**EE 1473 – Digital Communication Systems**

**Spring 2019 Simulation Project**

**16-QAM**

Josh Eaton

18 April 2019

**Introduction**

Quadrature Amplitude Modulation (QAM) is a form of digital modulation that

**Procedure**

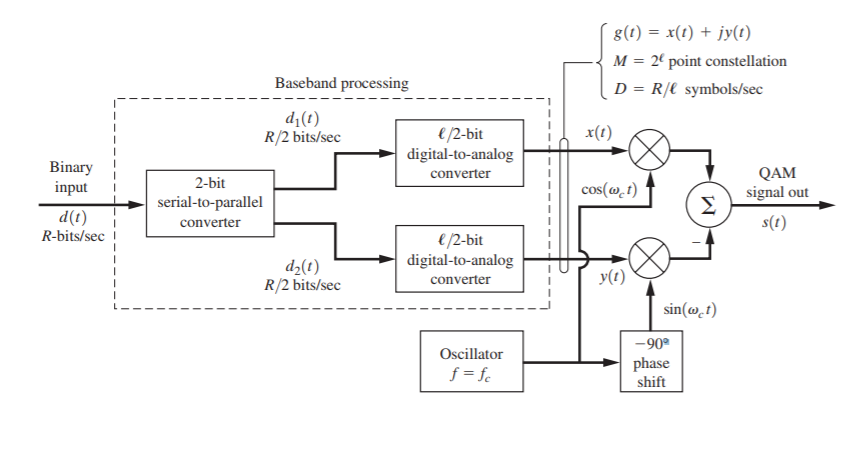
*Generation of Random Data*

The bits to be used as random data for the purposes of this simulation were generated using the *rand* function in MATLAB to generate values between 0 and 1, and these values were compared to a threshold of 0.5 to be considered as a digital 1 or 0.

*Baseband Signaling*

*Digital Modulation*

To implement 16-QAM for this simulation, the process shown in **Figure 1** below was followed.



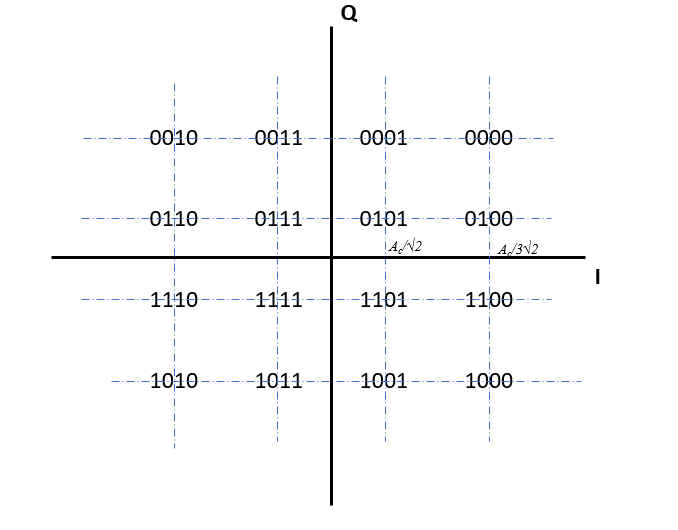
**Figure 1:** Rectangular QAM Modulator [1]

The binary input d(t) shown in the figure was supplied by the random data previously generated. The serial-to-parallel conversion was carried out by dividing the array of data bits into 4-bit words. The first two bits in each word were used as the input to create the in-phase channel x(t), and the last two bits in each word were used as the input for the out of phase y(t).

The digital-to-analog conversion for each 2-bit pair was assigned as shown in **Table 1** below, where A­c represents the carrier amplitude.

|  |  |
| --- | --- |
| **Table 1: 2-bit DAC Assignment** | |
| **Bit Pair** | **Analog Level** |
| 00 | -Ac/√2 |
| 01 | -Ac/(3√2) |
| 11 | Ac/(3√2) |
| 10 | Ac/√2 |

These assignments were chosen so that each word would be mapped to the Gray-coded constellation shown in **Figure 2** below. The levels were chosen such that the corners of the constellation would occur at a magnitude of Ac2 from the origin.



**Figure 2:** 16-QAM constellation

*Channel Effects*

For this simulation, the Channel was considered only to have Additive White Gaussian Noise (AWGN) with no other distortion.

This was modeled in MATLAB by using the *randn* function to generate an array of random values the same length as the modulated signal. The *randn* function provides values from the standard normal distribution, so to achieve a desired noise variance σ­2 the values from *randn* were each multiplied by that value.

The array of generated noise was element-wise added to the modulated signal to simulate the AWGN of the channel.

*Demodulation*

*Optimal Detection*

*Bit Error Rate*

*Power Spectral Density*

*Channel Coding Comparison*

**Results**

*Generation of Random Data*

*Baseband Signaling*

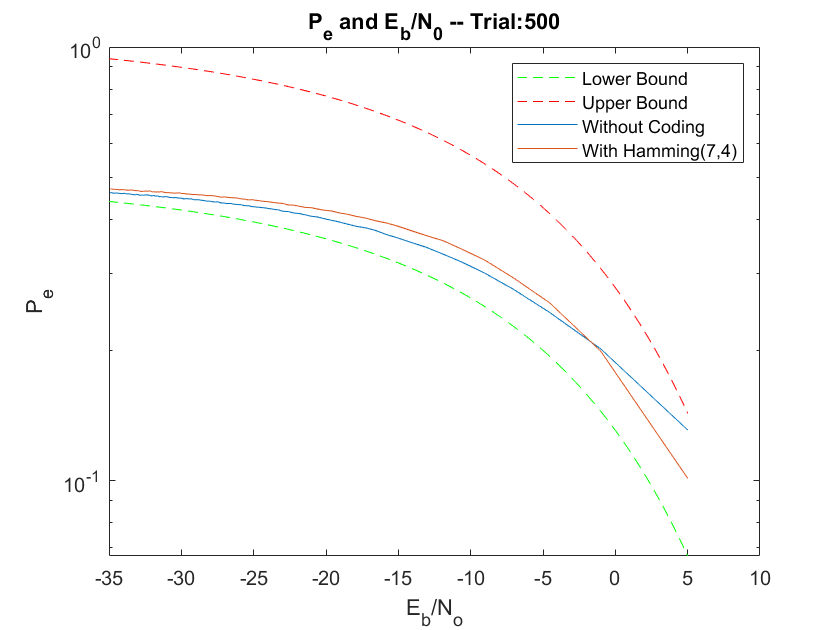
*Digital Modulation*

*Channel Effects*

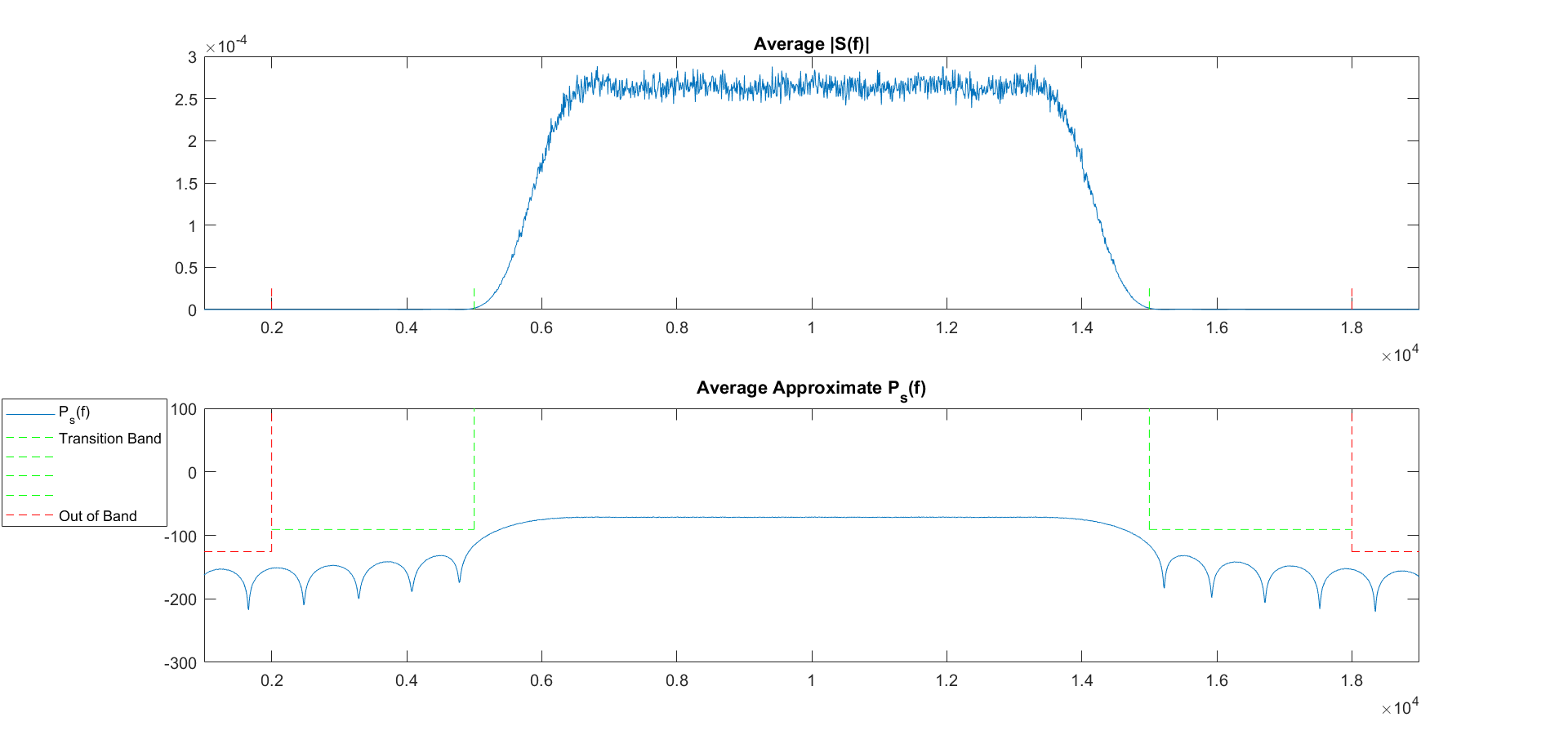
*Demodulation*

*Optimal Detection*

*Bit Error Rate*

**

*Power Spectral Density*

**

**Discussion**

**Conclusions**

**References**