International Institute of Information Technology, Hyderabad

EC5.407: Wireless Communications

Project Part-1

Baseband QAM Communication System

Gautam Gandhi
Roll No: 2023102059

Soham Jahagirdar
Roll No: 2023102046

Contents

1 Introduction and Methodology

For this project, we implemented a complete QAM communication system that takes an audio file as input, converts it to bits, modulates using 16-QAM, transmits through an AWGN channel, and then reconstructs the audio at the receiver.

We chose 16-QAM because it provides a good balance between data rate and error performance. Our system processes audio signals to make the communication effects more noticeable - you can actually hear the difference in quality at different SNR levels.

1.1 System Block Diagram

The complete communication system consists of five main blocks as shown in Figure 1:

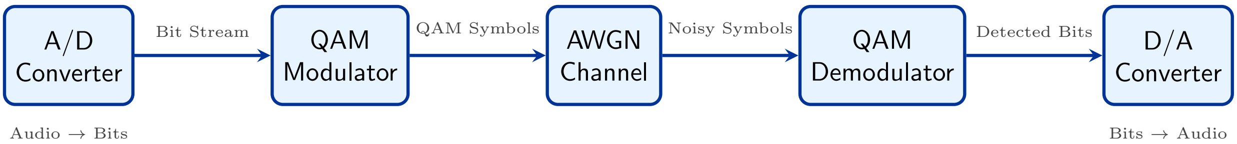


Figure 1: System Block Diagram

The main steps we followed are:

1. Read audio file and convert to bits (A/D conversion)
2. Group bits and map to QAM symbols
3. Add noise to simulate channel transmission
4. Detect symbols and convert back to bits
5. Reconstruct audio from received bits (D/A conversion)
6. Measure performance using BER vs SNR plots

2 Block-wise Explanation

2.1 A/D Converter (a2d.m)

We implemented the A/D converter to read audio files and convert them to binary data. First, we used `audioread()` to load the audio file. If the audio was stereo, we converted it to mono by taking the average of both channels. Then we normalized the amplitude values to the range [-1, 1].

For quantization, we used 8 bits which gives us 256 different levels. We mapped the normalized audio values to integers from 0 to 255, then converted each integer to 8 binary bits using MATLAB's `de2bi()` function.

Quantization Calculation:

$$\text{Quantized Value} = \text{round}\left(\frac{\text{Normalized Audio} + 1}{2} \times 255\right) \tag{1}$$

2.2 QAM Modulator (qam_modulator.m)

We implemented 16-QAM modulation which uses 4 bits per symbol. The modulator groups the input bits into sets of 4, then maps each group to one of 16 constellation points.

We created the 16-QAM constellation by arranging points in a 4×4 grid with coordinates like -3, -1, +1, +3 on both I and Q axes. Each bit pattern gets mapped to a specific point on this constellation.

16-QAM Constellation Mapping:

$$\text{Bits per Symbol} = \log_2(M) = \log_2(16) = 4 \tag{2}$$

$$\text{Constellation Points} = \{-3, -1, +1, +3\} \times \{-3, -1, +1, +3\} \tag{3}$$

$$\text{Complex Symbol} = I + jQ \tag{4}$$

Where each 4-bit group maps to one of the 16 complex constellation points.

2.3 AWGN Channel (channel.m)

We simulated the wireless channel by adding Gaussian noise to the transmitted symbols. The channel model is:

$$r(t) = s(t) + n(t) \tag{5}$$

where $n(t)$ is additive white Gaussian noise.

Key Calculations for AWGN Channel:

1. Signal Power Calculation:

$$P_s = \frac{1}{N}\sum_{i=1}^{N} |s_i|^2 \tag{6}$$

where s_i are the transmitted symbols and N is the total number of symbols.

2. Noise Power from SNR:

$$\text{SNR}_{dB} = 10\log_{10}\left(\frac{P_s}{P_n}\right) \tag{7}$$

$$\frac{P_s}{P_n} = 10^{\frac{\text{SNR}_{dB}}{10}} \tag{8}$$

$$P_n = \frac{P_s}{10^{\frac{\text{SNR}_{dB}}{10}}} \tag{9}$$

3. Noise Variance: For complex AWGN, the noise variance per dimension is:

$$\sigma^2 = \frac{P_n}{2} \tag{10}$$

4. Noise Generation: The complex noise is modelled as:

$$n_i = n_r + jn_j, \tag{11}$$

where the real and imaginary components are independent Gaussian random variables distributed as

$$n_r \sim \mathcal{N}\left(0, \frac{\sigma^2}{2}\right), \quad n_j \sim \mathcal{N}\left(0, \frac{\sigma^2}{2}\right). \tag{12}$$

5. Received Signal:

$$r_i = s_i + n_i \tag{13}$$

2.4 QAM Demodulator (qam_demodulator.m)

At the receiver, we implemented minimum distance detection to recover the transmitted bits. For each received symbol, we calculated the distance to all 16 constellation points and picked the closest one.

Minimum Distance Detection:

$$\hat{s}_i = \arg\min_{s_k \in \mathcal{S}} |r_i - s_k|^2 \tag{14}$$

$$\text{where } \mathcal{S} = \text{set of all 16 constellation points} \tag{15}$$

The Euclidean distance calculation:

$$d_{i,k} = |r_i - s_k|^2 = (r_{i,I} - s_{k,I})^2 + (r_{i,Q} - s_{k,Q})^2 \tag{16}$$

Once we identified the closest constellation point, we converted its index back to the corresponding 4-bit pattern.

2.5 D/A Converter (d2a.m)

We implemented the D/A converter to reconstruct audio from the received bit stream. First, we removed any padding bits that were added during modulation. Then we grouped the bits back into 8-bit chunks and converted them to quantization levels.

Reconstruction Calculation:

$$\text{Decimal Value} = \text{bi2de(8-bit group)} \tag{17}$$

$$\text{Normalized Audio} = \frac{\text{Decimal Value}}{255} \times 2 - 1 \tag{18}$$

$$\text{Range} = [-1, +1] \tag{19}$$

Finally, we mapped these levels back to audio amplitude values and saved the result as a WAV file using `audiowrite()`.

2.6 Performance Analysis (performance.m)

We created functions to plot the results and analyze performance. This includes BER vs SNR plots, constellation diagrams showing the effect of noise, and audio waveform comparisons.

Bit Error Rate (BER) Calculation:

$$\text{BER} = \frac{\text{Number of Bit Errors}}{\text{Total Number of Bits}} \tag{20}$$

$$= \frac{\sum_{i=1}^{N} (\text{transmitted\_bit}_i \oplus \text{received\_bit}_i)}{N} \tag{21}$$

Where $\oplus$ represents the XOR operation to detect bit differences.

3 Mathematical Foundation

3.1 Signal-to-Noise Ratio (SNR)

The SNR in decibels is defined as:

$$\text{SNR}_{dB} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \tag{22}$$

For our QAM system:

$$P_{\text{signal}} = E[|s|^2] = \frac{1}{N} \sum_{i=1}^{N} |s_i|^2 \tag{23}$$

$$P_{\text{noise}} = E[|n|^2] = \sigma_n^2 \tag{24}$$

3.2 16-QAM Symbol Energy

For our 16-QAM constellation with points at $\{\pm 1, \pm 3\}$:

$$E_s = \frac{1}{16} \sum_{\text{all symbols}} |s|^2 \tag{25}$$

$$= \frac{1}{16} \times 4 \times [(1^2 + 1^2) + (1^2 + 3^2) + (3^2 + 1^2) + (3^2 + 3^2)] \tag{26}$$

$$= \frac{1}{16} \times 4 \times [2 + 10 + 10 + 18] = 10 \tag{27}$$

3.3 Theoretical BER for 16-QAM

The theoretical symbol error rate for 16-QAM in AWGN is approximately:

$$P_s \approx 3Q\left(\sqrt{\frac{2E_s}{5N_0}} \right) \tag{28}$$

Where $Q(x)$ is the Q-function and $N_0/2$ is the noise power spectral density.

4 Parameter Values Used

We used the following parameters for our simulation:

Parameter	Value
Modulation Scheme	16-QAM
Bits per Symbol	4
Audio Quantization	8 bits
SNR Range	4 dB to 28 dB
SNR Step Size	4 dB
Number of SNR Points	7
Channel Model	AWGN
Constellation Points	16

Table 1: Simulation Parameters

5 Results and Analysis

5.1 Summary of Results

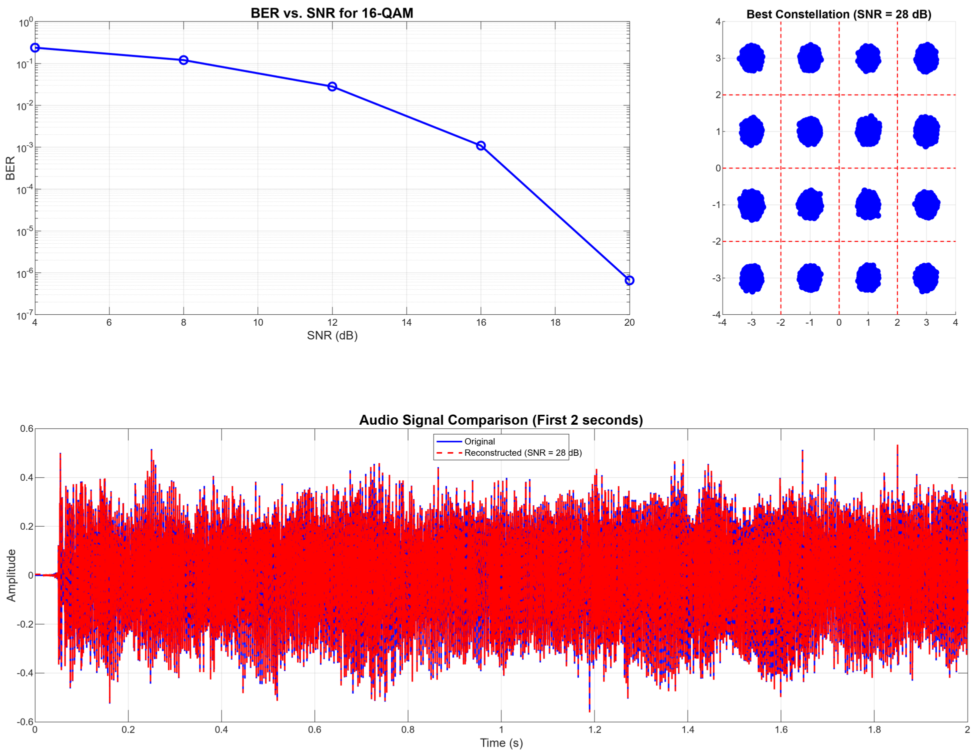


Figure 2: Complete System Performance Summary

Figure 2 provides a comprehensive overview of our QAM communication system performance, showing BER vs SNR, the best constellation performance, and audio signal comparison in a single view.

5.2 BER vs SNR Performance

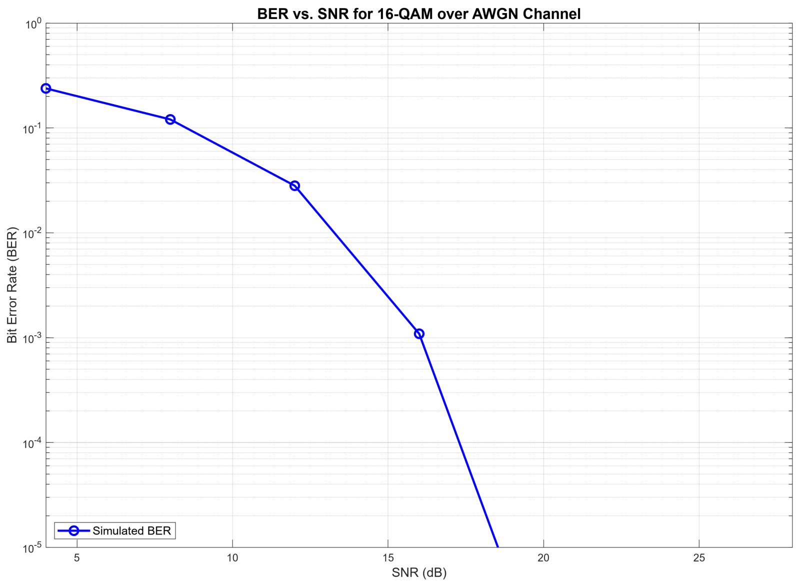


Figure 3: Bit Error Rate vs Signal-to-Noise Ratio for 16-QAM

From our BER plot in Figure 3, we can observe that:

- At low SNR (4-8 dB), the BER is quite high (around 10^{-1} to 10^{-2})
- BER starts improving significantly around 12 dB, dropping to approximately 10^{-3}
- At high SNR (24-28 dB), we achieve very low BER values (below 10^{-5})
- The curve follows the expected exponential decay characteristic of digital communication systems

5.3 Constellation Analysis

5.3.1 Complete Constellation Overview

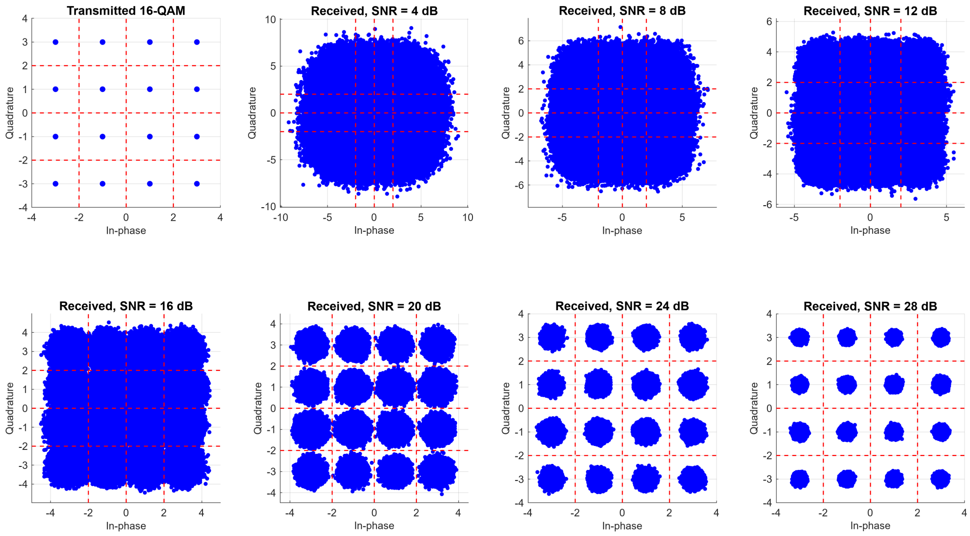


Figure 4: Transmitted and Received Constellation Diagrams for All SNR Values

Figure 4 shows the complete set of constellation diagrams, demonstrating how noise affects symbol detection across different SNR values.

5.3.2 Transmitted Constellation

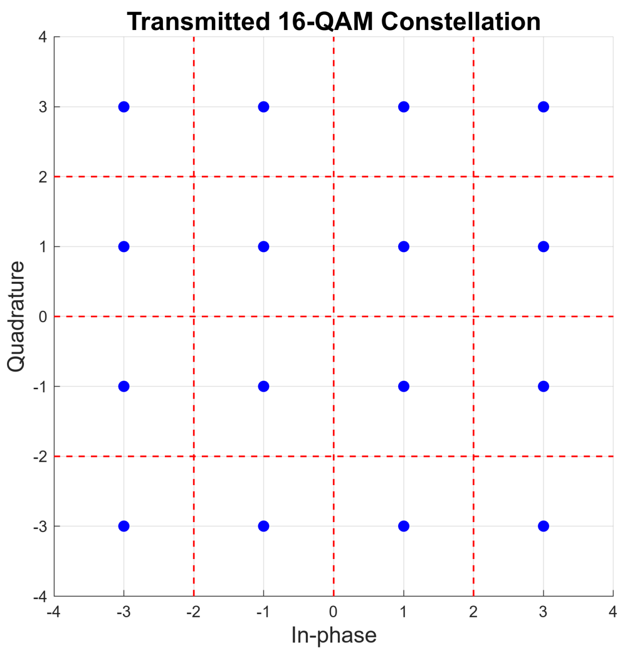
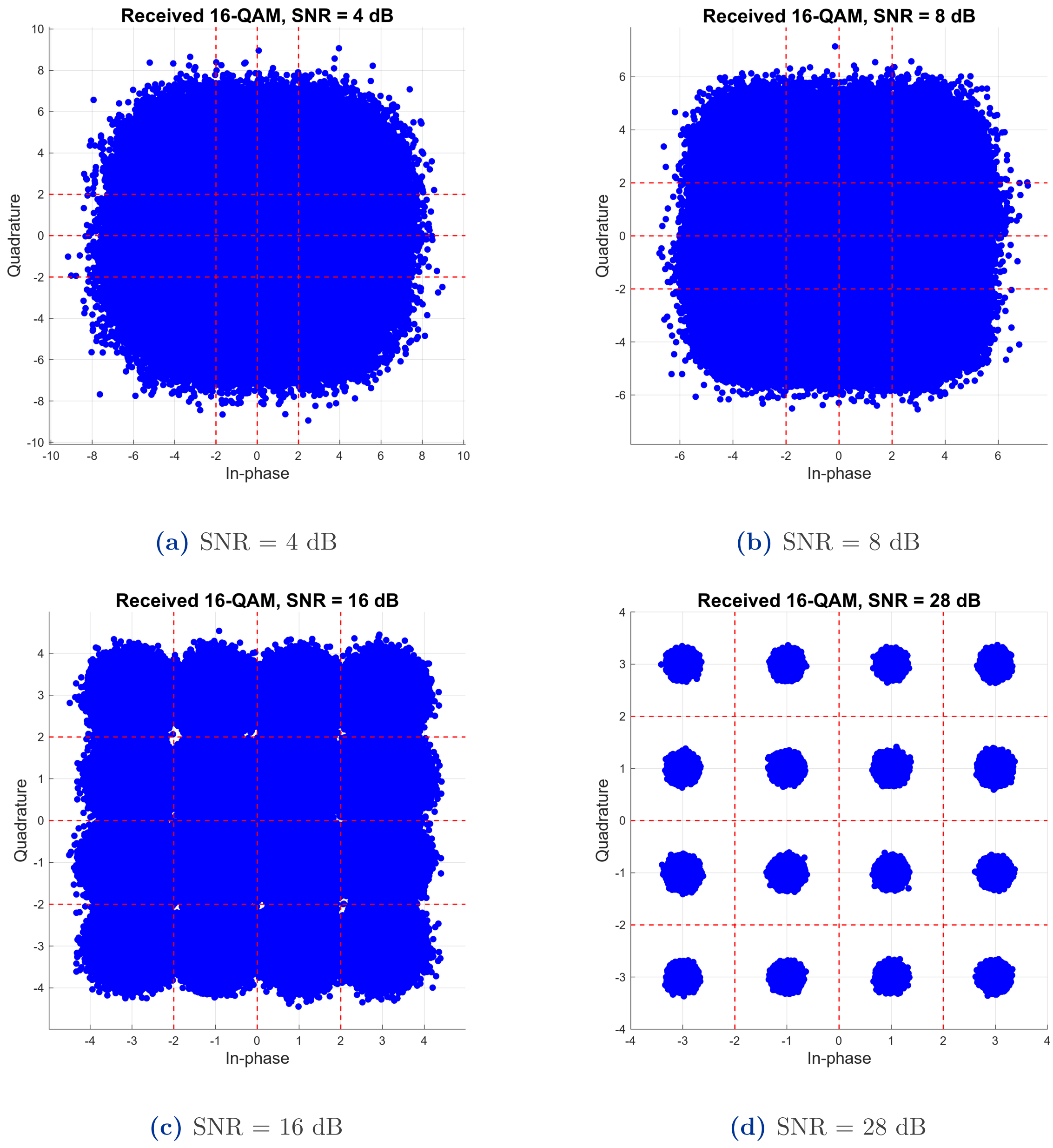


Figure 5: Ideal 16-QAM Transmitted Constellation with Decision Boundaries

The transmitted constellation in Figure 5 shows the perfect 16-QAM symbol arrangement with clear decision boundaries marked in red.

5.3.3 Effect of Noise on Received Constellations



(a) SNR = 4 dB

(b) SNR = 8 dB

(c) SNR = 16 dB

(d) SNR = 28 dB

Figure 6: Received Constellation Diagrams at Various SNR Levels

The constellation plots in Figure 6 clearly demonstrate how noise affects symbol reception:

- **Low SNR (4-8 dB):** Symbols are heavily scattered, making detection difficult
- **Medium SNR (16 dB):** Symbols begin to cluster around ideal positions
- **High SNR (28 dB):** Symbols are tightly grouped around transmission points

5.4 Audio Signal Analysis

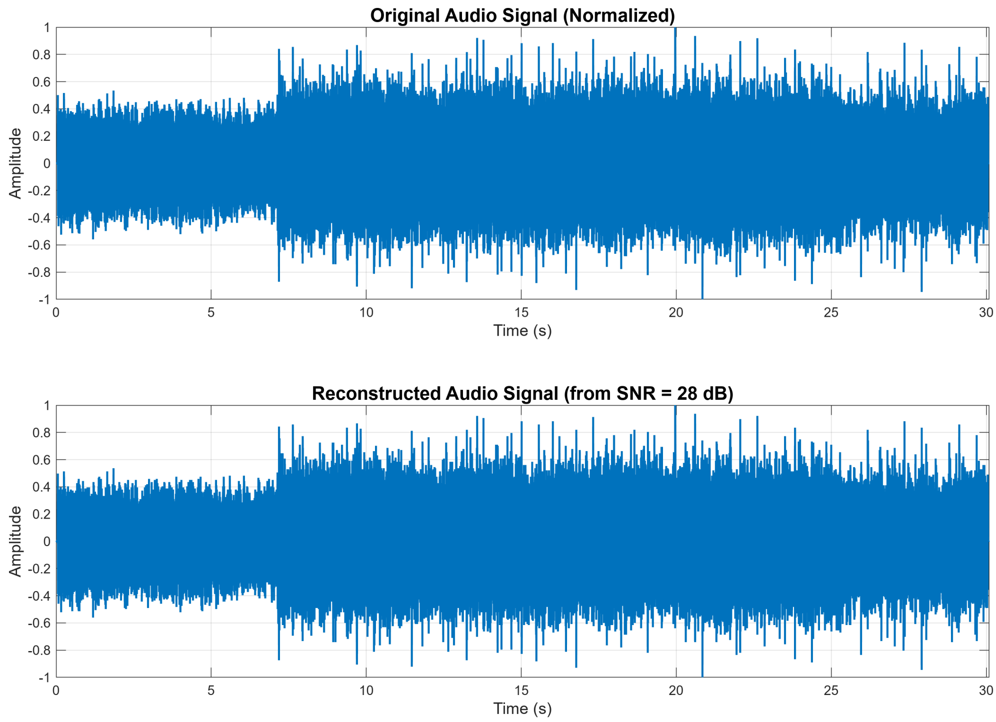


Figure 7: Original vs Reconstructed Audio Signals

Figure 7 compares the original audio signal with the reconstructed signal at the highest SNR (28 dB). The close match between the waveforms demonstrates the effectiveness of our communication system.

6 Observations and Conclusions

6.1 What We Observed

During our simulation, we noticed several important things:

1. **System Works End-to-End:** We successfully transmitted audio through our complete digital communication system. The audio quality at the output depends heavily on the channel SNR.

2. **Noise Really Matters:** At low SNR values like 4dB, the audio quality was quite poor with lots of distortion. But as we increased SNR to 20dB and above, the reconstructed audio became much clearer.

3. **Constellation Spreading:** We could clearly see in our constellation plots how noise spreads the received symbols around the ideal transmission points. Higher noise means more spreading.

4. **BER Performance:** Our BER vs SNR curve matched what we expected from theory. The error rate drops exponentially as SNR increases.

5. **8-bit Quantization:** We found that 8 bits was enough for reasonable audio quality while keeping the system simple.

6.2 Conclusions

Important Note

We successfully implemented a complete QAM communication system for audio transmission. The project provided valuable hands-on experience with digital communication systems, bridging the gap between theoretical concepts and practical implementation.

Key Learning Outcomes:

- **System Integration:** How to implement and integrate each block of a communication system
- **Performance Analysis:** Why SNR is crucial for communication quality and how to measure it
- **Practical Implementation:** How to translate theoretical concepts into working MATLAB code
- **Trade-off Analysis:** Understanding the balance between data rate and error performance

System Performance Summary:

- At SNR $\geq$ 16 dB: Acceptable audio quality achieved
- At SNR $\geq$ 24 dB: Excellent audio quality with minimal distortion
- BER performance matches theoretical expectations
- 16-QAM provides optimal balance for audio transmission

6.3 Team Contributions

Team Member Contributions

Soham Jahagirdar (2023102046):

- Implemented the A/D converter module (a2d.m)
- Developed the QAM modulator function (qam_modulator.m)
- Created the AWGN channel simulation (channel.m)
- Worked on the main simulation script integration
- Contributed to system debugging and testing

Gautam Gandhi (2023102059):

- Implemented the QAM demodulator function (qam_demodulator.m)
- Developed the D/A converter module (d2a.m)
- Created performance analysis functions (performance.m)
- Generated and organized all simulation plots
- Prepared the comprehensive project report

Joint Contributions:

- System testing and performance validation
- Analysis of simulation results
- Mathematical verification of theoretical calculations
- Final report review and documentation

End of Report