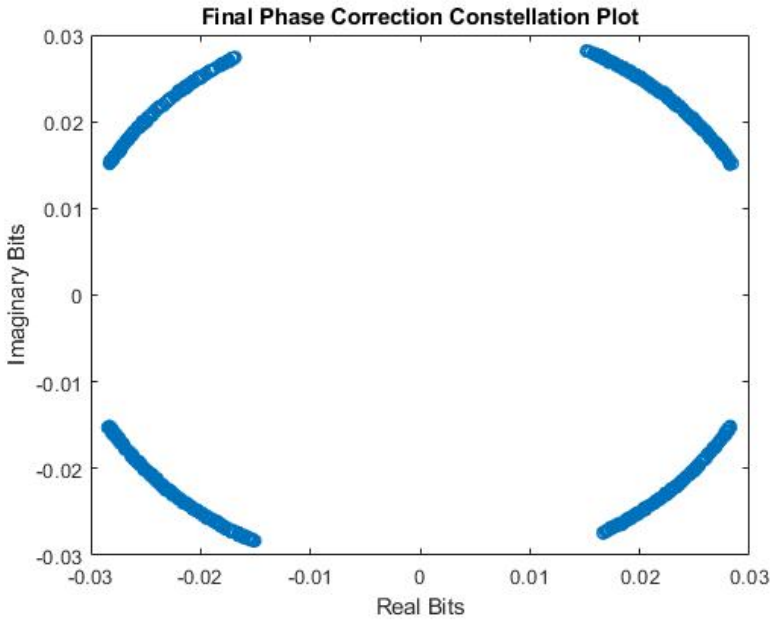

Figure 12: This figure shows the correct constellation plot after accounting for the extra $\frac{\pi}{4}$ phase offset.

After this correction was made, we needed to account for the fact that raising our received signal to the fourth power caused us to loose information regarding the bits' signs. As a result, we needed to rotate this corrected constellation plot in multiples of $\frac{\pi}{2}$ in order to find the correct orientation, which was found by determining which orientation's header had the least amount of error when compared to the actual header values. After this correction was made we were able to extract, we were able to extrac the bits stored in both the imaginary and real components of the complex numbers as shown below in Figures 13 and 14.

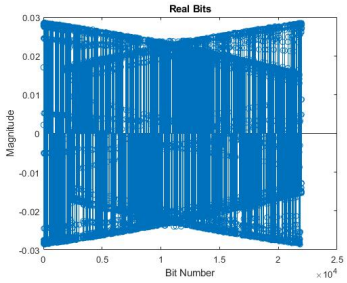

Figure 13: Real Component of received bits.

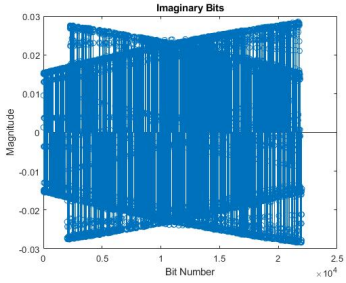

Figure 14: Imaginary component of received bits.

The two bit components are not completely square because there is still some frequency offset that we did not correct for. Below, Figure 15 shows a zoomed in version of the imaginary bits.

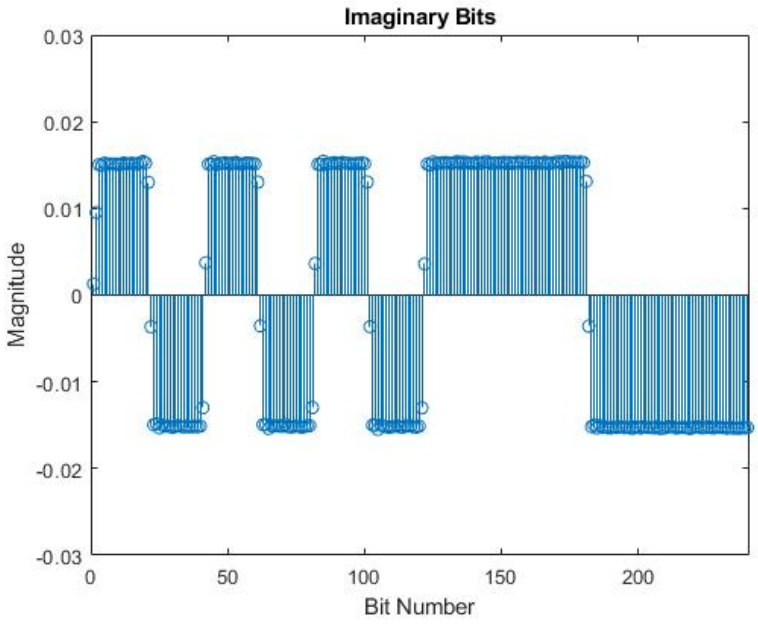

Figure 15: This figure shows the correct constellation plot after accounting for the extra $\frac{\pi}{4}$ phase offset.

In this figure one can clearly see the received bits convolved with a rectangular pulse of 20 samples which we used to send the signal. In order to extract the bits, the real and imaginary components simply need to be down sampled. To calculate our bit rate for 4-QAM we can perform the following calculation.

$$250k\frac{samples}{second} \times \frac{1}{20}\frac{symbols}{sample} \times 2\frac{bits}{symbol} = 25,000\frac{bits}{second}$$

Our received signal looks *very* clean as can be seen in Figure 15, which means that we can probably up the sample rate of our transmission or decrease the amount of samples per symbol.

4 Conclusion

We found that using QPSK has a slight advantage as compared to using BPSK due to its efficient usage of bandwidth. Comparing the plots of the headers above using BPSK and QPSK on the receiving end, we found that we could use QPSK to extract signal that produces no error and also matches closest to our transmitted header file. However, we do realize that our bit rate as calculated is 25,000 bits per second, which is low and we could probably increase it. Moreover, the channel that we are using is clean without noises and thus we are able to implement both BPSK and QPSK without any errors. As our results from the comparison of the transmitted and received signal shows no error, a possible area to explore would be to push for higher bit rate. Another possible extension would be use a raised-cosine instead of convolving with a rectangular pulse so that we have a better pulse shape that allows more error tolerance.

References

[1] M. Kavehrad T. D.C. Little N. Chi, H. Hass. *Visible light communications: Demand factors, benefits and opportunities*. IEEE Wireless Communications, 2015.

[2] P. Han Joo Chong S. Yongchareon D. Komosny S. Ur Rehman, S. Ullah. *Visible Light Communication: A System Perspective—Overview and Challenges*. MDPI, 2019.