16-QAM-OFDM Transmitter and Receiver Simulation in MATLAB(project report)

Name – Akshay Modak

Reg no – 180907722 Roll no – 55

Batch – 3

Abstract - The objective of the project is to demonstrate the transmission and reception of data in 16-QAM-OFDM channel with AWGN by simulation in MATLAB. At the transmitter random numbers are generated that correspond to the 16-QAM symbols. The effect of the AWGN channel for different SNRs on these symbols is demonstrated by means of scatter plots. At the receiver the symbols are estimated using maximum likelihood estimation. The estimated symbols are compared with the original symbols to see the effectiveness of this system.

I. Introduction

Quadrature Amplitude Modulation and Orthogonal frequency division multiplexing are technologies that are widely used in modern communication systems. In this project a simple transmitter and receiver are simulated in MATLAB. The different functions used for this simulation and the results obtained from the same are explained in the following sections.

II. Transmitter

Random numbers from the range 0 to 15 are generated. This is because 16-QAM encodes 4-bits at a time. The length of this array of random numbers is 48. This will act as the input data for the transmitter. The random numbers are generated using the *randi()* function in MATLAB. These functions are then modulated using 16-QAM modulator. The modulation is done using the *qammod()* function. Which modulates the symbols according to the constellation diagram shown in figure 1.

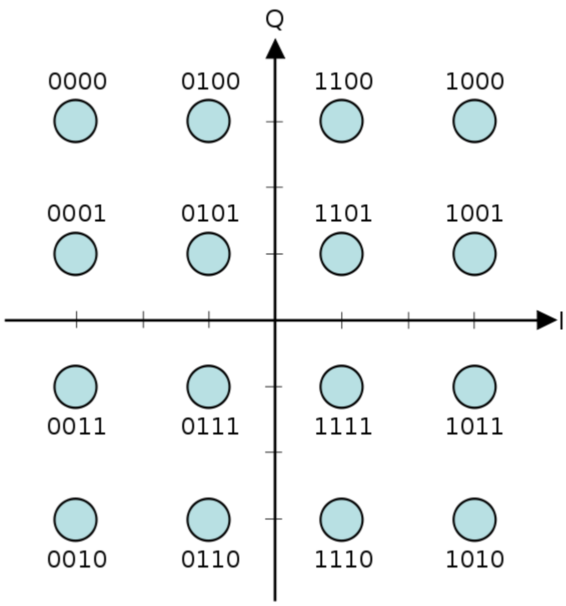

fig 1. Constellation diagram used by MATLAB

Since the transmitter uses a 64-point IFFT the vector is padded with zeros. These zeros can be ignored at the receiver as they do not carry any data. After the IFFT is taken a time domain signal is obtained which is the sum of orthogonal sinusoids which is the key to OFDM using bandwidth effectively. However, before transmission a cyclic prefix is added to prevent inter symbol interference. To add this cyclic prefix the end of the signal is prepended to itself. The amount that is prepended depends on the nature of the channel. For this simulation the last 16 samples are used as the cyclic prefix. The signal is now ready to be transmitted.

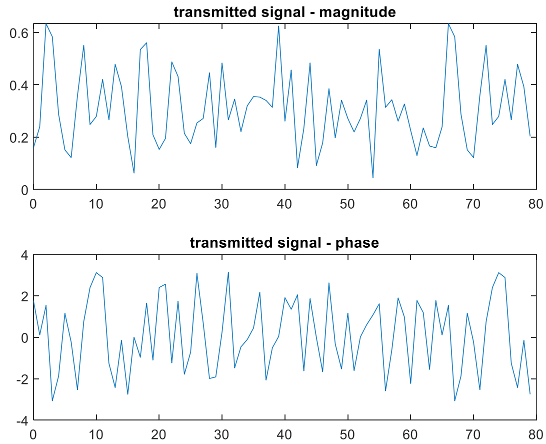

fig 2. Transmitted signal

The MATLAB code for the transmitter is shown below.

```
% TRANSMITTER
x = randi([0 , 15] , 48 , 1)';
q = qammod(x , 16);
% 0 pad q to make it 64 symbols
q = [q zeros(1 , 16)];
% take ifft
tx = ifft(q , 64);

% add cyclic prefix to this signal
cp_length = 16;
tx = [tx(: , end-cp_length+1:end) tx];
figure();
```

MATLAB code for Transmitter.

III. Channel

For this simulation an AWGN channel is simulated. The *awgn()* function is used for this purpose. This function takes the signal to which the noise has to be added and the signal to noise ratio. The effects of the channel on the transmitted signal is shown in fig 3.

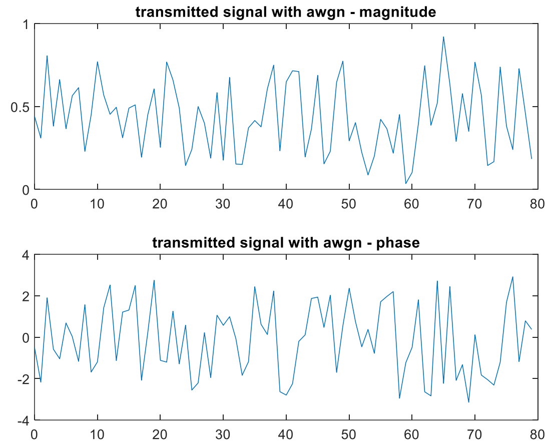

fig 3. Transmitted signal with AWGN

The code for adding noise to the signal is as shown below.

```
% CHANNEL
% add awgn to transmitted signal
snr = 20;
tx_noisy = awgn(tx , snr);
```

MATLAB code for AWGN channel

IV. Receiver

At the receiver the first step is to remove the cyclic prefix. After this the signal is passed through an FFT block outputs the 16-QAM symbols. However, These symbols have noise added to them. The effect of noise on the symbols for different values of SNR is shown in figure 4.

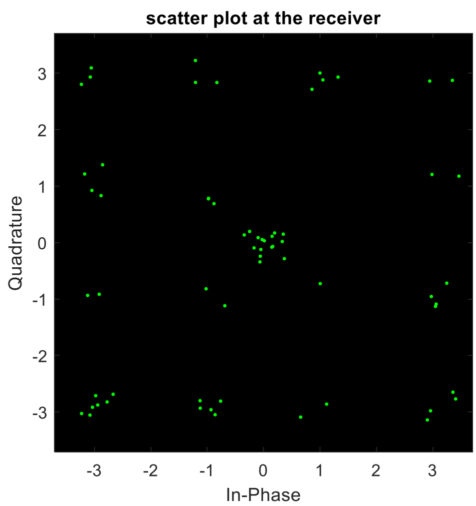

fig 4(a). SNR = 30 dB

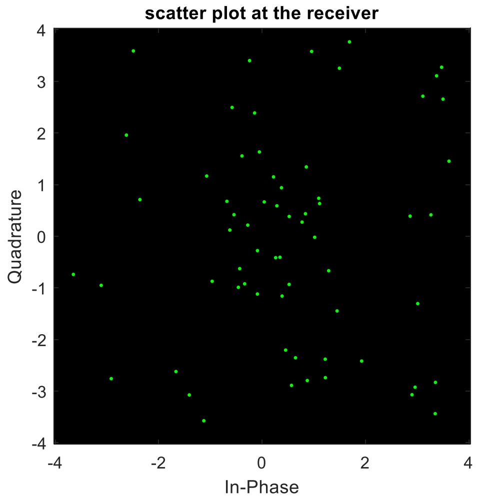

fig 4(b). SNR = 20 dB

These symbols have to be passed through an estimator before they can be demodulated. maximum likelihood estimation is used for this purpose. In maximum likelihood estimation the Euclidian distance between a received symbol and the actual 16-QAM symbols is calculated for each of the 16-QAM symbols. The minimum of this quantity is then chosen to be the estimate. The implementation for this algorithm is as shown.

```
points = [qammod((0 : 15) , 16) 0]; % 16-QAM points
mindist = inf;
minpt = 0;
q_r_estimate = (zeros(1 , 64));
for k = 1 : length(q_r)
    mindist = inf;
    for m = 1 : length(points)
        if norm(q_r(k) - points(m)) < mindist
            mindist = norm(q_r(k) - points(m));
            minpt = points(m);
        end
    end
    q_r_estimate(k) = minpt;
end
```

MATLAB code to assign symbol based on minimum Euclidian distance

After the estimation step, the symbols can be demodulated. The demodulation is done using *qamdemod()* function in MATLAB. Which uses the constellation diagram shown in figure 1 to map the symbols back to data.

V. Conclusion

Thus, a simple transmitter and receiver for a 16-QAM OFDM system has been demonstrated. The code associated with each of the components has also been explained and the results from each stage in the system have also been shown.

The transmitted and received data for different SNR values has been compared in figure 5.

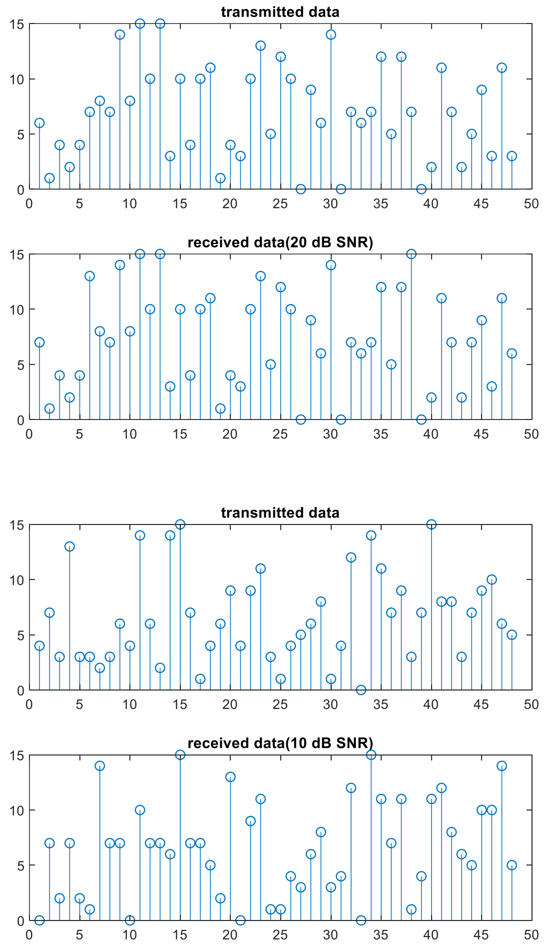

fig 5. Transmitted and received data

It can be seen that the received and transmitted data are quite similar which indicates that the transmitted data was received successfully.

REFERENCES

[1] Aditya K. Jagannatham, "Orthogonal Frequency-Division Multiplexing," in Principles of Modern Wireless Communication Systems Theory and Practice: McGraw Hill, 2015, ch. 7, pp. 230–266.

[2] Leslie Rusch. GEL7014 - Week 10b - OFDM implementation. (Nov 12, 2020). Accessed: March 3, 2021 . [Online Video]. Available: https://www.youtube.com/watch?v=oOOS9xDXLaU

[3] "Concepts of orthogonal frequency division multiplexing". keysight.com. http://rfmw.em.keysight.com/wireless/helpfiles/89600b/webhelp/subsystems/wlan-ofdm/Content/ofdm basicprinciplesoverview. htm

[4] "802.11 OFDM Overview". keysight.com. http://rfmw.em.keysight.com/wireless/helpfiles/89600b/webhelp/subsystems/wlan ofdm/Content/ ofdm 80211-overview.htm