

Presented to the

**Department of Electronics and Computer Engineering**

De La Salle University - Manila

**Term 2**,A.Y. **2022-2023**

In partial fulfillment

of the course

In **LBYEC4A-EK2**

**Project Paper**

Submitted by:

**Guimalan, Francis Xavier**

**Pozas, Raphael Dominic**

**Sabino, Cleajane**

Submitted to:

**Engr. Ramon Stephen Ruiz**

Implementation of 32-QAM Tx-Rx System using MATLAB

Guimalan, Francis Xavier, francis\_guimalan[@dlsu.edu.ph](mailto:francis_magallanes@dlsu.edu.ph), Department of Electronics and Computer Engineering De La Salle University-Manila, Philippines; Pozas, Raphael Dominic, raphael\_pozas[@dlsu.edu.ph](mailto:francis_magallanes@dlsu.edu.ph), Department of Electronics and Computer Engineering De La Salle University-Manila, Philippines; Sabino, Cleajane, cleajane\_sabino[@dlsu.edu.ph](mailto:francis_magallanes@dlsu.edu.ph), Department of Electronics and Computer Engineering De La Salle University-Manila, Philippines

***Abstract*— Quadrature Amplitude Modulation (QAM) is a digital modulation technique. It involves combining the Amplitude and Phase Modulation of the carrier wave. Thus, creating a more efficient use of the bandwidth. In this project, the student creates a 32-bit QAM Transmitter to Receiver System using MATLAB. The transmitted data was mixed with noise so that the system would be closer to real life. The results show successful transmission and recovery of the audio file.**

***Index Terms*— audio, Filter, MATLAB, Modulation, noise, QAM, signals**

## INTRODUCTION

Digital Modulation is an important modulation scheme in modern communication systems. It is more efficient than Analog Modulation for two major reasons. First, it can transmit large amounts of data over long distances without significant loss of the modulation signal. Second is that by encoding the information signal to the carrier signal, the signal becomes less susceptible to noise and interference [1].

There are many types of Digital Modulation schemes. The first one is the Amplitude-Shift Keying (ASK). This modulation scheme varies the carrier wave amplitude. The amplitude represents the digital data that must be transmitted. Since the amplitude carries the data, ASK is more susceptible to noise compared to other schemes. This scheme is used for transmissions where the environment is considered noiseless [2].

Another known type is the Frequency-Shift Keying (FSK). This varies the carrier frequency depending on the digital data. This becomes less susceptible to noise as changes in signal strength is unlikely affect the frequency. This scheme is used for wireless communication and digital broadcast systems [3].

The next major type is the Phase-Shift Keying (PSK). This modulates the carrier phase. This is more complex than the previous two as it divides the carrier phase by bits. This scheme is used in satellite communication systems and mobile phone networks [4]. Lastly, the Quadrature Amplitude Modulation (QAM) combines both ASK and PSK [5]. This is the modulation scheme that the system would use and would later be discussed in the next chapter.

The project aims to achieve the following objectives. First, is to create a Transmitter to Receiver system using Quadrature Amplitude Modulation using MATLAB. QAM Modulation technique has advantages over other digital modulation schemes like Amplitude-Shift Keying (ASK), Phase-Shift Keying (PSK) and Frequency-Shift Keying (FSK).

Second, the quality of the QAM Transmitter (Tx) – Receiver (Rx) system is based on the Bit Error Rate (BER) of the received signal. BER shows the number of incorrect bits within a sample number of bits [6]. The excellent range of BER for high-speed data rate applications is from 0% to 0.6%.

The last objective is to test the noise tolerance of the system. To do this, several Signal-to-Noise Ratio (SNR) are implemented. SNR is the ratio of the signal relative to the noise [7]. The higher the SNR means the lesser the noise is compared to the Signal. In this project, the noise that would be used is the Additive White Gaussian Noise (AWGN). This is the common type of noise used as a benchmark for communication systems because of its randomness and distribution across the signal [8].

## THEORETICAL CONSIDERATION

1. **Quadrature Amplitude Modulation**

Quadrature Amplitude Modulation (QAM) is a digital modulation scheme that uses two carrier waves. The amplitude and phase of each carrier wave represents the Digital Data [9]. For example, in 32-QAM, eight amplitude levels and four phase levels are used to represent the data.

The number of combinations also shows the number of bits of data that can be transmitted per symbol. Also due to its nature, it is also resistant to noise. QAM is the most used in high data rates applications such as Cable Modems and Wireless Local Area Networks.

1. **Filter**

Filters are electronic devices that are used to shape the frequency spectrum of a signal. In the context of QAM systems, it serves multiple purposes such as limiting the bandwidth and noise rejection. Normally, a Lowpass filter is for QAM systems. However, in this project, the author used Bandpass filter because in Radio Frequency applications, a Bandpass filter is more preferred [10].

In QAM systems, there is a need for a sharper roll-off in the stopband to avoid distorting the QAM signal. A Butterworth filter is not preferred despite having a smooth frequency response because of its slow roll-off. In this project, the author used a Chebyshev Type 1 filter which has a sharper roll-off curve [11].

1. **Fast Fourier Transform**

Fast Fourier Transform (FFT) is an efficient algorithm for calculating the Discrete Fourier Transform (DFT) of a sequence. This is needed to transform a digital signal from its time-domain into its frequency-domain [12]. This algorithm is needed for the project so that the author can perform signal analysis and filter design.

1. **Hypothesis**

The author expects that the input signal will be recovered as an output of the receiver. Since the Additive White Gaussian Noise is impossible to filter out completely, then the output signal received should have minimized background noise.

Regarding the Signal-to-Noise Ratio (SNR) and Bit Error Rate (BER), the author expects that both values would have an inverse linear relationship since the SNR indicates a stronger signal relative to noise and BER indicates the incorrect number of bits in each received bits. Due to this, the author expects a Bit Error Rate between 0 to 0.6%, which is the standard of good rate while having an inverse linear relationship with the SNR.

1. METHODOLOGY

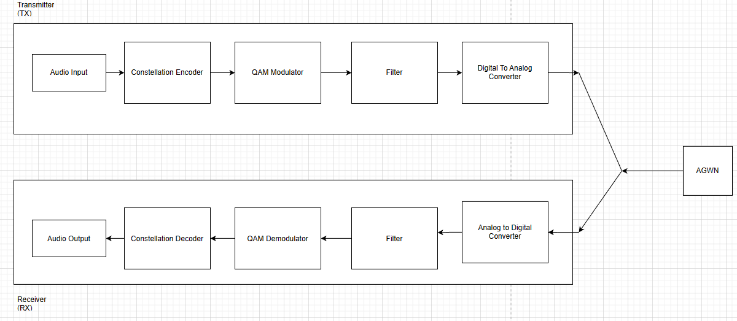


FIGURE 3.1. Input-Process-Output (IPO) model

Figure 3.1 shows the general process of the MATLAB code. First, the input was created by recording the signal using MATLAB. As shown from the Figure, the Transmitter block is composed of an encoder, QAM Modulator and Filter. The DAC and ADC blocks are not needed since the system is made using MATLAB. After recording the signal, it was plotted in its Time and Frequency Domains as shown in Figure 3. The process involved using Fast Fourier Transform for the conversion. This was needed so that the modulation and filter parameters can be obtained. After the necessary parameters have been obtained, the author encoded the audio signal into its discrete form. The syntax *uencode()* was used so that the input is stored in symbols. The next step was designing a QAM modulator block. The modulation was done using *qammod()*. Before transmitting the signal, a bandpass filter was created to limit the frequency band of the signal. The code and results for the transmitter block are shown from Table 3.1 to Table 3.5

TABLE 3.1. Input Signal Generator Code

|  |
| --- |
| %Signal Generator  fs = 8000;  fc = 600; % Carrier frequency, Limit fc<800 to avoid freqdomain aliasing  nbits = 8; % 8 bits is enough;  audioduration = 3;  ini\_phase = 0; % Phase initial  signal0 = audiorecorder(fs,nbits,1,0);% device id is zero (Default Mic)  recordblocking(signal0,audioduration);  speech\_data = getaudiodata(signal0,'int16') % save as a int16 array  input\_signal\_location = 'C:\Users\Xavier\Desktop\LBYEC4A\inputaudio.wav'  audiowrite(input\_signal\_location,speech\_data,fs)  input\_audio = audioread('inputaudio.wav');  [inputaudio, Fs] = audioread('inputaudio.wav');  st = length(inputaudio);  time = st/Fs;  t = (0:1/Fs:time-1/Fs); %Time Domain  [Nf,P] = size(inputaudio);  f = (-Fs/2:Fs/(Nf-1):Fs/2);  input\_fft = fftshift(fft(inputaudio)); %Frequency Domain  figure(1)  subplot(211);  plot(t,inputaudio); %Time Domain  xlabel('Time (s)');  ylabel('Amplitude');  title('Input Signal in Time Domain');  subplot(212);  plot(f,abs(input\_fft)); %Frequency Domain  xlabel ('Frequency (Hz)'); ylabel('Amplitude'); xlim([0 max(f)]);  title('Input Signal in Frequency Domain'); |

TABLE 3.2. Analog to Digital Encoder Code

|  |
| --- |
| % Encode the Analog Signal to Discrete Signal  N = 5;  M = 2^N;  yy = uencode(inputaudio,N);  figure(2)  subplot(211);  plot(t,yy); %Time Domain  xlabel('Time (s)');  ylabel('Amplitude');  title('Encoded Signal in Time Domain');  modfft = fft(yy);  subplot(212);  plot(f,abs(modfft)); %Frequency Domain  xlabel ('Frequency (Hz)'); ylabel('Amplitude'); xlim([0 max(f)]);  title('Encoded Signal in Frequency Domain'); |

TABLE 3.3. Modulator Code

|  |
| --- |
| % Modulate the Signal using QAM 32  modulatedsig = qammod(yy,M);  figure(3)  scatterplot(modulatedsig);title('QAM Modulated Signal'); |

TABLE 3.4. Bandpass Filter Code

|  |
| --- |
| %  fs = 8000;  Fn = fs/2;  Wp = 3900/Fn;  Ws = 3000/Fn;  Rp = 1;  Rs = 150;  [n,Wn] = cheb1ord(Wp, Ws, Rp, Rs);  [z,p,k] = cheby1(n, Rp, Wn) ; % Chebyshev 1 filter  [sosb,gb] = zp2sos(z,p,k);  freqz(sosb, 2^20, Fs) % Filter Bode Plot  tx\_filteredsig = filtfilt(sosb, gb, modulatedsig);  scatterplot(tx\_filteredsig);title('QAM TX Filtered Signal'); |

TABLE 3.5. Amplifier Code

|  |
| --- |
| % Amplify the Signal for better tranmission  tx\_filteredsig = 10\*tx\_filteredsig |

After transmitting the signal, the author mixed the QAM Signal with additive white gaussian noise (AWGN) using *awgn()* function. This is to test the system with more real-life factors. There are three values of Signal-to-Noise Ratio that were used. These are 10, 50 and 100 dB. The author recovered the noisy signals for comparison with the input and output signals. The code is shown in Table 3.6.

TABLE 3.6. AWGN Code

|  |
| --- |
| % Add Noise (show constellation diagram with different S/N)  SNR0 = 100;  snr100 = awgn(tx\_filteredsig, SNR0);  scatterplot(snr100);title('QAM Modulated Signal (100dB SNR)');  SNR1 = 50;  snr50 = awgn(tx\_filteredsig, SNR1);  scatterplot(snr50);title('QAM Modulated Signal (50dB SNR)');  SNR2 = 10;  snr10 = awgn(tx\_filteredsig, SNR2);  scatterplot(snr10);title('QAM Modulated Signal (10dB SNR)');  % Recover the Noisy Signal  Demodulatednoisysnr100 = qamdemod(rx\_filteredsignal100,M);  Demodulatednoisysnr50 = qamdemod(rx\_filteredsignal50,M);  Demodulatednoisysnr10 = qamdemod(rx\_filteredsignal10,M);  Noisysnr100 = udecode(uint32(Demodulatednoisysnr100),N);  Noisysnr50 = udecode(uint32(Demodulatednoisysnr50),N);  Noisysnr10 = udecode(uint32(Demodulatednoisysnr10),N);  receivedsignalfile = 'C:\Users\Xavier\Desktop\LBYEC4A\noisysnr100.wav';  audiowrite(receivedsignalfile,Noisysnr100,fs);  receivedsignalfile = 'C:\Users\Xavier\Desktop\LBYEC4A\noisysnr50.wav';  audiowrite(receivedsignalfile,Noisysnr50,fs);  receivedsignalfile = 'C:\Users\Xavier\Desktop\LBYEC4A\noisysnr10.wav';  audiowrite(receivedsignalfile,Noisysnr10,fs); |

The receiving block is composed of a Filter, Amplifier, QAM Demodulator. The filter used is the same code created in the Transmitter block. After filtering the signal, it was amplified to increase the amplitude for some losses in the transmission and filter process. The next step was to demodulate the filtered signal. The author used *qamdemod()* for the QAM demodulation. The resulting signals were decoded to convert into its numerical value and were plotted in their respective time and frequency domains. The resulting signals were also saved as .wav files for audio comparison.

TABLE 3.7. Filter Code

|  |
| --- |
| % Filter the noisy signal  rx\_filteredsignal100 =filter(Wn,n,snr100);  scatterplot(snr100);title('QAM Modulated Signal (100dB SNR)');  rx\_filteredsignal50 =filter(Wn,n,snr50);  scatterplot(snr50);title('QAM Modulated Signal (50dB SNR)');  rx\_filteredsignal10 = filter(Wn,n,snr10);  scatterplot(snr10);title('QAM Modulated Signal (10dB SNR)'); |

TABLE 3.8. Receiver Amplifier Code

|  |
| --- |
| % Amplify the Signal  Amplifiedrx\_filteredsignal100 = rx\_filteredsignal100\*3;  Amplifiedrx\_filteredsignal50 = rx\_filteredsignal50\*3;  Amplifiedrx\_filteredsignal10 = rx\_filteredsignal10\*3; |

TABLE 3.9. QAM Demodulator Code

|  |
| --- |
| % Demodulate the Signal  Demodulatedsnr100 = qamdemod(Amplifiedrx\_filteredsignal100,M);  Demodulatedsnr50 = qamdemod(Amplifiedrx\_filteredsignal50,M);  Demodulatedsnr10 = qamdemod(Amplifiedrx\_filteredsignal10,M); |

TABLE 3.10. Bit Error Code

|  |
| --- |
| % Compute the Bit Error Code  [S0, R0] = biterr(Demodulatedsnr100,yy)  [S1, R1] = biterr(Demodulatedsnr50, yy)  [S2, R2] = biterr(Demodulatedsnr10, yy) |

TABLE 3.11. Recover the Received Signal

|  |
| --- |
| Recoveredsnr100 = udecode(uint32(Demodulatedsnr100),N)  Recoveredsnr50 = udecode(uint32(Demodulatedsnr50),N);  Recoveredsnr10 = udecode(uint32(Demodulatedsnr10),N);  % SNR = 100  subplot(211);  plot(t,Recoveredsnr100); %Time Domain  title('Recovered Signal (100dB SNR) in Time Domain');  xlabel('Time (s)');ylabel('Amplitude');  onehsnrFFT = fftshift(fft(Recoveredsnr50));  subplot(212);  plot(f,abs(onehsnrFFT)); %Frequency Domain  xlabel ('Frequency (Hz)');ylabel('Amplitude'); xlim([0 max(f)]);  title('Recovered Signal (100dB SNR) in Frequency Domain');  receivedsignalfile = 'C:\Users\Xavier\Desktop\LBYEC4A\snr100.wav';  audiowrite(receivedsignalfile,Recoveredsnr100,fs);  % SNR = 50  subplot(211);  plot(t,Recoveredsnr50); %Time Domain  title('Recovered Signal (50dB SNR) in Time Domain');  xlabel('Time (s)');ylabel('Amplitude');  fiftysnrFFT = fftshift(fft(Recoveredsnr50));  subplot(212);  plot(f,abs(fiftysnrFFT)); %Frequency Domain  xlabel ('Frequency (Hz)');ylabel('Amplitude'); xlim([0 max(f)]);  title('Recovered Signal (50dB SNR) in Frequency Domain');  receivedsignalfile = 'C:\Users\Xavier\Desktop\LBYEC4A\snr50.wav';  audiowrite(receivedsignalfile,Recoveredsnr50,fs);  % SNR = 10  subplot(211);  plot(t,Recoveredsnr10); %Time Domain  xlabel('Time (s)');ylabel('Amplitude');  title('Recovered Signal (10dB SNR) in Time Domain');  tensnrFFT = fftshift(fft(Recoveredsnr10));  subplot(212);  plot(f,abs(tensnrFFT)); %Frequency Domain  xlabel ('Frequency (Hz)'); ylabel('Amplitude'); xlim([0 max(f)]);  title('Recovered Signal (10dB SNR) in Frequency Domainn');  receivedsignalfile = 'C:\Users\Xavier\Desktop\LBYEC4A\snr10.wav';  audiowrite(receivedsignalfile,Recoveredsnr10,fs); |

## RESULTS

Figure 4.1 shows the Input Signal in its Time and Frequency Domains. In the time domain, the audio file is 3 seconds long. This shows that the audio recording block works because its duration parameter is set to 3 seconds. In the frequency domain, most of the digital data are in the frequencies of 100 to 500 Hz. This falls to the typical frequency range of human speech [13]. The peak amplitude is observed to be around 100 dB.

Chart, box and whisker chart

Description automatically generated

FIGURE 4.1. Time and Frequency plots of the Input Signal

Figure 4.2 to 4.4 show the scatterplots of the modulated signals for 10, 50, and 100 dB respectively. There are differences from the 10 dB plot to the other two. 50 dB and 100 dB are similar. This is expected since the higher the SNR, the more ideal the result is. The 10 dB result has more overlapping and clustering of points which indicates Inter Symbol Interference (ISI) [14]. This means there are some errors in the received signal. The three scatterplots have clustered or spread points which show a presence of noise. This is concluded to be coming from AWGN.

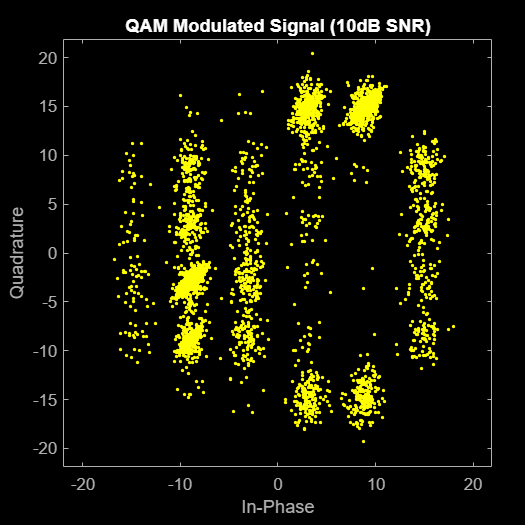


FIGURE 4.2. Scatter plot of the 10 dB Modulated Signal at SNR = 10 dB

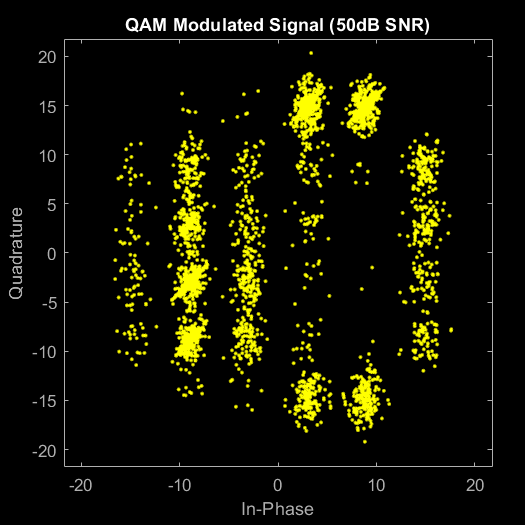


FIGURE 4.3. Scatter plot of the Modulated Signal at SNR = 50 dB

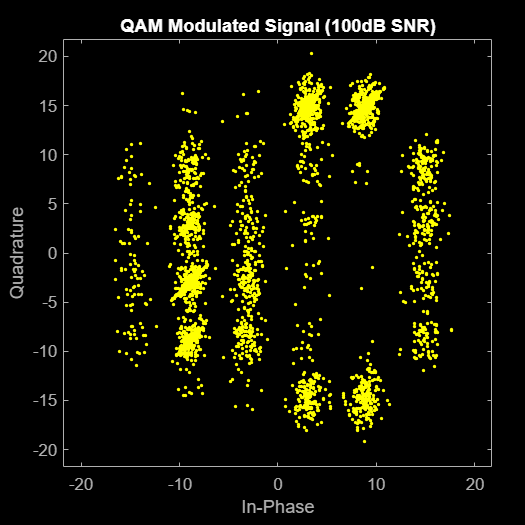


FIGURE 4.4. Scatter plot of the Modulated Signal at SNR = 100 dB

Figure 4.5 to 4.7 show the Time and Frequency Domain plots of the recovered signal respective to the SNR. Compared to Figure 4.1, there is no significant change in the frequency spectrum. It means the filter parameter is fitting to the input signal. There are also signs of AWGN as well. However, the evidence of noise is lesser as the SNR is higher. This result is to be expected as the SNR is the ratio between signal quality and background noise.

Chart, box and whisker chart

Description automatically generated

FIGURE 4.5. Time and Frequency plots of the Recovered Signal at SNR = 10 dB

Chart, box and whisker chart

Description automatically generated

FIGURE 4.6. Time and Frequency plots of the Recovered Signal at SNR = 50 dB

Chart, box and whisker chart

Description automatically generated

FIGURE 4.7. Time and Frequency plots of the Recovered Signal at SNR = 100 dB

Table 4.1 show the BER values for different SNR. The 10 dB has the highest BER of 0.52%. It means out of 2500 received bits, there are 13 bits that are wrong or destroyed. The 50- and 100-dB values have more accuracy of 0.48%, which means out of 625 bits, there are 3 bits that are wrong. As expected, the SNR and BER values produce an inverse linear relationship.

TABLE 4.1. Bit Error Rate (BER) for different SNR Values

|  |  |  |
| --- | --- | --- |
| SNR | Number of Bits | Ratio |
| 10 | 146 | 0.0052 |
| 50 | 136 | 0.0048 |
| 100 | 137 | 0.0048 |

.

## DISCUSSION

The main objective of the project is to create an efficient QAM Tx-Rx System using MATLAB. The results showed that the system was working successfully and there were no major problems that need to be addressed. The audio files show that the input data was not clipped and there were no other signs of data distortion. The BER for the different SNR Values was within the expected range and considered to be good. It also has an inverse relationship with the SNR.

The Chebyshev Type 1 filter was working successfully by minimizing the bandwidth in the transmitter and minimizing the AWGN in the receiver. This was done while having an order of 7. It means that the filter is efficient while having less complexity and cheaper components.

1. CONCLUSION

In conclusion, the proponent has successfully implemented a QAM Tx-Rx System using MATLAB. The output audio files show an audible quality without any major signs of clipping and distortion. The system also the range of a good BER and the results show inverse relationship between the SNR and BER.

The project is only limited to creating a QAM Tx-Rx system using MATLAB. It means further studies can be comparative projects between other Digital Modulation Tx-Rx systems. The project is also limited to Chebyshev Type 1 filter, as it is preferred because QAM needs a sharper roll-off in the stop band. This reason behind can be further investigated by using other types of filters.

# AUTHORS CONTRIBUTION

Guimalan, Francis Xavier – Project Code, Introduction, Theoretical Consideration, Methodology, Result, Discussion and Conclusion

Pozas, Raphael Dominic – Theoretical Consideration

Sabino, Cleajane – Introduction, Theoretical Consideration, Discussion, Poster, and Powerpoint Presentation

# REFERENCES

1. Agarwal, T. (2020, January 13). Digital Modulation : Types and Differences between Analog and Digital. ElProCus - Electronic Projects for Engineering Students. https://www.elprocus.com/digital-modulation-different-types-and-their-differences/
2. Agarwal, T. (2019, September 11). Amplitude Shift Keying : Circuit Diagram, Working and Its Applications. ElProCus - Electronic Projects for Engineering Students. https://www.elprocus.com/amplitude-shift-keying-ask-working-and-applications/
3. Agarwal, T. (2019b, September 11). Frequency Shift Keying (FSK) :Working, Advantages and Disadvantages. ElProCus - Electronic Projects for Engineering Students. https://www.elprocus.com/frequency-shift-keying-fsk-working-applications/.
4. Y, R. (2020, December 10). What is Phase Shift Keying (PSK)? BPSK modulation, BPSK demodulation, advantages, disadvantages and applications of PSK - Electronics Coach. Electronics Coach. https://electronicscoach.com/phase-shift-keying.html
5. Zola, A. (2021, August 16). QAM (quadrature amplitude modulation). Networking. https://www.techtarget.com/searchnetworking/definition/QAM
6. What is a good bit error rate? – Sage-Answer. (2020, November 13). https://sage-answer.com/what-is-a-good-bit-error-rate/
7. Altunian, G. (2021, August 12). What Is Signal-to-Noise Ratio and Why Does It Matter? Lifewire. https://www.lifewire.com/signal-to-noise-ratio-3134701
8. MathWorks. (n.d.). Add white Gaussian noise to signal - MATLAB awgn. https://www.mathworks.com/help/comm/ref/awgn.html
9. Xia, Z. (2022, April 2). What Is QAM? How Does QAM Work? - Huawei. Huawei. https://info.support.huawei.com/info-finder/encyclopedia/en/QAM.html
10. Y, R. (2018b, December 12). What is a Bandpass Filter? Definiton, design, response curve and applications of Bandpass Filter - Electronics Coach. Electronics Coach. <https://electronicscoach.com/bandpass-filter.html>
11. Agarwal, T. (2015, June 16). Chebyshev Filter - Diffferent Types of Chebyshev Filters. ElProCus - Electronic Projects for Engineering Students. https://www.elprocus.com/types-of-chebyshev-filters/
12. MathWorks. (n.d.-b). Fast Fourier Transform (FFT). MATLAB & Simulink. <https://www.mathworks.com/discovery/fft.html>
13. proAV / data and information, lists, tables and links. (n.d.). https://av-info.eu/index.html?https&&&av-info.eu/audio/speech-level.html
14. MathWorks. (n.d.-c). Quadrature amplitude modulation (QAM) - MATLAB qammod. <https://www.mathworks.com/help/comm/ref/qammod.html>