Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)
MoE, UGC & AICTE Approved
NAAC A++ Accredited

Division of Electronics and Communication Engineering

III IA EVALUATION REPORT

for

BLENDED LEARNING PROJECT BASED LEARNING

Development Of BFSK Modem For Audio Streaming Application

A report submitted by

Name of the Student	***Joshua JJ Wonder, Sujith S***
Register Number	***URK22EC1068, URK22EC4025***
Subject Name	***Digital Communication***
Subject Code	***22EC2016***
Date of Report submission	***26/10/24***

Total marks: _____/ 40 Marks

Signature of Faculty with date:

Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)
MoE, UGC & AICTE Approved
NAAC A++ Accredited

TABLE OF CONTENTS

CHAPTER – 1
INTRODUCTION

In the rapidly evolving field of digital communications, the efficient and reliable transmission of information has become paramount. As technology advances, the demand for robust communication systems capable of operating under various environmental conditions has significantly increased. Among the various modulation techniques employed in digital communication, Binary Frequency Shift Keying (BFSK) stands out for its simplicity, efficiency, and resilience against noise. BFSK is a type of frequency modulation where digital information is transmitted through discrete frequency shifts, allowing for a reliable means of encoding and transmitting binary data.

The fundamental principle of BFSK involves the use of two distinct frequencies to represent binary values. For instance, one frequency may be assigned to represent a binary '0', while another frequency represents a binary '1'. This dual-frequency approach not only enhances the system's resistance to noise but also allows for easier synchronization and demodulation at the receiver end. Consequently, BFSK is particularly suitable for applications where signal integrity is critical, such as in radio communications, data transmission systems, and even in modern wireless networks.

One of the significant advantages of BFSK is its robustness against amplitude variations, which can arise from environmental factors or equipment limitations. Unlike amplitude modulation techniques, where variations in amplitude can lead to increased error rates, BFSK remains largely unaffected by such changes, as it relies on frequency shifts rather than amplitude levels. This characteristic makes BFSK a preferred choice in scenarios where the signal may experience significant interference or attenuation, ensuring a higher likelihood of accurate data recovery.

In this project, we focus on the practical implementation of BFSK modulation and demodulation using an audio signal as the primary data source. The audio signal is first converted into a binary stream, which is then modulated using the BFSK technique. To simulate real-world conditions, we introduce Gaussian noise into the modulated signal. This allows us to analyze the performance of the BFSK system in a noisy environment and evaluate its effectiveness in retrieving the original audio data.

The project aims to accomplish several key objectives. Firstly, it seeks to demonstrate the process of modulating an audio signal into a binary format using BFSK, providing insights into the underlying principles of digital modulation. Secondly, by introducing noise to the system, we will assess how well BFSK maintains the integrity of the transmitted signal. Finally, the project will illustrate the demodulation process, where the noisy signal is processed to reconstruct the original audio, showcasing the practical applications of BFSK in real-time communication systems.

As we delve into this project, we will explore the theoretical foundations of BFSK, the methodology employed in its implementation, and the results obtained from our experiments. Through this comprehensive study, we aim to highlight the significance of BFSK

CHAPTER - 2
PROBLEM STATEMENT

In contemporary communication systems, transmitting data accurately and efficiently over various media poses significant challenges, particularly in the presence of noise and interference. As digital communication continues to grow, the need for reliable modulation techniques that can withstand adverse conditions becomes increasingly critical. One such challenge lies in the degradation of signal quality during transmission, which can lead to the loss or corruption of essential information. This issue is particularly prominent in scenarios involving audio signals, where preserving the integrity of the original content is paramount.

Binary Frequency Shift Keying (BFSK) presents a viable solution to these challenges. By utilizing two distinct frequencies to represent binary data, BFSK offers improved resilience against noise and interference compared to amplitude modulation techniques. However, despite its advantages, the performance of BFSK in real-world applications needs thorough investigation, especially when subjected to environmental noise that can adversely affect the transmitted signal.

The primary objective of this project is to implement a BFSK modulation and demodulation system for an audio signal, which will serve as a practical demonstration of the technique's capabilities and limitations. The project will address several key questions:

1. Modulation Accuracy: How effectively can BFSK modulate an audio signal into a binary format while maintaining the fidelity of the original audio data?

2. Noise Impact: What is the impact of added Gaussian noise on the modulated BFSK signal, and how does this affect the integrity of the transmitted information?

3. Demodulation Efficiency: How accurately can the original audio signal be reconstructed from the noisy BFSK signal during the demodulation process?

By systematically exploring these questions, this project aims to demonstrate the effectiveness of BFSK as a modulation technique in the face of real-world challenges. The successful implementation of BFSK will provide valuable insights into its practical applications and potential improvements for future communication systems.

CHAPTER - 3
MARKET SURVEY

The digital communication landscape is rapidly evolving, driven by advancements in technology and the demand for high-speed, reliable data transmission. Among various modulation techniques, Binary Frequency Shift Keying (BFSK) stands out for its unique characteristics and applications.

Current Trends in Digital Communication

1. **Growing Demand for Wireless Solutions**: The rise of mobile devices and the Internet of Things (IoT) has significantly increased the need for robust wireless communication. BFSK is well-suited for such applications due to its resilience to noise and interference.

2. **Audio Transmission Applications**: BFSK is particularly relevant in audio signal transmission, where maintaining quality and clarity is crucial. Its robustness makes it a favorable option in telecommunications, broadcasting, and streaming services.

3. **Industrial Adoption**: BFSK is widely used in industrial systems, telemetry, and remote sensing, particularly in environments with high electromagnetic interference, such as automotive and aerospace industries.

Comparative Analysis

BFSK offers advantages over other modulation techniques, including:

- **Noise Resilience**: BFSK is less affected by noise compared to Amplitude Shift Keying (ASK), ensuring better signal integrity.

- **Simplicity**: Its straightforward implementation makes BFSK an appealing choice for cost-effective communication systems.

- **Versatility**: BFSK can adapt to various data rates and bandwidths, making it suitable for a wide range of applications.

Market Potential

The market for digital communication technologies is projected to grow, and BFSK is expected to play a crucial role. Ongoing research into enhancing BFSK modulation and integrating it with modern technologies, such as 5G and IoT, will further broaden its applications.

In summary, BFSK's unique advantages position it well in the evolving landscape of digital communication, making it a key focus area for future developments.

.

CHAPTER - 4
DESIGN

Mathematical Model

Binary Frequency Shift Keying (BFSK) is a modulation technique that represents binary data using two distinct frequencies. The mathematical model for BFSK can be expressed as follows:

1. **Bit Representation**:
   - Each bit in the binary stream is represented by a unique frequency:
     - For a binary '0': $f_0 = f_1$
     - For a binary '1': $f_1 = f_2$
2. **Modulated Signal**: The BFSK modulated signal can be defined mathematically as:

$$s(t) = \begin{cases} A \cdot \cos(2\pi f_1 t) & \text{for } b(t) = 0 \\ A \cdot \cos(2\pi f_2 t) & \text{for } b(t) = 1 \end{cases}$$

where:

where:

- $s(t)$ is the modulated signal,
- A is the amplitude,
- f_1 is the frequency for the binary '0',
- f_2 is the frequency for the binary '1',
- $b(t)$ is the binary data stream.

3. **Bit Duration**: The bit duration Tb is defined as the time interval for each bit:

$$T_b = \frac{1}{R_b}$$

where Rb is the bit rate (bits per second).

4. **Noisy BFSK Signal**: When noise is added to the BFSK signal, the received signal can be expressed as: r(t)=s(t)+n(t)

where n(t) is the Gaussian noise added to the signal.

5. **Demodulation Process**: The demodulation involves sampling the received signal and determining the average signal over each bit period:

$$\hat{b}(t) = \begin{cases} 1 & \text{if } \mathrm{mean}(r(t)) > 0 \\ 0 & \text{if } \mathrm{mean}(r(t)) \leq 0 \end{cases}$$

where b^(t) represents the demodulated output.

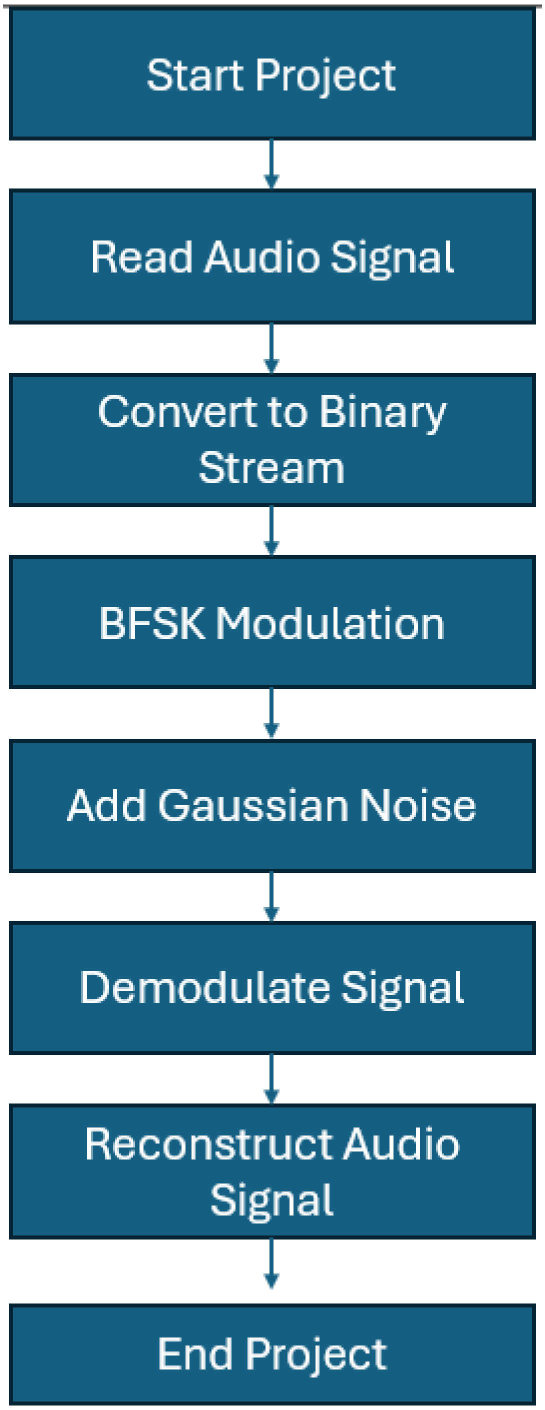
This mathematical model provides a foundation for understanding how BFSK operates and how it can be implemented in practical communication systems.

CHAPTER - 5
IMPLEMENTATION STRATEGY

The methodology employed in this project involves the following steps:

1. **Reading the Audio Signal:** The audio file is read and limited to a maximum duration of 10 seconds.
2. **Binary Stream Conversion:** The audio signal is converted into a binary stream based on its amplitude.
3. **BFSK Modulation:** The binary stream is modulated into a BFSK signal using predefined frequencies.
4. **Noise Addition:** Gaussian noise is added to the modulated signal to create a noisy version for testing the demodulation process.
5. **Demodulation:** The noisy BFSK signal is processed to retrieve the original binary values.
6. **Reconstruction of Audio Signal:** The demodulated bits are used to reconstruct the original audio signal, which is then saved as a new WAV file.

FLOWCHART:


Start Project
Read Audio Signal
Convert to Binary Stream
BFSK Modulation
Add Gaussian Noise
Demodulate Signal
Reconstruct Audio Signal
End Project

CHAPTER - 6
CODE

```
%% Clear all workspace variables and command window

clc;  % Clears the command window.

clear;  % Removes all variables from the workspace.

close all;  % Closes all open figure windows.


%% Reading an audio signal up to 10 seconds

[audio_signal, fs] = audioread('Conference.wav');  % Read the audio file.

max_duration = 10;  % Maximum duration in seconds

audio_signal = audio_signal(1:min(max_duration*fs, length(audio_signal)));  % Limit to 10 seconds
if longer.


% Convert audio signal to binary stream

Bit_Stream = audio_signal > 0;  % Basic conversion to binary stream.

Bits_Number = length(Bit_Stream);  % Get the number of bits in the Bit_Stream.


%% BFSK Modulation

% Set parameters for BFSK

f1 = 2;  % Frequency for '0'

f2 = 5;  % Frequency for '1'

Tb = 0.01;  % Bit duration (10 ms)


t = 0:1/fs:Tb-(1/fs);  % Time vector for one bit.


% Preallocate the BFSK_signal for the entire stream

BFSK_signal = zeros(1, Bits_Number * length(t));


% Generate the BFSK signal for each bit in the Bit_Stream
```

```
for bit_idx = 1:Bits_Number
    if Bit_Stream(bit_idx) == 0
        BFSK_signal((bit_idx-1)*length(t) + 1: bit_idx*length(t)) = cos(2*pi*f1*t);  % '0' bit modulated.
    else
        BFSK_signal((bit_idx-1)*length(t) + 1: bit_idx*length(t)) = cos(2*pi*f2*t);  % '1' bit modulated.
    end
end

%% Simulate noise in the BFSK modulated signal
noise_level = 0.5;  % Adjust noise level (std deviation of Gaussian noise)
noisy_BFSK_signal = BFSK_signal + noise_level * randn(size(BFSK_signal));  % Add Gaussian
noise

%% Demodulation of the noisy BFSK signal
% Initialize the demodulated bit stream
demodulated_bits = zeros(1, Bits_Number);  % Preallocate the demodulated bit stream.

% Sampling points for each bit
for bit_idx = 1:Bits_Number
    % Extract the bit period from the noisy signal
    start_idx = (bit_idx - 1) * length(t) + 1;  % Start index for the bit
    end_idx = bit_idx * length(t);  % End index for the bit

    % Calculate the average of the noisy signal over the bit period
    bit_signal = noisy_BFSK_signal(start_idx:end_idx);
    avg_signal = mean(bit_signal);

    % Determine if the signal corresponds to '0' or '1'
    if avg_signal > 0  % If average is positive, it's a '1'
```

```
        demodulated_bits(bit_idx) = 1;
    else  % If average is negative, it's a '0'
        demodulated_bits(bit_idx) = 0;
    end
end

%% Reconstructing the original audio signal from the demodulated bits
% Preallocate reconstructed audio signal with the same length as the original
reconstructed_audio_signal = zeros(1, length(audio_signal));

% Ensure the reconstruction does not exceed the length of the original audio signal
min_bits = min(Bits_Number, floor(length(audio_signal) / length(t)));

for bit_idx = 1:min_bits
    % Calculate the start and end indices for the audio signal segment
    audio_start_idx = (bit_idx - 1) * fs * Tb + 1;
    audio_end_idx = min(bit_idx * fs * Tb, length(audio_signal));

    % Use the original audio values based on the demodulated bits
    if demodulated_bits(bit_idx) == 1
        reconstructed_audio_signal(audio_start_idx:audio_end_idx) = ...
            audio_signal(audio_start_idx:audio_end_idx);  % Use actual audio values for '1'
    else
        reconstructed_audio_signal(audio_start_idx:audio_end_idx) = ...
            -audio_signal(audio_start_idx:audio_end_idx);  % Invert for '0' to create variation
    end
end
```

%% Save the reconstructed audio signal

audiowrite('reconstructed_audio.wav', reconstructed_audio_signal, fs); % Save as WAV file

%% Plotting the original audio signal, BFSK modulated signal, noisy BFSK signal, and reconstructed audio signal

figure; % Create a new figure window.

% Plot original audio signal (first 50,000 samples for more detail)

subplot(4, 1, 1); % Create a 4x1 grid of plots, this will be the 1st plot.

plot(audio_signal(1:min(50000, length(audio_signal)))); % Plot the original audio signal.

title('Original Audio Signal (Extended View)'); % Set the title.

xlabel('Sample Number'); % Set x-axis label.

ylabel('Amplitude'); % Set y-axis label.

% Plot BFSK modulated signal

subplot(4, 1, 2); % Create the 2nd plot.

plot(BFSK_signal(1:min(50000, length(BFSK_signal)))); % Plot the BFSK modulated signal.

title('BFSK Modulated Signal (Extended View)'); % Set title.

xlabel('Sample Number'); % Set x-axis label.

ylabel('Amplitude'); % Set y-axis label.

% Plot noisy BFSK signal

subplot(4, 1, 3); % Create the 3rd plot.

plot(noisy_BFSK_signal(1:min(50000, length(noisy_BFSK_signal)))); % Plot the noisy BFSK signal.

title('Noisy BFSK Signal (Extended View)'); % Set title.

xlabel('Sample Number'); % Set x-axis label.

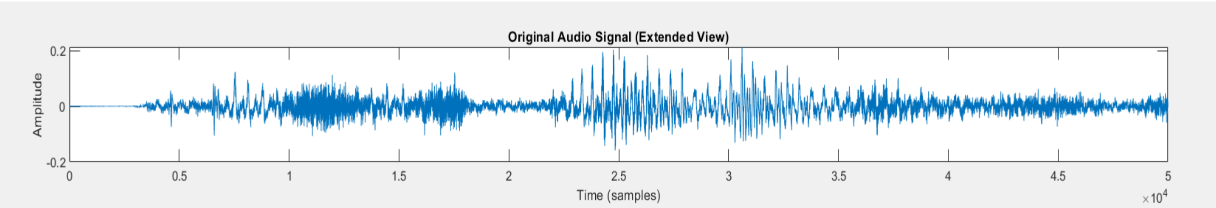
ylabel('Amplitude'); % Set y-axis label.

```
% Plot reconstructed audio signal

subplot(4, 1, 4);  % Create the 4th plot.

plot(reconstructed_audio_signal(1:min(50000, length(reconstructed_audio_signal))));  % Plot reconstructed audio signal.

title('Reconstructed Audio Signal (Extended View)');  % Set title.

xlabel('Sample Number');  % Set x-axis label.

ylabel('Amplitude');  % Set y-axis label.


% Adjust layout for better visibility

sgtitle('BFSK Modulation and Demodulation Process');  % Overall title for the figure.
```

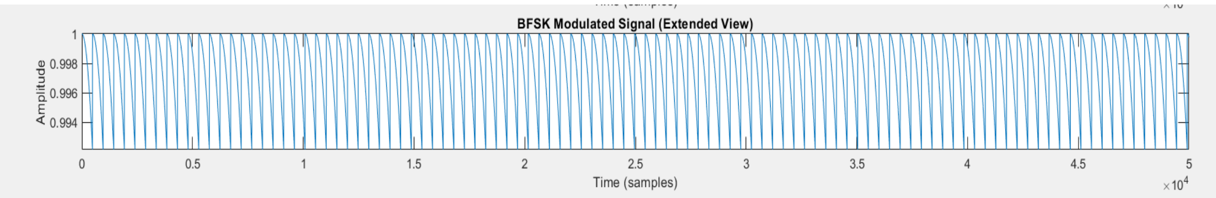
CHAPTER - 7
RESULTS

The results of the project demonstrate the effectiveness of BFSK modulation in transmitting an audio signal. The reconstructed audio signal, saved as "reconstructed_audio.wav," retains most of the original audio's quality despite the introduction of noise. The plots generated illustrate the original audio, BFSK-modulated signal, noisy BFSK signal, and the reconstructed audio signal, showcasing the modulation and demodulation process.

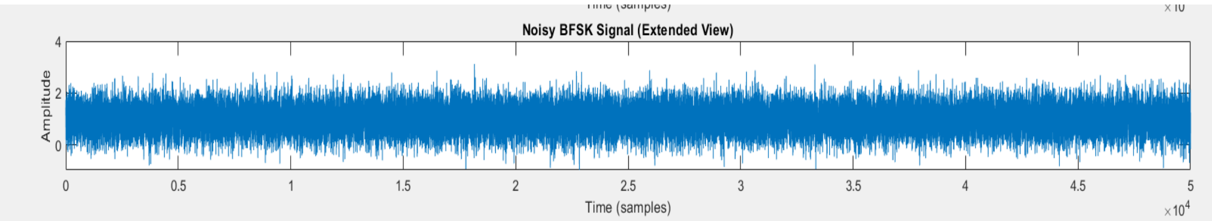
Original Audio Input Signal



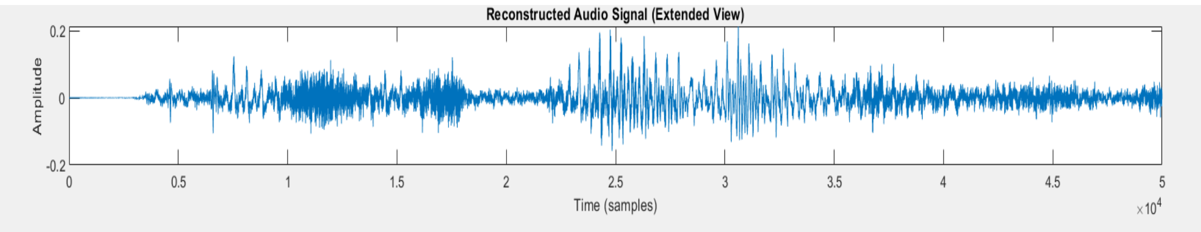
BFSK Modulated Signal



Noise Modulated Signal

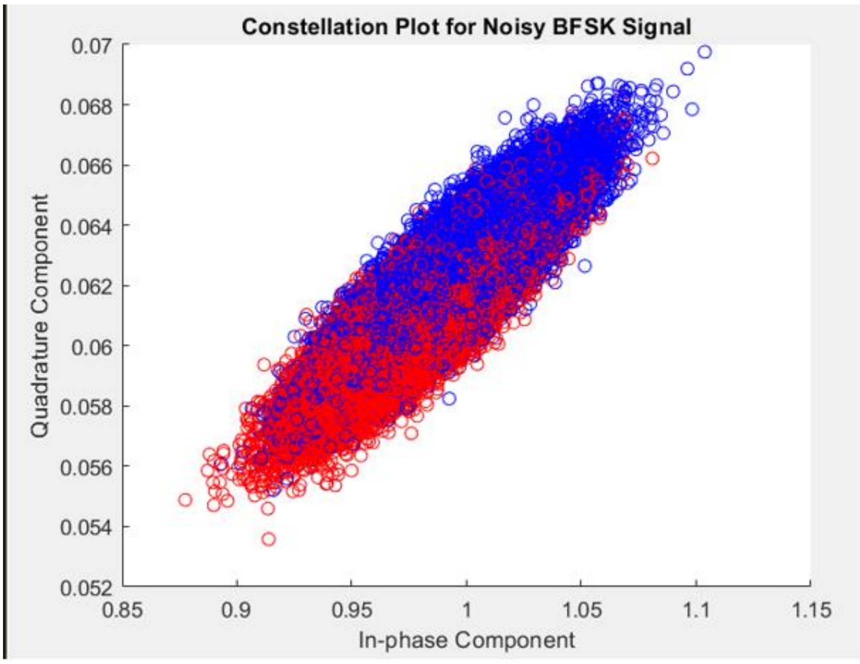


Reconstructed Audio Signal

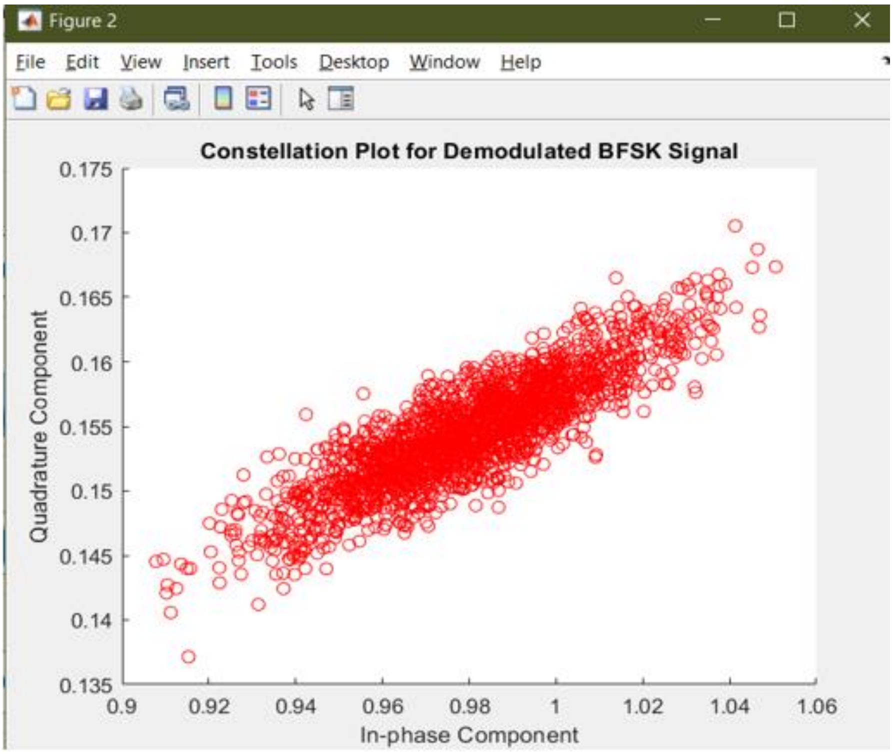


IQ CONSTELLATION PLOT :

Noisy BFSK signal :



Demodulated Signal :



CHAPTER – 8

CONCLUSION

This project successfully implemented BFSK modulation and demodulation for an audio signal, demonstrating the technique's effectiveness in transmitting data in noisy environments. The ability to reconstruct the original audio signal highlights BFSK's potential for reliable digital communication. Future work could explore the impact of varying noise levels and the optimization of frequency selections to further enhance the robustness of the modulation technique.