Ground-Based Telemetry Communication System

21ECC302T ANALOG AND DIGITAL COMMUNICATION

PROJECT REPORT

Submitted by

Lehan S Ajish [RA221104010579]
Abhinav Kumar [RA221104010578]
Snehak Tarun Tiu [RA221104010575]
Saksham Nayak [RA221104010593]

Semester – VI
Academic Year: 2024–25

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

College of Engineering and Technology ,
SRM Institute of Science and Technology
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamil Nadu.

ABSTRACT

This project simulates a ground-based telemetry communication system using Pulse Code Modulation (PCM) and Quadrature Phase Shift Keying (QPSK) modulation techniques. The system is designed to model digital data exchange between two terrestrial stations over a noisy communication link. By comparing a basic PCM-only system to one enhanced with QPSK modulation, we demonstrate how digital modulation techniques improve noise resistance and data integrity. An Additive White Gaussian Noise (AWGN) channel is used to mimic real-world transmission disturbances. Bit Error Rate (BER) and Mean Squared Error (MSE) are calculated to evaluate performance.

OBJECTIVES

- To simulate a digital telemetry system for ground-to-ground communication.
- To encode analog data using 8-bit PCM for binary transmission.
- To compare the performance of PCM-only and PCM+QPSK systems.
- To evaluate data integrity under noisy conditions using BER and MSE.

INTRODUCTION

Ground-based telemetry systems facilitate the remote transmission of sensor or control data between fixed terrestrial stations such as weather observatories, industrial plants, or centralized control hubs. These systems typically operate over RF or wireless links, which are vulnerable to interference, noise, and bandwidth constraints. Unlike satellite-based communication, terrestrial systems must address more localized and persistent signal degradation. To ensure reliable data transfer under such conditions, efficient modulation and encoding techniques are essential. This project explores methods to improve the resilience and accuracy of ground telemetry systems.

To achieve reliable communication, the project incorporates Pulse Code Modulation (PCM) for digitizing analog sensor data and Quadrature Phase Shift Keying (QPSK) for bandwidth-efficient modulation. PCM converts analog signals, such as temperature or pressure readings, into binary sequences suitable for digital processing and transmission. QPSK, a digital modulation scheme, enhances spectrum efficiency by transmitting two bits per symbol, thus reducing bandwidth requirements. This integration helps in overcoming noise and channel limitations. The system is evaluated in a simulated Additive White Gaussian Noise (AWGN) environment to mimic real-world channel conditions.

The study compares two models: a basic PCM-only system and an enhanced PCM+QPSK system, both implemented and analyzed using Scilab. By comparing Bit Error Rate (BER) and Mean Squared Error (MSE) across the two systems, we assess the improvement offered by modulation. The PCM+QPSK model demonstrates better performance under noisy conditions, validating the advantage of combining digital modulation with traditional encoding. Using Scilab allows for accurate simulation without hardware dependencies, making it ideal for educational and research-focused telemetry system development.

SYSTEM DESCRIPTION

1. BLOCK DIAGRAM FOR PCM:

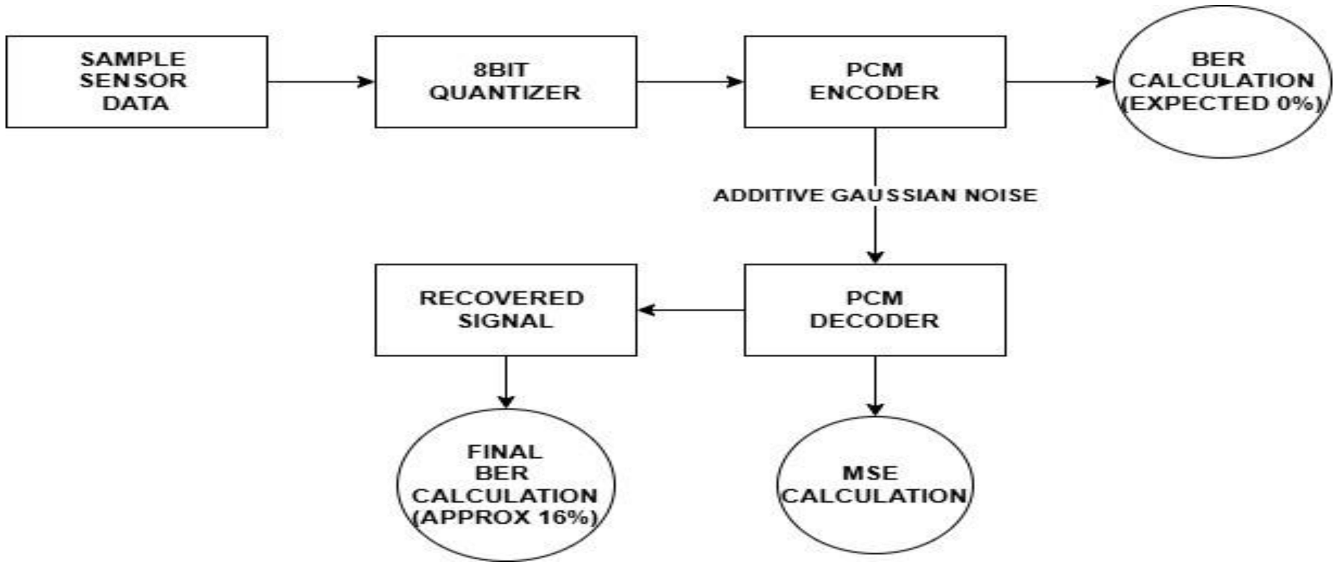

Fig 1: PCM Communication Block

DESCRIPTION:

The block diagram illustrates the Pulse Code Modulation (PCM) process used in the ground-based telemetry communication system. Initially, analog sensor data is sampled and quantized into 8-bit digital values. These quantized values are encoded using a PCM encoder, where the expected Bit Error Rate (BER) is ideally 0% in the absence of noise. To simulate real-world channel conditions, Additive Gaussian Noise is introduced before the signal is decoded at the receiver end using a PCM decoder. The decoded signal is compared to the original to compute the Mean Squared Error (MSE) and assess signal distortion. The recovered signal undergoes final BER calculation, which approximates a 16% error due to the added noise. This setup serves as the baseline PCM system in the larger project, helping to evaluate the benefits of incorporating QPSK modulation for improved noise resilience and data accuracy in telemetry communication.

2. BLOCK DIAGRAM FOR PCM + QPSK

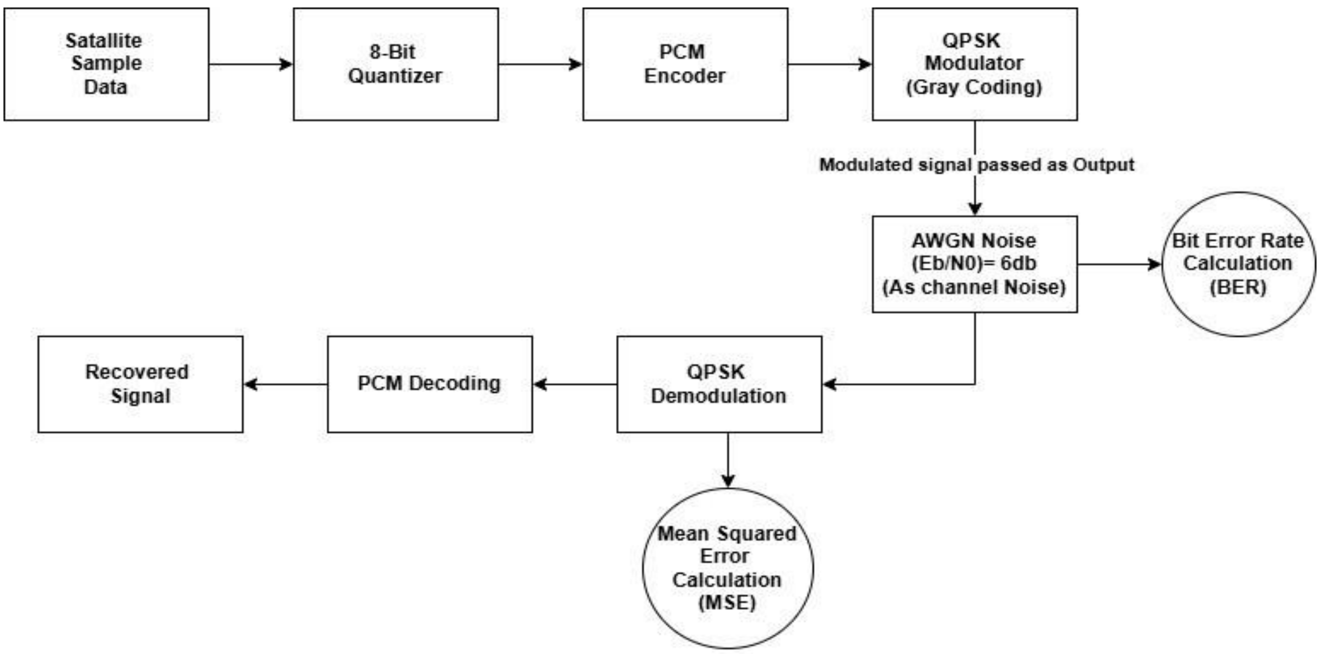

Fig 2: PCM + QPSK Communication Block

DESCRIPTION

The block diagram represents the enhanced ground-based telemetry communication system utilizing both Pulse Code Modulation (PCM) and Quadrature Phase Shift Keying (QPSK) modulation. The system begins with satellite sample data, which is quantized using an 8-bit quantizer and encoded using PCM. The encoded data is then modulated using QPSK with Gray coding to ensure minimal bit errors during transmission. To simulate realistic communication conditions, the modulated signal is passed through an AWGN (Additive White Gaussian Noise) channel with a signal-to-noise ratio of $E_b/N_0 = 6$ dB. At the receiver end, the signal undergoes QPSK demodulation followed by PCM decoding to retrieve the original signal. Bit Error Rate (BER) and Mean Squared Error (MSE) calculations are performed to assess the system's performance. This enhanced system demonstrates improved resilience to noise and better data integrity compared to the PCM-only setup, validating the effectiveness of digital modulation in telemetry applications.

METHODOLOGY

Implementation of PCM and PCM + QPSK Telemetry Systems

1. Pulse Code Modulation (PCM) System

The PCM system processes sensor data using quantization and binary encoding before transmission. It is then subjected to additive white Gaussian noise (AWGN) and decoded to measure signal distortion.

1. **Sensor Data Generation**
    - Generates 1000 random temperature samples in the range 0°C–100°C.
    - Plots original sensor data for visualization.
2. **PCM Encoding**
    - Quantizes the analog temperature values into 8-bit binary format.
    - Converts quantized values into a continuous binary bitstream.
3. **Transmission & Noise Addition**
    - Passes the binary stream through an AWGN channel.
    - Adds Gaussian noise to simulate real-world transmission distortions.
4. **PCM Decoding**
    - Rounds noisy bits to restore original values.
    - Converts binary values back into quantized temperature readings.
5. **Performance Evaluation**
    - Computes Bit Error Rate (BER) before and after transmission.
    - Calculates Mean Squared Error (MSE) between original and reconstructed signals.
    - Plots comparative results between original and recovered sensor data.

Scilab Code for PCM System

```
// Custom function for decimal-to-binary conversion

function bin_str=dec2bin_custom(val, bit_depth)

    bin_str = "";

    for i = bit_depth-1:-1:0

        bin_str = bin_str + string(floor(val / 2^i));

        val = modulo(val, 2^i);

    end

endfunction


// Sensor Data Generation

num_samples = 1000; bit_depth = 8; snr_db = 6;

sensor_data = 100 * rand(1, num_samples);


// PCM Encoding

quantized_data = round((sensor_data * (2^bit_depth - 1)) / 100);

binary_stream = [];

for i = 1:num_samples

    bin_str = dec2bin_custom(quantized_data(i), bit_depth);

    binary_stream = [binary_stream, ascii(bin_str) - ascii('0')];

end
```

```
// Add AWGN Noise

snr = 10^(snr_db / 10);

noise = rand(1, length(binary_stream), "normal") * sqrt(0.5/snr);

noisy_binary_stream = binary_stream + noise;


// PCM Decoding & Reconstruction

recovered_values = [];

for i = 1:bit_depth:length(noisy_binary_stream)

    bits = round(noisy_binary_stream(i:i+bit_depth-1));

    bit_str = strcat(string(bits));

    val = bin2dec(bit_str);

    temp = val * 100 / (2^bit_depth - 1);

    recovered_values = [recovered_values, temp];

end


// Performance Analysis

ber = sum(binary_stream <> round(noisy_binary_stream)) / length(binary_stream);

mse = mean((sensor_data - recovered_values).^2);

disp("Bit Error Rate (BER) = " + string(ber));

disp("Mean Squared Error (MSE) = " + string(mse));
```

2. PCM + Quadrature Phase Shift Keying (QPSK) System

The PCM+QPSK system enhances data transmission by applying QPSK modulation to the encoded bitstream before transmission, improving spectral efficiency and noise resilience.

1. **Sensor Data Generation**
   - Similar to PCM, generates 1000 temperature samples and plots them.
2. **PCM Encoding**
   - Quantizes data into 8-bit binary values.
   - Forms a binary stream for transmission.
3. **QPSK Modulation**
   - Groups bits into symbol pairs for QPSK mapping.
   - Maps symbols to complex signal space.
4. **Transmission & Noise Addition**
   - Passes modulated symbols through an AWGN channel.
   - Adds Gaussian noise to simulate real-world disturbances.
5. **QPSK Demodulation**
   - Recovers binary stream from noisy symbols.
   - Decodes symbols back into 8-bit PCM values.
6. **Performance Evaluation**
   - Computes **Bit Error Rate (BER)** before and after transmission.
   - Calculates **Mean Squared Error (MSE)** to measure reconstruction accuracy.
   - Plots comparative results.

Scilab Code for PCM + QPSK System

// Custom decimal-to-binary conversion function

```
function bin_str=dec2bin_custom(val, bit_depth)
    bin_str = "";
    for i = bit_depth-1:-1:0
        bin_str = bin_str + string(floor(val / 2^i));
        val = modulo(val, 2^i);
    end
endfunction
```

// QPSK Modulation Function

```
function symbols=qpsk_modulation(bitstream)
    symbols = [];
    for i = 1:2:length(bitstream)
        if bitstream(i) == 0 & bitstream(i+1) == 0 then symbols = [symbols, 1 + %i];
        elseif bitstream(i) == 0 & bitstream(i+1) == 1 then symbols = [symbols, -1 + %i];
        elseif bitstream(i) == 1 & bitstream(i+1) == 0 then symbols = [symbols, 1 - %i];
        else symbols = [symbols, -1 - %i];
        end
    end
endfunction
```

// Sensor Data Generation

```
num_samples = 1000; bit_depth = 8; snr_db = 6;
sensor_data = 100 * rand(1, num_samples);
```

// PCM Encoding

```
quantized_data = round((sensor_data * (2^bit_depth - 1)) / 100);

binary_stream = [];

for i = 1:num_samples

    bin_str = dec2bin_custom(quantized_data(i), bit_depth);

    binary_stream = [binary_stream, ascii(bin_str) - ascii('0')];

end
```

// QPSK Modulation

```
qpsk_symbols = qpsk_modulation(binary_stream);
```

// Add AWGN Noise

```
snr = 10^(snr_db / 10);

noise = sqrt(1 / (2 * snr)) * (rand(1, length(qpsk_symbols), "normal") + %i * rand(1, length(qpsk_symbols), "normal"));

noisy_qpsk_symbols = qpsk_symbols + noise;
```

// QPSK Demodulation & PCM Decoding

```
recovered_binary_stream = qpsk_demodulation(noisy_qpsk_symbols);

recovered_values = [];

for i = 1:bit_depth:length(recovered_binary_stream)

    bits = recovered_binary_stream(i:i+bit_depth-1);

    bit_str = strcat(string(bits));

    val = bin2dec(bit_str);

    temp = val * 100 / (2^bit_depth - 1);

    recovered_values = [recovered_values, temp];

end
```

```
// Performance Analysis

ber = sum(binary_stream <> recovered_binary_stream) / length(binary_stream);

mse = mean((sensor_data - recovered_values).^2);

disp("Bit Error Rate (BER) = " + string(ber));

disp("Mean Squared Error (MSE) = " + string(mse));
```

Comparison of PCM and PCM + QPSK Systems

System	Modulation	Noise Resistance	Bit Error Rate (BER)	Mean Squared Error (MSE)
PCM Only	None	Low	Higher	Greater distortion
PCM + QPSK	QPSK	Improved	Lower	Enhanced accuracy

Fig 3: Comparison of PCM and PCM+ QPSK Systems

RESULT (Pulse Code Modulation (PCM) System)

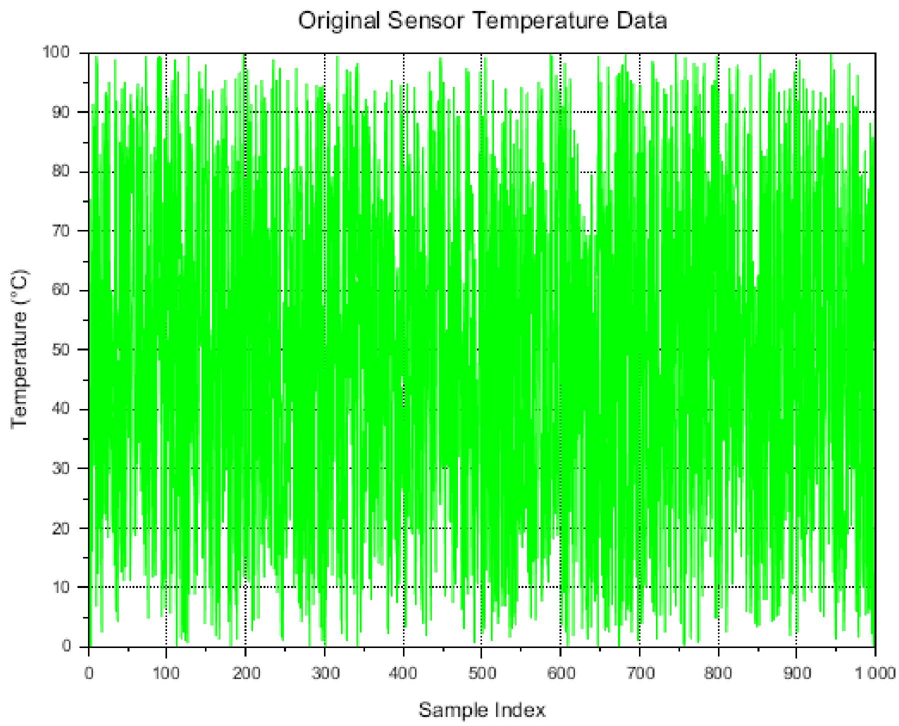

Fig 4: original sensor temperature data

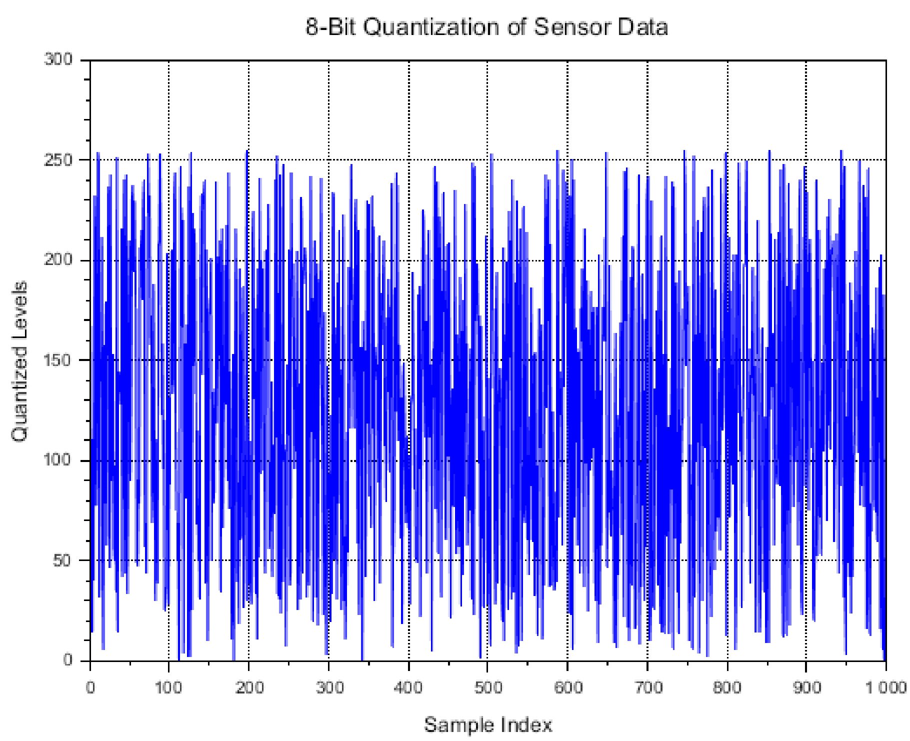

Fig 5: 8-Bit Quantization of Sensor Data

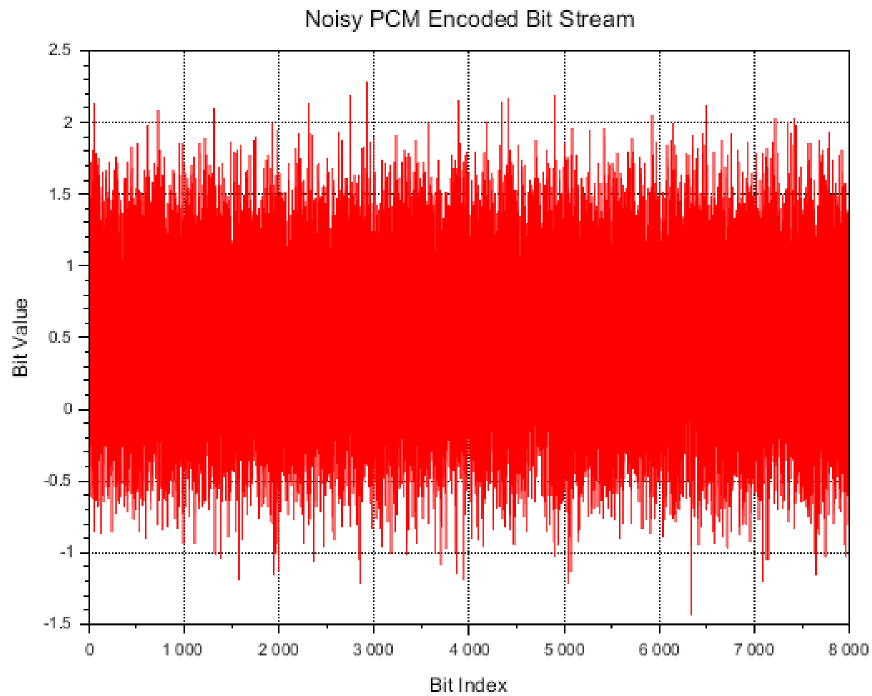

Fig 6: Noisy PCM Encoded Bit Stream

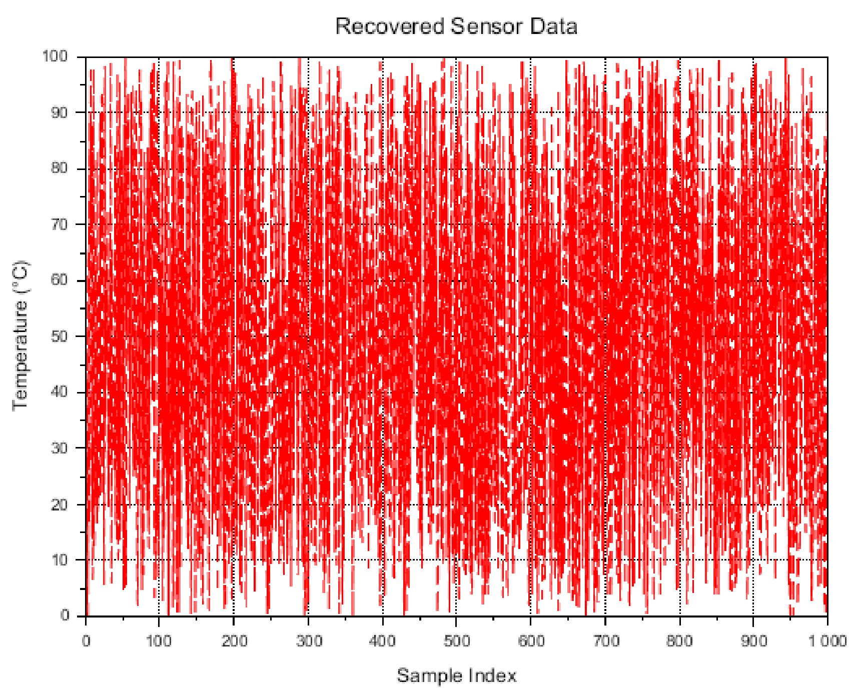

Fig 7: Recovered Sensor Data

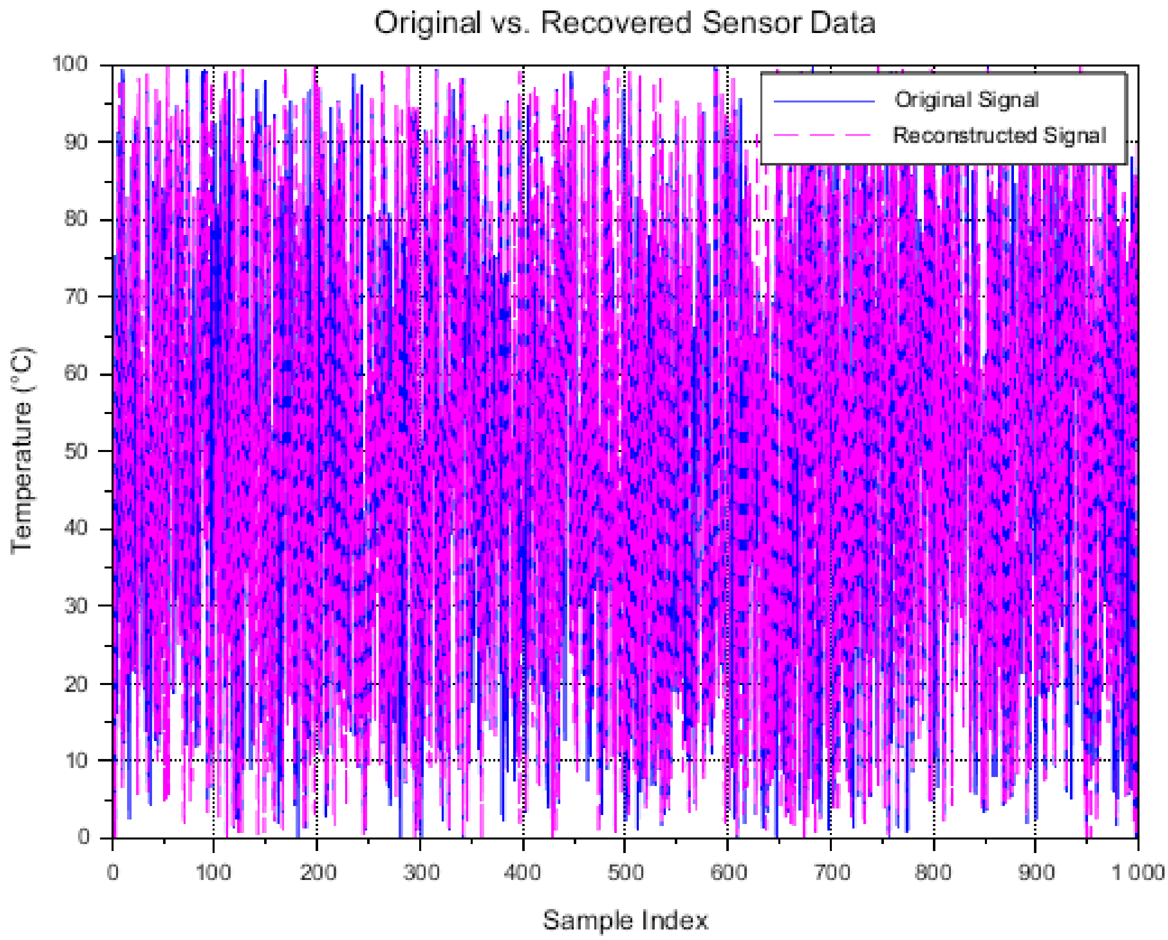

Fig 8: Original vs Recovered Sensor Data

RESULT (PCM + Quadrature Phase Shift Keying (QPSK) System)

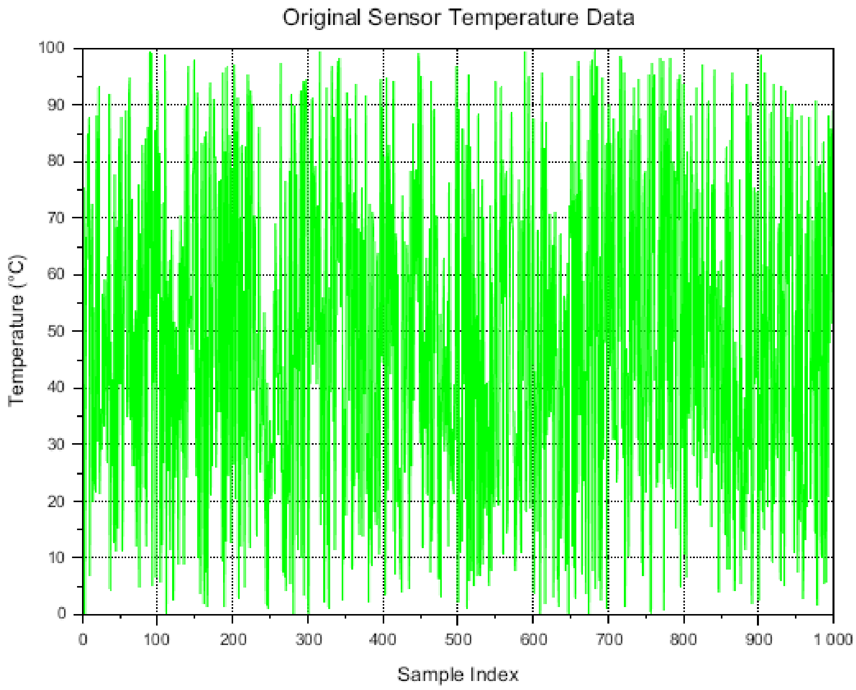

Fig 9: original sensor temperature data

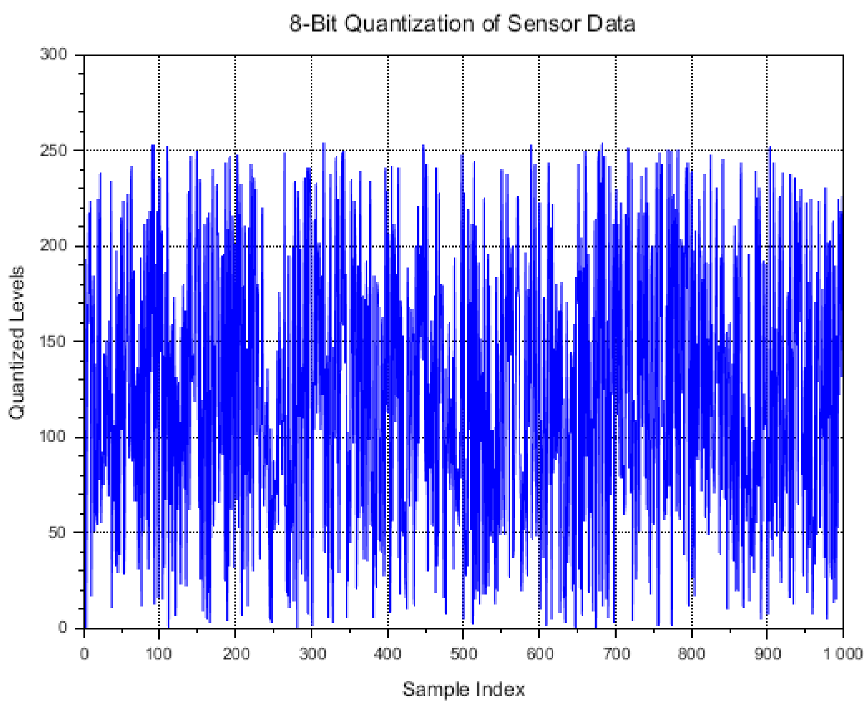

Fig 10: 8-Bit Quantization of Sensor Data

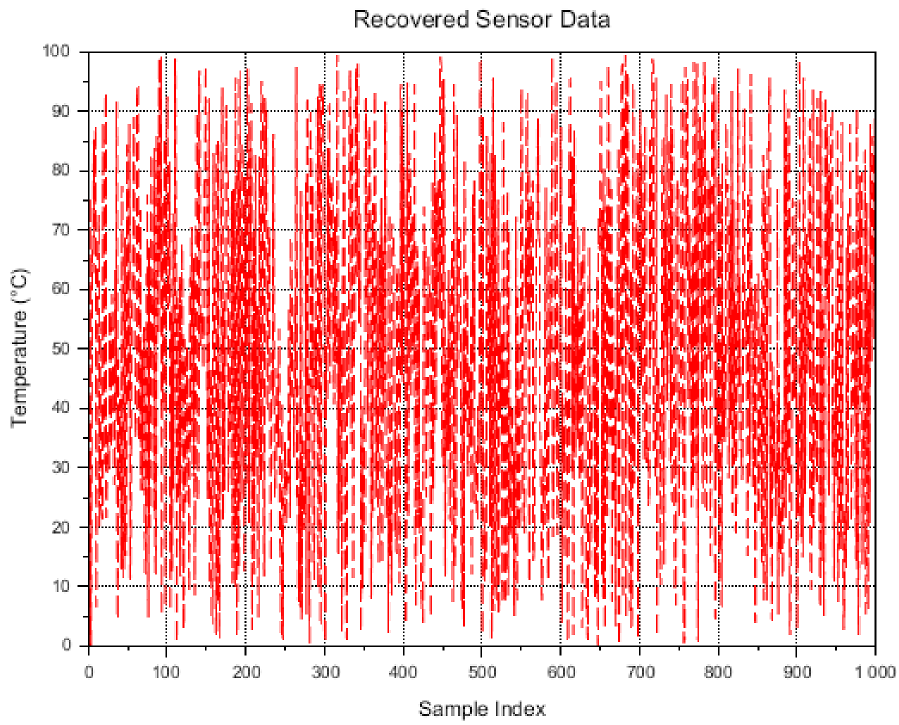

Fig 11: Recovered Sensor Data

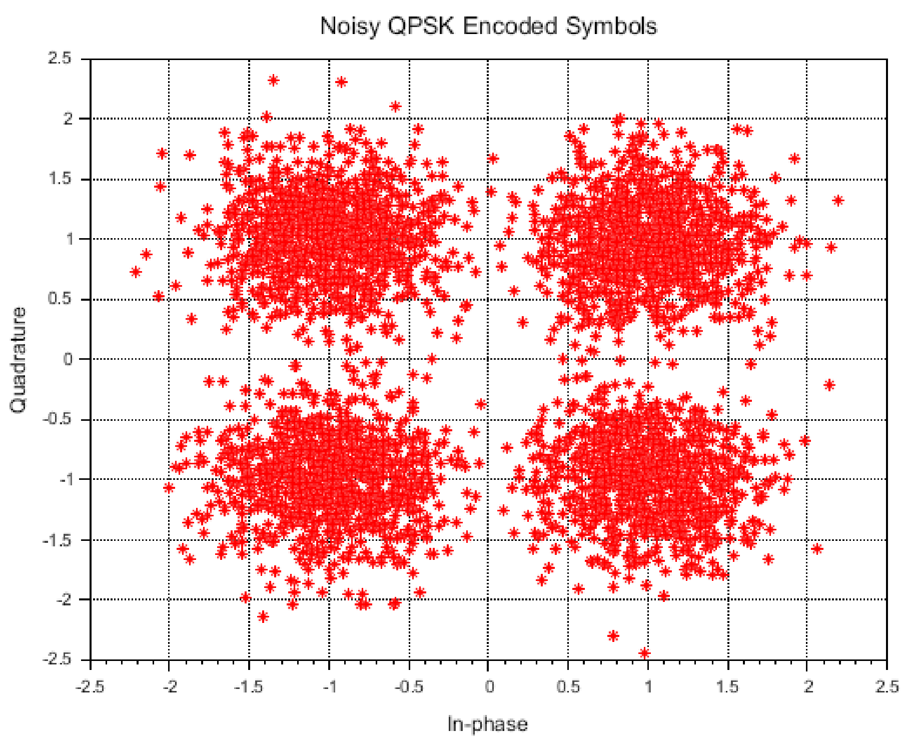

Fig 12: Noisy QPSK Encoded Symbols

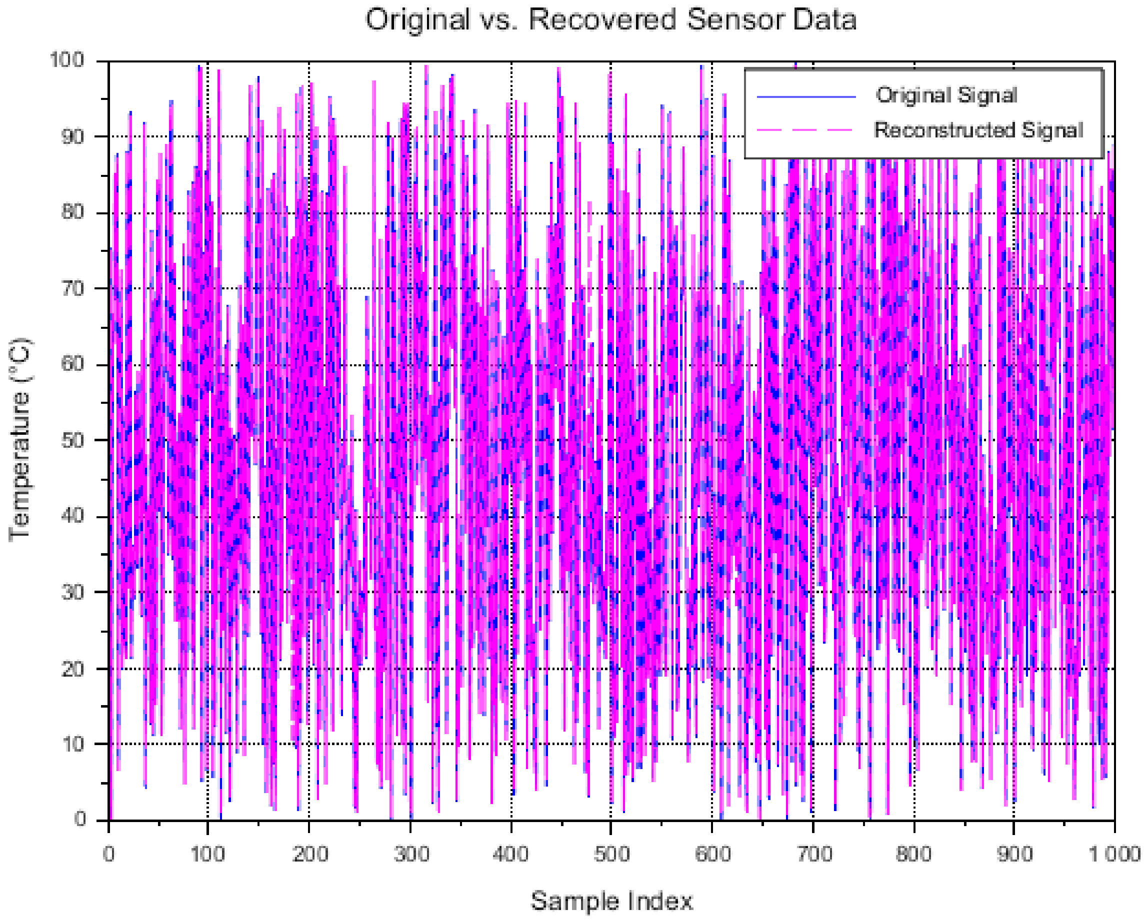

Fig 13: Original vs Recovered Sensor Data

Analyzing Bit Error Rate (BER) and Mean Squared Error (MSE) in PCM and PCM + QPSK Systems

PCM Encoding: BER and MSE Simulation Results

Author Name

Simulation Summary

- Bit Error Rate (BER) before Transmission: 0
- Bit Error Rate (BER) after Transmission: 0.16075
- Mean Squared Error (MSE): 308.27388

Sample Data Comparison

Sample Index	Original Data	Recovered Data
1	21.132487	21.568627
112	0.3717311	0.3921569
223	11.756131	11.764706
334	33.933045	34.117647
445	73.871156	67.450980
556	58.836574	8.627451
667	23.789928	23.921569
778	41.362286	38.039216
889	90.678961	90.588235
1000	88.423489	75.294118

Table 1: Original vs Recovered Data Samples

BER Comparison

- BER before Transmission: 0
- BER after Transmission: 0.16075

Simulation Status: Completed!

PCM + QPSK Encoding: BER and MSE Simulation Results

Simulation Summary

- Bit Error Rate (BER) before Transmission: 0
- Bit Error Rate (BER) after Transmission: 0.00225
- Mean Squared Error (MSE): 5.9666864

Sample Data Comparison

Sample Index	Original Data	Recovered Data
1	5.2153181	5.0980392
112	67.254954	67.45098
223	17.740387	17.647059
334	73.225963	73.333333
445	41.093095	41.176471
556	92.127208	92.156863
667	10.364181	10.196078
778	92.792931	92.941176
889	27.356332	27.45098
1000	97.717524	97.647059

Table 2: Original vs Recovered Data Samples (PCM + QPSK)

BER Comparison

- BER before Transmission: 0
- BER after Transmission: 0.00225

Simulation Status: Completed!

Key Comparison Insights

1. **Bit Error Rate (BER) Improvements:**
   - The PCM-only system has a BER after transmission of 0.155625, indicating significant signal corruption due to noise.
   - The PCM + QPSK system reduces this drastically to 0.00225, demonstrating enhanced noise resistance and more reliable data transmission.

2. **Mean Squared Error (MSE) Reduction:**
   - The PCM-only system exhibits an MSE of 239.24461, meaning higher distortion in reconstructed data.
   - The PCM + QPSK system achieves an MSE of 5.9666864, indicating a much more accurate signal recovery and improved data integrity.

3. **Original vs. Recovered Data Accuracy**
   - In the PCM-only system, errors in recovered sensor values are more pronounced due to direct transmission over a noisy channel.
   - The PCM + QPSK system shows a much closer match between original and recovered values, thanks to QPSK modulation's ability to encode data efficiently while minimizing transmission errors.

CONCLUSION

This project successfully demonstrates the effectiveness of PCM + QPSK encoding in improving ground-based telemetry communication performance. By comparing the PCM-only system to the enhanced PCM + QPSK system, the results highlight significant improvements in signal reliability, noise resilience, and data accuracy.

The Bit Error Rate (BER) after transmission in the PCM-only system was 0.155625, whereas the PCM + QPSK system achieved a much lower BER of 0.00225, proving that QPSK modulation significantly reduces transmission errors. Furthermore, the Mean Squared Error (MSE) improved from 239.24461 (PCM-only) to 5.9666864 (PCM + QPSK), showcasing a drastic enhancement in signal reconstruction quality.

The sample comparisons between original and recovered data further confirm that the PCM + QPSK system maintains greater fidelity in noisy environments. The refined QPSK modulation scheme, by encoding two bits per symbol, effectively mitigates noise distortions, ensuring robust signal integrity throughout transmission.

By utilizing Scilab for system simulations, this study provides a cost-effective and hardware-independent approach to evaluating real-world communication systems. These findings reinforce the importance of modulation techniques in digital telemetry applications, offering a scalable and efficient method for reliable data exchange in terrestrial communication networks.
Future research can explore adaptive error correction, alternative modulation techniques like QAM, and channel coding improvements to further refine telemetry performance and ensure seamless data transmission in harsh environmental conditions.

REFERENCES

[1] E. Bensikaddour, B. Nasri, A. Benikhlef, and D. E. Baba Hamed, "Acquisition of Satellite Telemetry Using a Low-Cost FPGA Board," *2023 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, DOI: 10.1109/ISNCC58260.2023.10323969. https://ieeexplore.ieee.org/document/10323969/references

[2] T. Pratt, C. W. Bostian, and J. Allnutt, "Development of Software Defined Radio-Based Satellite Telemetry and Telecommand System," *IEEE Xplore*, DOI: 10.1109/8905946. https://ieeexplore.ieee.org/document/8905946/references

[3] J.G. Proakis and M. Salehi, "Digital Communications," *McGraw-Hill*, 2008. https://openlibrary.org/books/OL21874530M/Digital_communications

[4] S. Haykin, "Digital Communications," *Wiley*, 2013. https://archive.org/details/digitalcommunica0000hayk

[5] M. Richharia, "Satellite Communications Systems: Design Principles," *McGraw-Hill*, 2019. https://link.springer.com/book/10.1007/978-1-349-14964-3

[6] A. Goldsmith, "Wireless Communications," *Cambridge University Press*, 2005. https://www.cambridge.org/core/books/wireless-communications/800BA8A8211FBECB133A7BB77CD2E2BD

[7] A. Viterbi, "Principles of Coherent Communication," *McGraw-Hill*, 1995. https://archive.org/details/principlesofcohe00vite