16-QAM Demodulator Block

ELEC4110 Course Project

Pranav Gupta

The Demodulator Design

In this project, a 16-QAM demodulator block was designed to accurately map the transmitted symbols into the respective bits. The input to this demodulator is the received waveform that has already undergone the timing synchronization and channel equalization processes, and ultimately, is a 4601x1 array of symbols. The first 251 symbols are discarded as they are the preamble of the frame – which is used in synchronization and channel estimation. The remaining 4350 symbols per frame are each to be decoded into 4 bits. This is because $M = 16$ and $\log_2 16 = 4$, meaning there are 16 possible symbols corresponding to the 2^4 possible combinations of the 4 bits.

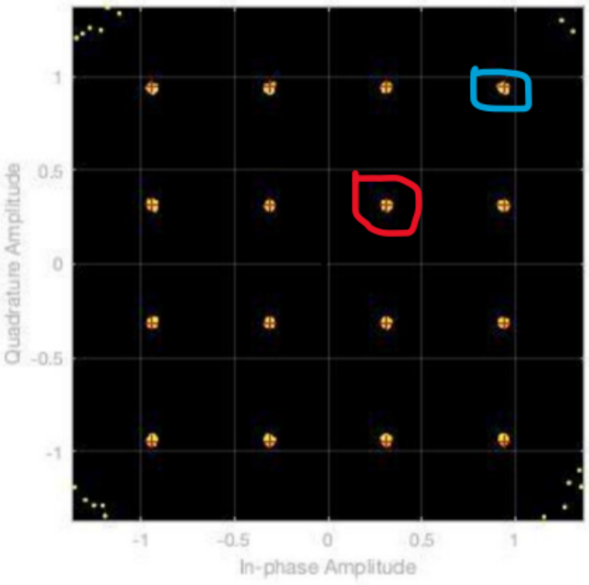

Figure 1: Constellation Diagram for 16-QAM

This is represented by the 16 unique signals in the constellation diagram in Figure 1. These signals are arranged in an equally spaced 4x4 grid pattern. To calculate the position of these points, assume α to be the projection on the real axis (In-Phase Amplitude) of the distance between the point marked in red and the origin of the axes. Since the points are equally spaced, α is also equal to the projection of the distance on the imaginary axis (Quadrature Amplitude). Hence, this point is at (α, α), and this is the true for all four points closest to the origin. Since the points are equally spaced, the point marked in blue is at $(3\alpha, 3\alpha)$. Therefore, the remaining two points in that quadrant are at $(3\alpha, \alpha)$ and $(\alpha, 3\alpha)$.

The average symbol (s) energy (E_s) is found by the equation:

$$E_s = \frac{1}{M} \sum_{i=0}^{M-1} \|\vec{s}\|^2,$$

$$E_s = \frac{1}{16} \times 4 \times [(\alpha^2 + \alpha^2) + 2((3\alpha)^2 + \alpha^2) + ((3\alpha)^2 + (3\alpha)^2)] = 10\alpha^2.$$

Since the average power of the signals is set to 1W,

$$\alpha = \sqrt{\frac{1\text{W}}{10}} \approx 0.3162\text{W}.$$

The coordinates of these symbols in the I-Q space are first defined in the program. The demodulator implemented essentially computes distance between a received I-Q symbol (complex number) and the symbols described above, and determines the symbol with the minimum distance to represent the received I-Q symbol. To implement this, a 4350x16 matrix is used that stores the distance values for all symbols in the sequence. The min() function is used to find the arguments (i.e. the symbols above) with the smallest distance for each transmitted symbol. After mapping the received symbols to the 16 pre-defined symbols, a for-loop iterates through all the received symbols and a switch statement is used to map each symbol to the 4 bits it represents in Gray Coding. Finally, this 17400-bit sequence is outputted as the final demodulated information bits.

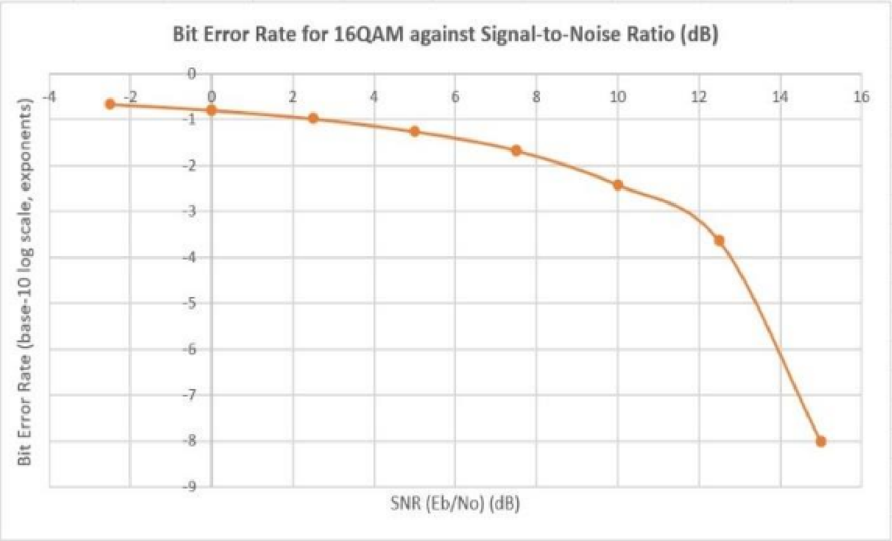

Figure 2: Plot of Bit Error Rate Against Signal-to-Noise Ratio (Simulation)

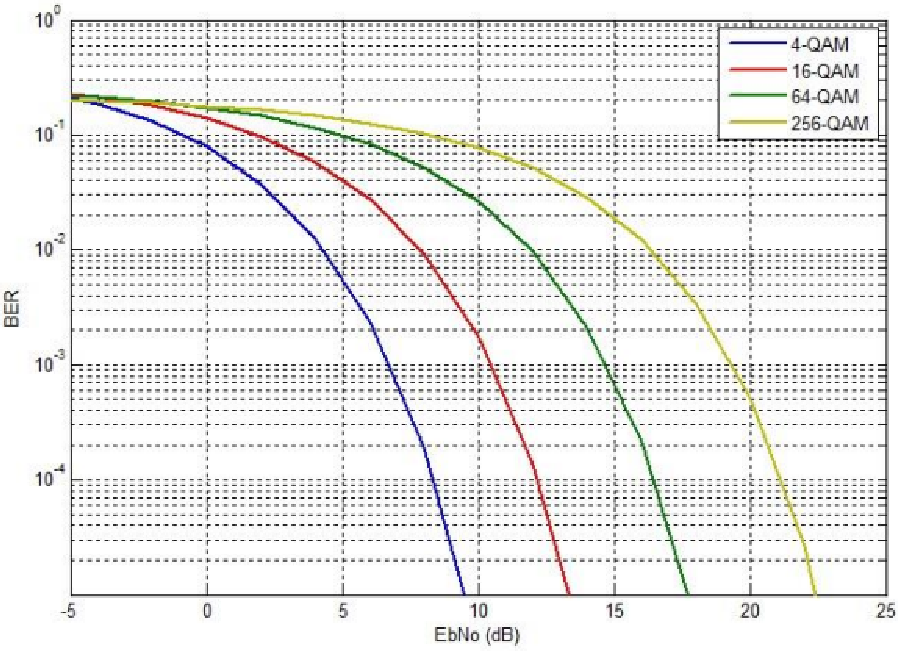

Figure 3: Plot of Bit Error Rate Against Signal-to-Noise Ratio (Reference)

Figure 2 shows the performance of this demodulator by plotting the Bit Error Rate to the received signal's Signal-to-Noise ratio. When compared to Figure 3, which shows the same graph referenced from external sources, the two plots are almost identical. Both have a similar curve shape, and the Bit Error Rate becomes negligible when the SNR is above 15dB.

WiFi IEEE 802.11b

The physical layer frame format of the 802.11b WiFi system can be divided into 3 major sections – namely the header, body, and trailer.

The Header starts with the Frame Control bits, which includes the protocol version used, the type and sub-type of commands to be executed (e.g. Authentication under Management Type, Acknowledgement under Control Type, etc.), and additional bits such as a retry bit to signal re-transmission of a packet, and ToDS and FromDS bits to signal if the packet's destination/source is a Wireless Distribution System. This is followed by the Duration bytes to restrict access to the medium for the duration specified in the Network Allocation Vector, which is carried by the Duration field. The Header can contain 4 addresses in a packet (not necessarily carrying 4 addresses in all frames). The 6-byte addresses enumerated from 1 to 4 represent the source address, destination address, transmitter address, and receiver address. Lastly, the sequence control bytes are to identify the message order and prevent duplicate frames.

The Frame Body is of variable length, and it contains the information bytes. The Body is followed by the 4-byte Frame Check Sequence (FCS), which is responsible for error detection in the overall frame being transmitted.

Top-Level Architecture of 802.11b

For the transmitter, there are two major blocks – the Frame Generation block and the Pulse Filter block. The information to be transmitted will first have to be split and fit into frames as specified in the section above. The bits will be converted into symbols for transmission. Moreover, this system is a multi-channel system (i.e. frames will be sent in parallel). In the Pulse Filter Block, the symbols are modulated and mapped into the complex plane and are upsampled for transmission.

For the receiver, the received packet will undergo a matched filter block to identify all the symbols. The sequence will then be passed to the Timing Synchronization block where it cross correlates the initial sequence to identify the internal boundaries of the frame and extract the information bits for demodulation. The synchronized received symbols will then be passed through a Channel Equalizer to neutralize the effects of any channel fading – by using the known initial sequence to determine the fading and correcting accordingly.

Finally, the information symbols will be inputted into the Demodulator to decode the information bits from the complex symbols used. The modulator used in 802.11b is a Complementary Code Keying (CCK) system in which there are 8 chips (each of which is using QPSK) to transmit symbols. The system determines 4 phase values based on the 8 symbols, each of the 8 having a unique combination of phase addition (e.g. first symbol's phase is the sum of all 4 phases, etc.). The demodulator will be implemented such that the phase values can be determined, and consequently, the symbols can be identified and mapped into information bits.

Bottleneck of the System

I strongly believe that the bottleneck of this system is in the Channel Estimator instead of the Demodulator. There are two sources of disturbances in the symbol transmission in this simulation – the flat-fading channel $H(t)$ and the Additive White Gaussian Noise (AWGN).

For the demodulator, it is assumed that the transmitted symbols are independent, the previous and future symbols should not affect the probability of a symbol. Plus, as we assume the symbols are equiprobable, one cannot increase or decrease the decision space solely based on symbol probability to identify a symbol in the I-Q space. Moreover, the AWGN is White noise, which means that the noise addition is equiprobable for all directions (noise cloud is spherical, instead of ellipsoid that favours certain regions around a symbol). Therefore, there are limited opportunities to further optimise the decision boundaries in the demodulator.

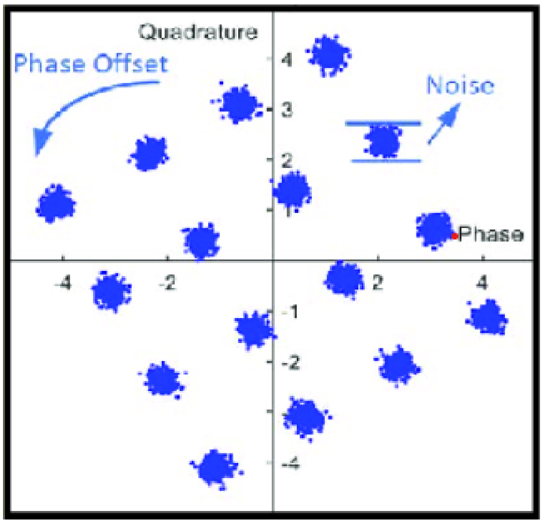

Figure 4: Example of 16-QAM Plot with Phase Shift and Scaling Effects

On the other hand, the flat-fading channel produces disturbances in the transmitted sequence with significant variance in the scaling and phase shifts. As depicted in Figure 4, in such shifts, all the transmitted symbols are rotated in the same direction. The minimum distance detection algorithm is bound to fail in many cases if the transmitted symbols are shifted towards/away from the correct symbol on the I-Q plane. Therefore, the channel estimator needs to be further optimized for significant improvements in the Bit-Error Rate Performance.