Principles of Wireless Communications Lab 1: QAM

Mark Goldwater and Anusha Datar

February 18, 2021

1 System Overview

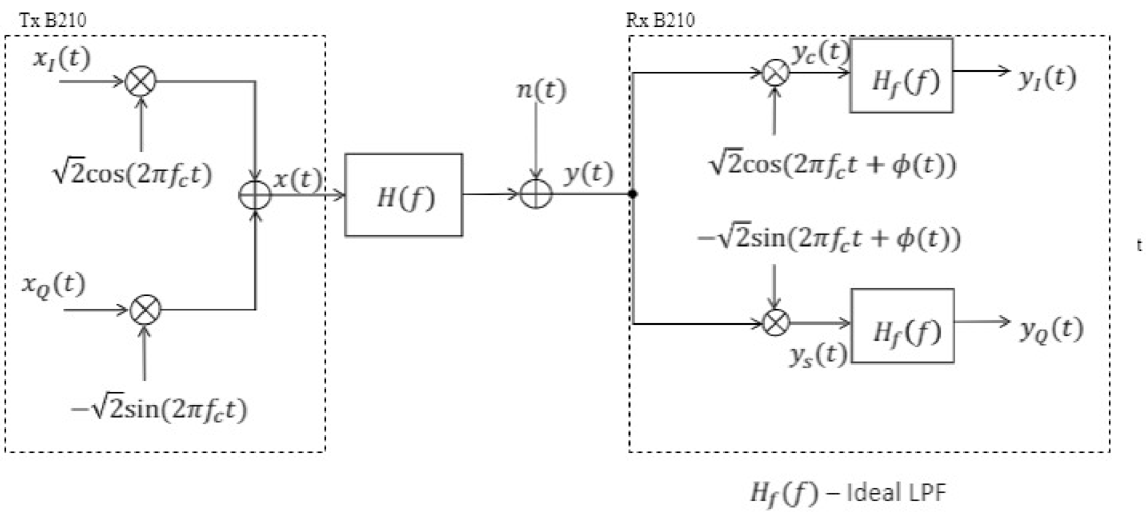

Figure 1: Block diagram of the system. The modulation and demodulation of the signals is handled by the B210 software defined radios.

1.1 Block Diagram Walk Through

Above is the block diagram for our 4-QAM communication system. At the beginning of the system, the in-phase and quadrature components of each two-bit element are multiplied by a cosine and negative sine carrier respectively so that they are able to pass through the channel. This yields the following signal to be transmitted where f_c is the carrier frequency.

$$x(t) = x_I(t)\sqrt{2}\cos(2\pi f_c t) - x_I(t)\sqrt{2}\sin(2\pi f_c t) \tag{1}$$

The signal $x(t)$ is then sent through the channel where additive noise is introduced, resulting in the output $y(t)$ via the equation below.

$$y(t) = (x \star h)(t) + n(t) \tag{2}$$

Lastly, in the receiving radio, $y(t)$ is demodulated to retrieve both y_I and y_Q by multiplying the received signal by their respective carriers and then applying a low-pass filter to extract the baseband signal as shown below.

$$y_c = y(t)\sqrt{2}\cos(2\pi f_c t + \phi(t)) \tag{3}$$

$$y_s = -y(t)\sqrt{2}\sin(2\pi f_c t + \phi(t)) \tag{4}$$

At this point, we encounter the problem that the carrier signals that we use to demodulate the message bit pairs have a phase offset which will end up cross-pollinating the in-phase and quadrature components. This way of modeling our system is a bit verbose and drawn out. In order to address this, let us remodel the system using the complex base-band representation. This way of viewing the system is also more in tune with how we will be sending and receiving data using the B210 radios.

1.2 Complex Base-Band Representation

To begin let us define the following two equations.

$$x_b(t) = x_I(t) + jx_Q(t) \tag{5}$$

$$y_b(t) = y_I(t) + jy_Q(t) \tag{6}$$

This is how we construct our data in MATLAB to send using the B210 radios. Using this model, we can represent the system using the following equation

$$y_b(t) = (h_b \star x_b)(t)e^{-j\phi(t)} + n_b(t) \tag{7}$$

where the exponential term accounts for the carrier phase offset ($\phi(t)$). In this model of the system, we have abstracted the modulation and demodulation of the message signals into a larger "channel." This provides a more compact and mathematically succinct representation of our system which will be helpful when using the B210 radios which take x_b as a system inputs and outputs y_b on the receiving end.

1.3 Physical Setup

Figure 2: The physical setup of the radio consisted of a transmit and receive B210 plugged into two separate comupters.

As seen above we used two computers to control the two B210 radios. We initiated the receiving end of the system, then began transmission of the bits, and then terminated the receiving end once all the bits had been sent. The results were written to a file which was further processed using MATLAB.

2 DT Flat Fading Channel and Digital Costas Loop

2.1 Phase Offset Cancellation [1]

In order to account for this phase offset in our modulating/demodulating carrier signals, we are going to take advantage of the DT flat fading model. Under the circumstances where the bandwidth of the transmit signal is much lower than the coherence bandwidth of the channel (the range of frequencies over which a channel frequency response can be considered to be constant), we can model the channel as a single complex coefficient as shown in the following equation using the complex base-band model.

$$y_b(t) = hx_b(t)e^{-j\phi(t)} + n(t) \tag{8}$$

If we assume that $x_b(t)$ comes originates from a complex DT signal (which it does) and that we sample y_b to produce a complex DT signal, we can arrive at the following model of our system.

$$y[k] = hx[k]e^{-j\phi[k]} + n[k] \tag{9}$$

In the above equation, $x[k]$ and $y[k]$ are complex DT signals, h is a complex number, $\phi[k]$ is a DT signal related to $\phi(t)$, and $n(t)$ is a sampled version of the noise in the system.

With this model of our system, we can take an approach to carrier synchronization called a **Digital Costas Loop** which takes advantage of FFT relationships to correct the phase and frequency offset of the carriers and channel. First, let's express the channel h, a complex number, in its exponential form.

$$h = \mid h \mid e^{j\angle h} \tag{10}$$

Then, let us apply the following substitution

$$\psi[k] = \phi[k] - \angle h \tag{11}$$

where $\psi[k]$ represents the combination of the phase offset from the oscillators at the transmitter and receiver not being matched, and the phase offset introduced by the channel. We can then substitute our exponential channel coefficient into Equation 9 and combine the exponential terms as follows.

$$\begin{aligned} y[k] &= (\mid h \mid e^{j\angle h})x[k]e^{-j\phi[k]} + n[k] \\ &= \mid h \mid x[k]e^{-j(\phi[k]-\angle h)} + n[k] \\ &= \mid h \mid x[k]e^{-j\psi[k]} + n[k] \end{aligned} \tag{12}$$

Next, using a computer we are able to approximate the the magnitude of the channel using the RMS of the received data.

$$\bar{y}[k] = x[k]e^{-j\psi[k]} + \frac{1}{\mid h \mid}n[k] \tag{13}$$

Suppose we are able to generate an estimate of this phase offset, $\hat{\psi}[k]$, then we could multiply $\bar{y}[k]$ to produce an estimate of $x[k]$ that accounts for phase offset which we call $\hat{x}[k]$ as follows.

$$\hat{x}[k] = \bar{y}[k]e^{j\hat{\psi}[k]} = x[k]e^{-j\psi[k]}e^{j\hat{\psi}[k]} + \tilde{n}[k] = x[k]e^{-j(\psi[k]-\hat{\psi}[k])} + \tilde{n}[k] \tag{14}$$

where $\tilde{n}[k] = \frac{1}{|h|}n[k]$. Additionally, when $\hat{\psi}[k] \approx \psi[k]$, we have

$$\hat{x}[k] \approx x[k] + \tilde{n}[k] \tag{15}$$

effectively canceling the phase offset. However, now we need to derive an estimate of the phase offset which we will do using the FFT.

2.2 Phase Offset Estimation [1]

Let's assume that we've already normalized the received data and that the noise is negligible such that we have the following equation.

$$\bar{y}[k] = x[k]e^{-j\psi[k]} \tag{16}$$

However in this case, $x[k] = \pm 1 \pm 1j$ which can be represented by the complex exponential $x[k] = e^{j\frac{\pi}{4}+jn\frac{\pi}{4}}$ where $n \in 0, 1, 2, 3, ...$ However, for simplicity, we will represent our message with $x[k] = e^{j\frac{\pi}{4}}$ because we will need to deduce the correct orientation of the data resulting from this approach eliminating the phase information of our data. Thus, we have the following.

$$\bar{y}[k] = e^{j\frac{\pi}{4}}e^{-j\psi[k]} \tag{17}$$

Next, suppose that our phase offset follows the following model

$$\psi[k] = f_\Delta k + \theta \tag{18}$$

where we incorporate a frequency offset (f_Δ) and a constant phase offset (θ). Then, let us define a new signal as follows.

$$\begin{aligned} s[k] &= (\bar{y}[k])^2 \\ &= (e^{j\frac{\pi}{4}})^4(e^{-j\psi[k]})^4 \\ &= (e^{j\pi})(e^{-j4(f_\Delta k+\theta)}) \\ &= e^{-j4\theta+j\pi}e^{-j4(f_\Delta k)} \end{aligned} \tag{19}$$

When we take the FFT of the above equation, we get an impulse with phase $4\theta + \pi$ positioned at $4f_\Delta$. Thus, we have our approximation as follows.

$$\hat{\psi}[k] = \frac{k_{peak}}{4}k + \frac{\text{angle}(FFT(s[k_{peak}])) + \pi}{4} \tag{20}$$

Lastly, when we raise our data to the fourth power, we loose all original phase information contain in our signal. As a result, we prefix a known header to our data and try all possible rotations of the data by multiplying by $e^{j\frac{\pi}{2}n}$ where $n \in \{0, 1, 2, 3\}$.

3 Costas Loop [2]

Alternatively, we can address the phase and frequency offset by using an implementation of a proportional-integral controller known as a Costas Loop. We can characterize this sum of the phase and frequency error as ψ. First we can initialize our approximation of the offset to zero $\hat{\phi} = 0$, and then apply our correction to the normalized data as shown below

$$\hat{x}[k] = \bar{y}[k]e^{j\hat{\psi}[k]} \tag{21}$$

we can then compute an error approximation by finding the product of the real and imaginary component of the signal as shown below.

$$\hat{e}[k] = -\Re(\hat{x}[k])\Im(\hat{x}[k]) \approx (\psi - \hat{\psi}) \tag{22}$$

From there, we can update the proportional-integral controller using the previous error and the current error with some tuned parameters β and α that we update for each new error value (denoted here as $e[k]$).

$$d[k] = \beta e[k] + \alpha \sum_{l=0}^{k} e[k]$$

$$\psi[k+1] = \psi[k] + d[k]$$

This update of the estimated ψ value allows the system to both lock on capture any drift in the overall error over time. We can then take our new $\hat{\psi}$[k] and use Equation 21 to begin another update iteration. Note that we implemented this loop in software, but we were not able to adequately tune it to match or exceed the performance of the FFT-based correction.

4 Software Architecture

To organize our code, we created a single function for preparing the data for transmission and another single function for processing the received data.

Our transmit function generates a series of random bits, adds a header made up of known a known number of known data bits, and then multiplies the signal by a pulse of known width.

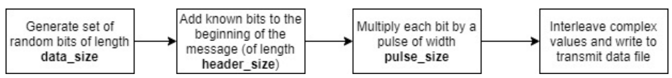

Figure 3: Transmit Data Function Diagram

Our receive data function calls a series of more modular functions that correct the errors present in the raw received data and convert the information back into bits.

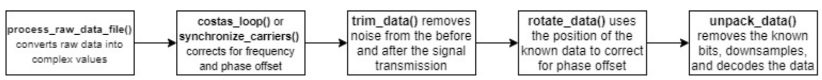

Figure 4: Receive Data Function Diagram

Our testing and development workflow involved running the data preparation function (which saves the file containing the data to transmit to the computer), running a command on one radio to receive data and a command to transmit the data from the file using the radio, and then running the data processing function on the received data file. This process assumes that both the sender and receiver have a shared expectation of the number of data bits, the number of bits in the header, and the width of the pulse (in bits).

Our source code and documentation are available at `https://github.com/anushadatar/usrp-qpsk`.

5 Results

To demonstrate the functionality of our system, we transmitted 1000 random bits using a sample rate of 2×10^6 and a pulse width of 50. Figure 5 displays the in-phase component of the original transmitted signal in the time domain and shows that the signal contains a set of positive and negative values multiplied by a pulse.

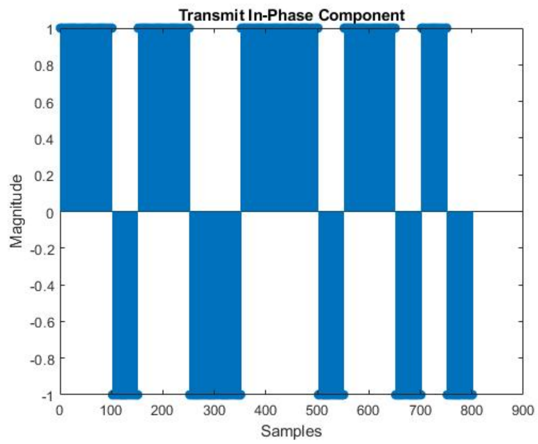

Figure 5: Raw Transmitted Signal

After generating this file, we transmitted its contents across the channel to a receiving radio. Figure 6 shows the in-phase component of the raw received signal in the time domain the interspersed nature of the signals reflects the fact that the real and complex component of the original signal have become interspersed. The constellation plot of the raw received signal shown in figure 7 more visually displays this smearing of the real and complex value.

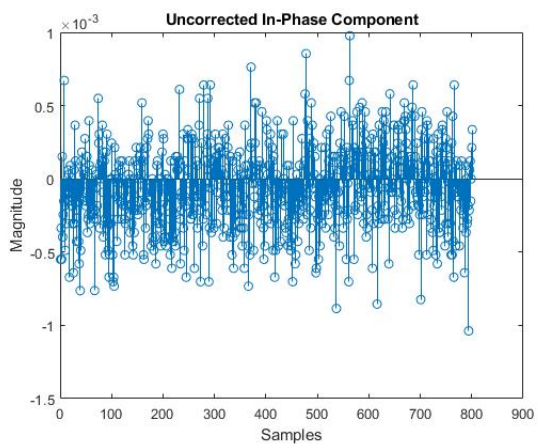

Figure 6: In-Phase Component of Raw Received Signal in the Time Domain

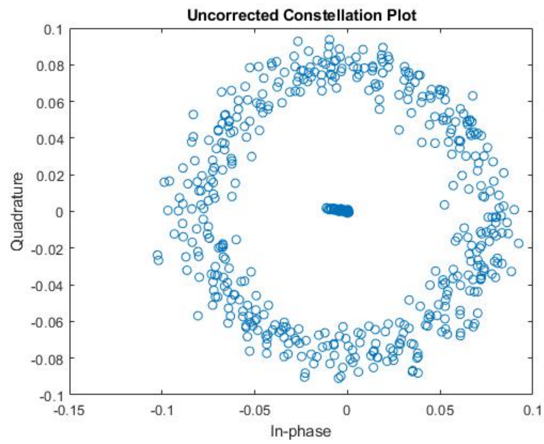

Figure 7: Constellation Plot of Raw Received Signal

Correcting the raw received data allows us to extract both the in-phase and quadrature components separately. Figure 8 shows the in-phase component of the corrected received signal in the time domain - the discrete blocks, while still imperfect, now represent the time-domain plot of the original pattern. Meanwhile, Figure 9 shows the constellation plot of the corrected received signal, which now contains four discrete points that correspond to each of the transmitted symbols.

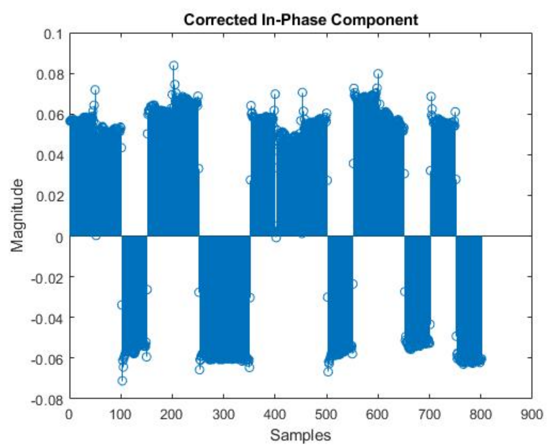

Figure 8: In-Phase Component of Corrected Received Signal in the Time Domain

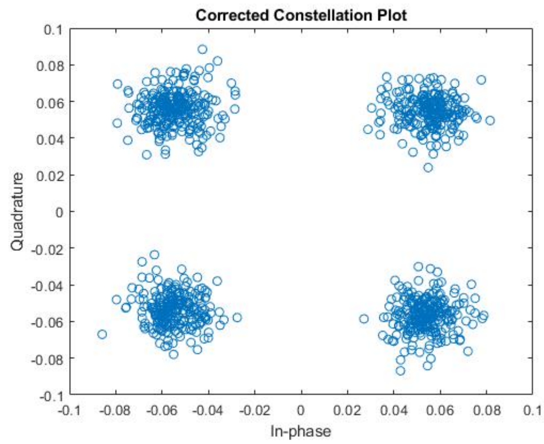

Figure 9: Constellation Plot of Corrected Received Signal

On this iteration, our data rate is 80 kbps. We determined the data rate via the following calculation.

$$\frac{2 \times 10^6 \text{ samples}}{1 \text{ second}} \cdot \frac{1 \text{ symbols}}{50 \text{ samples}} \cdot \frac{2 \text{ bits}}{1 \text{ symbol}} = 8 \times 10^5 \frac{\text{bits}}{\text{second}} \text{ or } 80 \text{ kbps} \tag{23}$$

We calculated the error rate of our system by comparing the data file we transmitted with the corrected received data and determined our system (in this case) has 0 error which means that it could likely be pushed to achieve a faster data rate.

6 References

[1] Govindasamy S., Lohmeyer W. *Carrier Synchronization using FFT Approach.*

[2] Govindasamy S., Lohmeyer W. *Carrier Synchronization.*