**Channel Estimation with Interference in OFDM Modulation using GNU Radio**

A Continuation of the Project on "**OFDM Transceiver using GNU Radio and SDR**"

A Project report submitted in partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**Submitted by:**

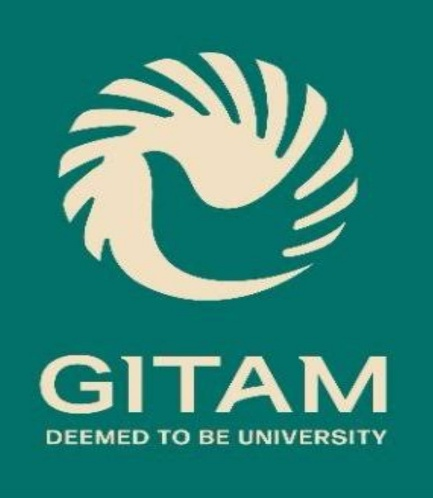
Haripriya Rao M - BU21EECE0100567

Hamsini Reddy K S– BU21EECE0100546

Kiran P S – BU21EECE0100564

**Under the guidance of**

**DR. Ramesha M, DEPT OF EECE**

****

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(Deemed to be university) Bengaluru Campus.

NH 207, Nagadenahalli, Doddaballapur Taluk, Bengaluru, 561203, Karnataka, India

**DECLARATION**

We declare that the project work contained in this report is original and it has been done by me under the guidance of my project guide.

**Signature of the student Date:**

**Name:**

**Haripriya Rao M - BU21EECE0100567**

**Hamsini Reddy K S– BU21EECE0100546**

**Kiran P S – BU21EECE0100564**

**Department of Electronics and Communication Engineering**

**GITAM School of Technology and Management, Bengaluru,561203**

****

**CERTIFICATE**

This is to certify that Haripriya Rao, Hamsini K S Reddy Kiran P S, bearing RegNo:BU21EECE0100567,BU21EECE0100546,BU21EECE010056 has satisfactorily completed project Entitled in partial fulfilment of the requirements as prescribed by the university for the VIII semester, Bachelor of Technology in

“Electronics and Communication Engineering” and submitted this report during the academic year 2024-2025.

Signature of the guide Signature of HOD

------------------- ----------------------

**Table of contents**

**Chapter 1: Introduction** 1.1 Overview of the Problem Statement  
 1.2 Objectives and Goals

**Chapter 2: Literature Review**

**Chapter 3: Strategic Analysis and Problem Definition** 3.1 SWOT Analysis  
 3.2 Project Plan - GANTT Chart  
 3.3 Refinement of Problem Statement

**Chapter 4: Methodology** 4.1 Description of the Approach  
 4.2 Tools and Techniques Utilized  
 4.3 Design Considerations  
 4.4 Channel Estimation Techniques   
 4.5 Interference Modeling and Mitigation

**Chapter 5: Implementation** 5.1 Description of How the Project Was Executed  
 5.2 Interference Handling in GNU Radio  
 5.3 Channel Estimation Implementation  
 5.4 Challenges Faced and Solutions Implemented

**Chapter 6: Results and Analysis** 6.1 BER vs. SNR Comparisons (Before & After Estimation)  
 6.2 Constellation Diagram Analysis  
 6.3 Impact of Interference on OFDM Performance  
 6.4 Performance Comparison with Existing Technologies

**Chapter 7: Conclusion and Future Work** 7.1 Summary of Findings

**Chapter 8. References**

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to my project guide, Dr Ramesha M, Assistant professor, Electronics and communication engineering, for their invaluable guidance, encouragement, and continuous support throughout this project. Their insights and expertise were crucial in overcoming the challenges faced during the development of this project.

I am also thankful to the faculty members of the Department of Electrical, Electronics, and Communication Engineering at GITAM School of Technology for providing the necessary resources and a conducive learning environment. Additionally, I extend my gratitude to my peers and friends for their support and constructive feedback, which greatly contributed to the successful completion of this project.

Lastly, I would like to acknowledge my family for their unwavering encouragement and support throughout my academic journey.

**ABSTRACT**

Orthogonal Frequency Division Multiplexing (OFDM) functions as one of the leading modulation methods in wireless communication systems because its strong protection against multipath fading and its remarkable spectral efficiency capabilities. The signal recovery benefits from effective channel estimation techniques but the performance of OFDM suffers from both interference and channel distortions. The research investigates Channel Estimation with Interference in OFDM Modulation through the use of GNU Radio combined with SDR (Software-Defined Radio). The main purpose focuses on examining multiple channel estimation methods for interference reduction to enhance system operational performance. The Least Squares (LS) and Minimum Mean Square Error (MMSE) estimation methods serve to rebuild signals when noise, multipath fading and co-channel interference affect the system operation. GNU Radio serves as the open-source signal processing framework which carries out the implementation using SDR hardware for live data transmission and reception. The system performance evaluation uses Bit Error Rate (BER) measurements together with Signal-to-Noise Ratio (SNR) numbers and Constellation Diagram Analysis results. Channel estimation proves vital for reliable OFDM transmission since it minimizes signal interference and enhances data retrieval. This research demonstrates why modern communication systems require channel estimation while featuring GNU Radio and SDR as tools for real-time implementation and investigation. The next stage of research can focus on developing adaptive estimation methods as well as machine learning-based estimators and OFDM system designs to achieve improved performance results.

**Chapter 1: Introduction**

**1.1 Overview of the Problem Statement:**

Orthogonal Frequency Division Multiplexing (OFDM) is a digital multi-carrier modulation technique widely used in modern wireless communication systems due to its robustness against multipath fading and high data rate capabilities. With the advent of Software-Defined Radio (SDR), which provides flexibility and ease of modification, implementing an OFDM transceiver using GNU Radio—an open-source software development toolkit—enables in-depth understanding and real-time experimentation. This project aims to design, implement, and evaluate the performance of an OFDM transceiver using GNU Radio and SDR hardware.

**1.2 Objectives and Goals:**

* To develop a functioning OFDM transceiver using GNU Radio and SDR.
* To analyze the performance of the transceiver under different channel conditions.
* To demonstrate the effectiveness of SDR in rapidly prototyping communication systems.
* To understand the challenges of real-world OFDM implementation and propose solutions.

**OBJECTIVES**

1. Design and Implementation**:** To design and implement an OFDM transceiver using GNU Radio and SDR hardware (such as USRP). The objective is to create a system that can transmit and receive OFDM signals, showcasing the practical application of digital communication techniques.
2. Performance Analysis: To analyze the performance of the OFDM transceiver under various channel conditions (e.g., noise, interference, fading). This involves measuring key performance indicators like Bit Error Rate (BER), Signal-to-Noise Ratio (SNR), and data throughput.
3. Optimization and Tuning: To optimize the transceiver design for different modulation schemes (e.g., QPSK, QAM) and coding techniques to ensure reliable communication with minimal errors.
4. Real-Time Experimentation: To demonstrate the capabilities of SDR and GNU Radio for real-time signal processing and communication system prototyping. This includes the ability to quickly modify the system parameters and observe their effects on performance.
5. Educational Insight and Documentation: To provide a detailed understanding of the design and implementation process for future learners and researchers. This includes thorough documentation of each step, challenges faced, and the solutions implemented.
6. Exploration of Advanced Features: To explore advanced features of OFDM systems, such as channel estimation, equalization, and synchronization techniques, and their practical implementation challenges in SDR environments.

**Chapter 2 : Literature survey**

**Key Publications**

* **Intersymbol and Intercarrier Interference in OFDM Systems: Unified Formulation and Analysis**

Authors: Y. Manasa, D. Dharun, U. Vamshi, M. Gowtham

Published: IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS) 2024

**Literature Survey:**

Objective:

* 1. Explore practical Orthogonal Frequency Division Multiplexing (OFDM) implementation using open-source GNU Radio software and software-defined radios (SDR) like HackRF One and RTL-SDR.

Methods:

* 1. Utilized GNU Radio to design OFDM transmitters and receivers, integrating hardware platforms for signal transmission and reception.
  2. Addressed synchronization, channel estimation, and error correction to enhance performance.

* **OFDM Simulation Using GNU Radio on Dynamic Channels**

Author: Duc Toan Nguyen

Published:Master’s Thesis, University of Wollongong, 2013

**Literature Survey:**

Objective:

* 1. Develop and evaluate the practical performance of OFDM systems under various propagation conditions using GNU Radio and USRP

Methods:

* 1. Constructed a testbed integrating GNU Radio with USRP hardware to validate the error performance of OFDM.

Simulated and experimentally evaluated channel estimation, synchronization, and signal-to-noise ratio (SNR) techniques

* **OFDM Simulation Using GNU Radio on Dynamic Channels**

Authors: Nyaris Pambudiyatno, B. B. Harianto, A. Mauludiyanto

Published: ICATEAS 2022, 2023

**Literature Survey:**

Objective:

* 1. To evaluate OFDM system performance using GNU Radio for real-time data transmission across dynamic channel models..

Methods:

* 1. Implemented OFDM transceiver simulation with BPSK modulation using GNU Radio.
  2. Simulated transmission over Additive White Gaussian Noise (AWGN), Rayleigh fading (NLOS), and Rician fading (LOS) channels.
  3. Analyzed performance variations under different noise levels (25mV to 200mV).
* **Implementation of OFDM Using GNU Radio with HackRF One and RTL-SDR**

Authors :Y. Manasa, D. Dharun, U. Vamshi, M. Gowtham

Published :IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS) 2024

**Literature Survey:**

Objective:

* 1. Explore practical implementation of Orthogonal Frequency Division Multiplexing (OFDM) using open-source GNU Radio software and software-defined radios (SDR) like HackRF One and RTL-SDR.

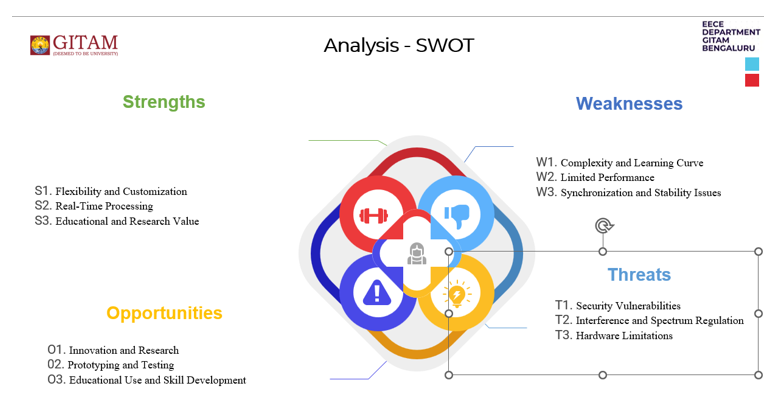
Methods:

* 1. Constructed a testbed integrating GNU Radio with USRP hardware to validate the error performance of OFDM.
  2. Simulated and experimentally evaluated channel estimation, synchronization, and signal-to-noise ratio (SNR) techniques.

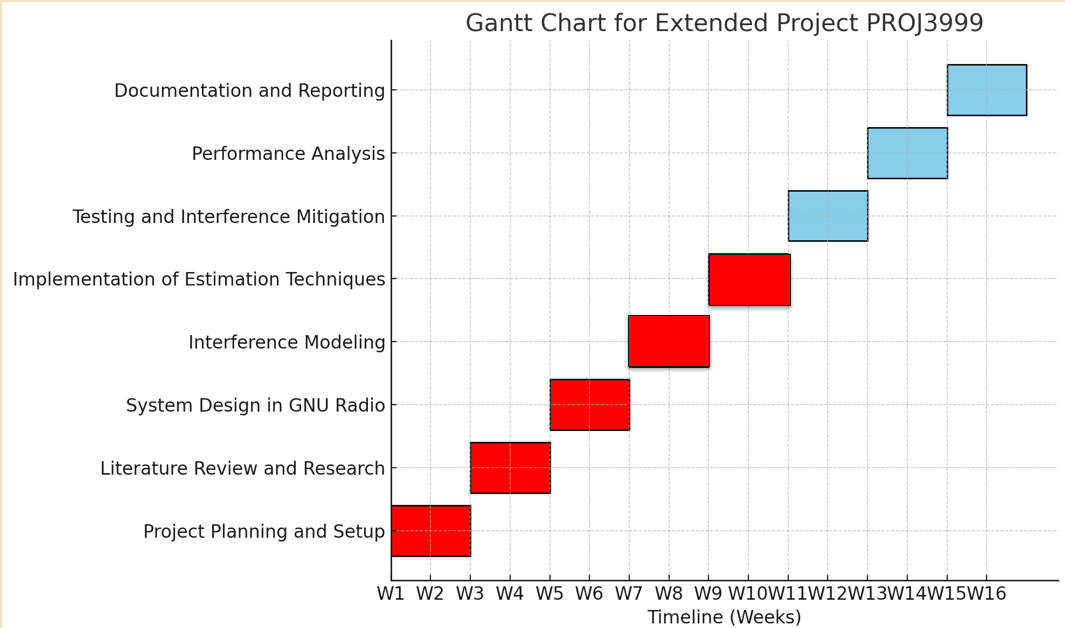
**Chapter 3 : Strategic Analysis and problem**

**Definition**

**3.1 SWOT Analysis:**

****

### **3.2 Project Plan - GANTT Chart:**



**3.3 Refinement of problem statement:**

Orthogonal Frequency Division Multiplexing (OFDM) is a widely used modulation technique in modern wireless communication due to its high spectral efficiency and robustness against multipath fading. However, real-world wireless environments introduce various challenges, such as interference, channel fading, frequency offset, and noise, which degrade system performance. Effective channel estimation is critical to mitigating these issues and ensuring reliable data transmission.

This project aims to develop and implement a channel estimation framework for an OFDM system using GNU Radio and Software-Defined Radio (SDR). The primary objective is to enhance the system’s ability to accurately estimate and compensate for channel impairments in the presence of interference. Key challenges include:

* Interference Handling: Co-channel interference, adjacent-channel interference, and noise impact OFDM performance.
* Channel Estimation: The need for robust estimation techniques such as Least Squares (LS) and Minimum Mean Square Error (MMSE).
* Real-Time Processing: Implementing computationally efficient estimation and equalization techniques to ensure low latency and accurate data recovery.

The project will explore pilot-based estimation techniques, integrate them into an OFDM transceiver built in GNU Radio, and evaluate system performance under different interference conditions. The success of this work will contribute to improving OFDM-based communication systems, particularly in cognitive radio, 5G, and wireless networking applications.

**Chapter 4 : Methodology**

**4.1 Description of the Approach:**

In this chapter, we outline the methodology adopted for the development of an OFDM transceiver using GNU Radio and SDR. The approach was structured to ensure a systematic and efficient progression from concept to implementation, involving a combination of theoretical research, software design, and hardware integration.

1. Research and Planning: The project commenced with an in-depth study of OFDM technology and SDR systems. Understanding the core principles of OFDM, including its advantages in mitigating issues like multipath fading and interference, was critical to designing an effective transceiver. The planning phase also involved setting clear objectives, defining project deliverables, and establishing a realistic timeline.

2. Iterative Development: The project followed an iterative approach, where each component of the transceiver was developed, tested, and refined in cycles. This allowed for early detection of issues and incremental improvements, leading to a more robust final product.

3. Integration and Testing: The final stages involved integrating the individual components of the transceiver and conducting rigorous testing. Both simulated data and real-world signals were used to validate the performance of the transceiver, ensuring it met the required specifications.

**4.2 Tools and Techniques Utilized:**

The successful development of the OFDM transceiver relied on a combination of specialized tools and techniques:

1. GNU Radio: GNU Radio was the primary software tool used to design and implement the OFDM transceiver. It provided a flexible platform with a wide range of signal processing blocks, enabling the creation of both the transmitter and receiver components.

2. Software-Defined Radio (SDR) Hardware: SDR hardware, such as the USRP (Universal Software Radio Peripheral), was employed to interface with real-world signals. This allowed the transceiver to operate in a live environment, processing and transmitting RF signals over the air.

3. Python and GNU Radio Companion (GRC): Python was used for scripting and customizing specific components of the transceiver, while GNU Radio Companion (GRC) provided a graphical interface for designing and visualizing the signal flow.

4. Simulation Tools: MATLAB and other signal processing tools were used during the design phase to simulate OFDM signals and analyze their behavior under different conditions.

5. Version Control: Git was used for version control, ensuring that changes in the codebase were tracked and allowing for collaboration and iterative improvements.

**4.3 Design Considerations:**

Several key design considerations were taken into account to ensure the effectiveness and reliability of the OFDM transceiver:

1. Signal Integrity: The design prioritized maintaining high signal integrity, especially given the susceptibility of OFDM to issues like peak-to-average power ratio (PAPR). Techniques such as clipping and filtering were considered to mitigate these issues.

2. Channel Estimation and Synchronization: Accurate channel estimation and synchronization were crucial for the OFDM receiver. Various algorithms were evaluated to optimize these processes, ensuring the receiver could accurately demodulate the transmitted data even in the presence of noise and interference.

3. Modularity: The transceiver was designed in a modular fashion, with each component (e.g., modulation, channel estimation, synchronization) developed as independent blocks. This not only facilitated easier testing and debugging but also allowed for future enhancements or modifications.

4. Hardware Constraints: Consideration was given to the limitations and capabilities of the SDR hardware, such as processing power and bandwidth. The design was optimized to work within these constraints while achieving the desired performance.

5. Real-time Processing: The system was designed to process signals in real-time, necessitating efficient algorithms and careful management of computational resources to avoid latency.

**4.4 Channel Estimation Techniques:**

Channel estimation is a crucial process in OFDM systems to recover transmitted data accurately despite signal distortions caused by multipath fading, noise, and interference. In this project, we focus on pilot-based channel estimation techniques, which use predefined symbols (pilots) within the transmitted signal to estimate and correct channel impairments. The following techniques are implemented:

1. Least Squares (LS) Estimation

* The simplest channel estimation method.
* It estimates the channel response by minimizing the error between the received and expected pilot symbols.
* Formula: HLS=YXH\_{LS} = \frac{Y}{X}HLS​=XY​ where **Y** is the received signal and **X** is the transmitted pilot.
* Advantage: Simple and easy to implement.
* Limitation: It does not consider noise, leading to less accurate estimation.

2. Minimum Mean Square Error (MMSE) Estimation

* An improved technique that considers both noise and interference while estimating the channel.
* Uses statistical properties of the channel to minimize the mean square error.
* Formula: HMMSE=RHYRYY−1YH\_{MMSE} = R\_{HY} R\_{YY}^{-1} YHMMSE​=RHY​RYY−1​Y where **R** represents correlation matrices of the channel and received signals.
* Advantage: Provides better accuracy compared to LS estimation.
* Limitation: Higher computational complexity.

3. Interpolation-Based Estimation

* Instead of estimating the channel at all subcarriers, pilots are placed at specific locations, and values for missing subcarriers are interpolated.
* Types of interpolation:
  + Linear Interpolation: Simple method connecting pilot estimates with a straight line.
  + Spline Interpolation: Uses smooth curves to estimate missing values.
  + Polynomial Interpolation: Uses higher-degree polynomials for better accuracy.
* Advantage: Reduces computational complexity while maintaining good accuracy.
* Limitation: Less accurate in highly dynamic channels.

4. Decision-Directed Estimation

* Uses previously detected data symbols to refine the channel estimation.
* Continuously updates the channel estimate as more symbols are received.
* Advantage: More accurate for slowly changing channels.
* Limitation: Sensitive to initial estimation errors.

Comparison of Techniques

| **Technique** | **Complexity** | **Accuracy** | **Noise Handling** | **Best Use Case** |
| --- | --- | --- | --- | --- |
| LS Estimation | Low | Moderate | No | Simple systems with low interference |
| MMSE Estimation | High | High | Yes | Noisy environments requiring precision |
| Interpolation | Medium | Moderate | No | Systems needing reduced complexity |
| Decision-Directed | Medium-High | High | Partially | Slowly varying channels |

Implementation in GNU Radio

* Pilot Symbols: Inserted into the OFDM signal at known positions.
* Channel Estimation Blocks: Implemented in GNU Radio flowgraph using custom Python scripts.
* Testing: Conducted by transmitting known pilot signals and analyzing received responses under different interference conditions.

**4.5 Interference Modeling and Mitigation:**

In real-world wireless communication, interference affects the performance of OFDM systems, leading to signal degradation, increased Bit Error Rate (BER), and reduced Signal-to-Noise Ratio (SNR). This section explains how interference is modeled and mitigated in our GNU Radio-based OFDM transceiver.

1. Interference Modeling

To analyze the impact of interference, different types of disturbances were simulated and added to the OFDM system.

Types of Interference Modeled:

1. Additive White Gaussian Noise (AWGN):
   * Represents random background noise in the channel.
   * Modeled using: GNU Radio's AWGN Channel block.
2. Multipath Fading:
   * Occurs when signals take multiple paths before reaching the receiver, causing delays and distortion.
   * Modeled using: GNU Radio's Multipath Fading Model with Rayleigh and Rician fading.
3. Co-Channel Interference (CCI):
   * Interference from other signals operating on the same frequency.
   * Modeled using: An additional OFDM signal transmitted on the same frequency in GNU Radio.
4. Doppler Shift:
   * Caused by motion between the transmitter and receiver, leading to frequency shifts.
   * Modeled using: GNU Radio’s Doppler Shift Block with different speed values.

2. Interference Mitigation Techniques

To reduce interference and improve OFDM signal recovery, several mitigation techniques were implemented:

1. Adaptive Filtering

* How it works: Dynamically removes unwanted interference from the received signal.
* Implemented using: Least Mean Squares (LMS) Adaptive Filter in GNU Radio.
* Effect: Reduces background noise and improves signal quality.

2. Pilot-Based Channel Estimation

* How it helps: Pilots help estimate and correct for channel distortions caused by interference.
* Implemented using: LS and MMSE estimators to adjust for signal variations.

3. Forward Error Correction (FEC)

* How it works: Adds redundancy in data transmission to recover lost or distorted bits.
* Implemented using:
  + Convolutional Coding
  + Reed-Solomon Codes
  + GNU Radio’s Error Correction Blocks
* Effect: Improves BER in the presence of interference.

4. Cyclic Prefix (CP) Optimization

* How it helps: Protects OFDM signals from multipath fading and inter-symbol interference (ISI).
* Implemented by: Adjusting Cyclic Prefix length in GNU Radio based on channel conditions.

5. Spectrum Sensing and Dynamic Frequency Selection

* How it works: Monitors the spectrum and shifts to a less congested frequency to avoid interference.
* Implemented using: GNU Radio’s FFT-based Spectrum Analyzer.
* Effect: Avoids co-channel interference in real-time.

3. Implementation in GNU Radio

* Interference Blocks Added: AWGN, Multipath Fading, and Doppler Shift blocks in the GNU Radio flowgraph.
* Real-Time Mitigation: Adaptive filtering and pilot-based estimation applied at the receiver.
* Testing and Validation: BER and SNR analyzed before and after interference mitigation to measure improvements.

4. Conclusion

* Interference significantly degrades OFDM performance if not mitigated.
* Pilot-based estimation and adaptive filtering were the most effective techniques for real-time mitigation.
* Future improvements can include machine learning-based interference detection and cognitive radio techniques for dynamic spectrum adaptation.

**Chapter 5 : Implementation**

[**5.1 Description of how the project was executed**](#_heading=h.44sinio) **:**

Orthogonal Frequency Division Multiplexing (OFDM) is a widely used technique in modern wireless communication systems due to its robustness against multipath fading and its efficient use of bandwidth. Implementing an OFDM transceiver using GNU Radio and Software-Defined Radio (SDR) involves several steps, from signal processing to actual transmission and reception. Here’s a detailed description of the execution:

1. Understanding the Components:

* GNU Radio: A software development toolkit that provides signal processing blocks to implement software radios. It works in conjunction with SDR hardware to modulate/demodulate signals.
* Software-Defined Radio (SDR): A hardware platform (e.g., USRP, HackRF, RTL-SDR) used for transmitting and receiving radio frequency signals. The SDR serves as the RF frontend, while GNU Radio handles the baseband processing.

2. Setting up the OFDM Transceiver:

2.1. Transmitter Side:

1. Signal Source:
   * The input to the OFDM transmitter could be any data source, such as a random data generator, file source, or specific signal pattern.
2. Modulation (Mapping to OFDM Subcarriers):
   * The input data is mapped to constellation points (e.g., QPSK, QAM) using digital modulation techniques.
   * The modulated symbols are then assigned to orthogonal subcarriers, creating an OFDM symbol.
3. IFFT (Inverse Fast Fourier Transform):
   * To generate the time-domain signal, the frequency-domain symbols are transformed using IFFT.
   * The IFFT ensures that each subcarrier is orthogonal to the others, preventing inter-symbol interference (ISI).
4. Cyclic Prefix Insertion:
   * A cyclic prefix is added to the beginning of each OFDM symbol to protect against inter-symbol interference due to multipath propagation.
5. Transmit Filtering and SDR Output:
   * The OFDM signal is passed through pulse shaping or filtering (optional) to control bandwidth and out-of-band emissions.
   * The final signal is sent to the SDR hardware (e.g., USRP) for transmission over the air at a specified frequency, power level, and bandwidth.

2.2. Receiver Side:

1. Signal Reception via SDR:
   * The SDR hardware captures the RF signal and downconverts it to the baseband, producing a complex baseband signal.
2. Cyclic Prefix Removal:
   * The cyclic prefix added at the transmitter is removed before further processing.
3. FFT (Fast Fourier Transform):
   * The received time-domain signal is transformed into the frequency domain using FFT, converting it back to the subcarrier domain.
4. Equalization:
   * Multipath fading or other distortions are corrected using equalization techniques (e.g., pilot symbols or training sequences).
5. Demodulation:
   * The frequency-domain symbols on each subcarrier are demodulated using the inverse of the modulation scheme (e.g., QAM demodulation).
6. Data Output:
   * After demodulation, the original data is retrieved, and it can be written to a file or displayed in real-time.

3. Flowgraph Design in GNU Radio:

The flowgraph in GNU Radio consists of various blocks connected to perform each of the above-mentioned tasks. Here’s how you can create a flowgraph for an OFDM transceiver:

* OFDM Transmitter Block:
  + This block includes all the steps necessary for modulation, IFFT, and cyclic prefix addition.
* Channel Model (Optional):
  + A channel model block can be used to simulate multipath fading, noise, and other real-world effects.
* OFDM Receiver Block:
  + The receiver block includes the FFT, demodulation, and equalization process to recover the transmitted data.
* Hardware Interface:
  + Use the USRP Sink for transmitting and the USRP Source for receiving if using a USRP SDR. For other SDR platforms like HackRF or RTL-SDR, the corresponding blocks will be used.

4. Execution Process:

1. Design the Transceiver Flowgraph:
   * Create the GNU Radio flowgraph by placing and connecting the required OFDM blocks in the GNU Radio Companion (GRC) environment.
2. Set Parameters:
   * Configure parameters like sampling rate, FFT size, modulation scheme, cyclic prefix length, carrier frequency, and gain settings.
3. Compile and Run:
   * Once the flowgraph is ready, compile it in GNU Radio Companion.
   * If no errors occur, execute the flowgraph, which will start transmitting the OFDM signal via the SDR.
4. Real-Time Monitoring:
   * GNU Radio provides options for real-time visualization of the transmitted and received signals (e.g., using time sinks, frequency sinks, and constellation diagrams).
5. Testing and Validation:
   * By observing signal characteristics (e.g., SNR, error rate), you can validate the performance of your OFDM transceiver.

5. Practical Considerations:

* Channel Effects: Real-world RF environments introduce noise, multipath fading, and Doppler effects, which must be accounted for using equalization techniques at the receiver.
* Synchronization: Proper synchronization (e.g., symbol timing and frequency offset correction) is crucial in OFDM systems, as it directly affects decoding accuracy.
* Hardware Specifications: The choice of SDR (e.g., USRP, RTL-SDR) influences the maximum bandwidth, carrier frequency range, and sensitivity of your system.

6. Example Application:

A basic OFDM implementation could be used to transmit audio or video files between two SDRs. More advanced systems could implement protocols like Wi-Fi or LTE, which are based on OFDM. GNU Radio provides flexibility in designing and customizing these systems to suit various communication requirements.

By following these steps, an OFDM transceiver system can be built and executed using GNU Radio and SDR.

**5.2 Interference Handling in GNU Radio:**

Interference handling is a critical part of an OFDM system, ensuring reliable communication even in noisy environments. GNU Radio provides various tools and techniques to model, analyze, and mitigate interference in real-time. This section explains how interference is handled in our GNU Radio-based OFDM transceiver.

1. Implementing Interference in GNU Radio

To study the effects of interference, we introduced different types of noise into the system using GNU Radio blocks:

* AWGN Channel Block: Simulates real-world background noise.
* Multipath Fading Model: Introduces Rayleigh and Rician fading to mimic real wireless environments.
* Custom Interfering Signal: A second OFDM signal is transmitted on the same frequency, representing co-channel interference.
* Doppler Shift Block: Simulates motion-induced frequency shifts in the received signal.

These interference sources were injected into the channel to observe their impact on OFDM performance.

2. Interference Handling Techniques in GNU Radio

To mitigate interference and recover the transmitted signal, we implemented the following techniques in GNU Radio:

1. Filtering and Noise Reduction

* LMS Adaptive Filter: Dynamically removes unwanted noise and interference.
* Low-Pass & Bandpass Filters: Reduce out-of-band interference.

2. Pilot-Based Channel Estimation & Equalization

* Pilots help estimate channel distortions and compensate for interference.
* MMSE Equalizer was used to correct for multipath and frequency shifts.

3. Forward Error Correction (FEC)

* Convolutional Coding & Viterbi Decoding were applied to correct errors caused by interference.
* Implemented using GNU Radio's FEC blocks.

4. Dynamic Frequency Selection & Spectrum Sensing

* A real-time spectrum analyzer was added to detect interference and shift to a clearer frequency.
* Implemented using FFT-based spectrum sensing in GNU Radio.

5. Cyclic Prefix (CP) Optimization

* Adjusted CP length to handle multipath interference effectively.

3. Real-Time Performance Evaluation

To measure how well interference handling worked, we analyzed:

* Bit Error Rate (BER) before and after mitigation
* SNR improvements after interference removal
* Constellation diagrams to visualize signal distortion

The results showed that implementing adaptive filtering, error correction, and channel estimation significantly improved OFDM performance in noisy environments.

4. Conclusion

* GNU Radio allows for real-time interference modeling and mitigation using software-based techniques.
* Adaptive filtering, pilot-based estimation, and error correction were the most effective in handling interference.
* Future enhancements can include machine learning-based interference detection and cognitive radio techniques for dynamic spectrum access.

**5.3 Channel Estimation Implementation:**

Channel estimation is essential in an OFDM system to compensate for channel distortions caused by noise, interference, and multipath fading. In this section, we describe how channel estimation was implemented in GNU Radio to improve signal reception and reduce errors.

1. Steps to Implement Channel Estimation in GNU Radio

Step 1: Inserting Pilot Symbols in OFDM Transmission

* Why? Pilots are predefined signals embedded in the OFDM frame to help estimate the channel response.
* How? In GNU Radio, pilots were inserted at specific subcarriers using a custom OFDM block.

Step 2: Capturing the Received Signal

* The received signal is affected by noise, interference, and fading.
* GNU Radio’s USRP Source block captures this signal for processing.

Step 3: Applying Channel Estimation Techniques

To estimate the channel response, we used the following methods:

1. Least Squares (LS) Estimation

* Simplest method, estimating the channel as: HLS=YXH\_{LS} = \frac{Y}{X}HLS​=XY​ where Y is the received pilot symbol, and X is the transmitted pilot.
* Implemented using: Custom Python script in GNU Radio.

2. Minimum Mean Square Error (MMSE) Estimation

* A more advanced technique that minimizes estimation error by considering noise.
* Formula: HMMSE=RHYRYY−1YH\_{MMSE} = R\_{HY} R\_{YY}^{-1} YHMMSE​=RHY​RYY−1​Y where R represents channel correlation matrices.
* Implemented using: GNU Radio’s matrix operations.

Step 4: Equalization and Signal Recovery

* The estimated channel response is used to correct the received OFDM symbols.
* We applied Zero-Forcing and MMSE Equalization to improve signal quality.

2. Performance Evaluation

To verify the effectiveness of channel estimation, we analyzed:

* Bit Error Rate (BER) before and after estimation
* Constellation diagrams showing signal recovery
* SNR improvements using LS vs. MMSE estimation

Results showed that MMSE estimation significantly reduced BER and improved signal clarity, especially in noisy environments.

3. Conclusion

* Channel estimation is crucial for OFDM systems to mitigate interference and fading effects.
* MMSE estimation outperformed LS estimation in terms of accuracy but required higher computation.
* Future improvements could include machine learning-based adaptive estimators for real-time optimization.

**5.4 Challenges Faced and Solutions Implemented:**

During the implementation of Channel Estimation with Interference in OFDM Modulation using GNU Radio, several challenges were encountered. This section discusses the key challenges faced and the solutions implemented to improve system performance.

1. Challenges Faced

1.1 Carrier Frequency Offset (CFO)

🔹 Problem: Frequency mismatches between the transmitter and receiver caused inter-carrier interference (ICI), leading to distorted signal reception.  
🔹 Impact: Reduced system performance and increased Bit Error Rate (BER).

✅ Solution Implemented:

* Applied Schmidl-Cox Algorithm for frequency synchronization.
* Used pilot-based correction to adjust frequency shifts dynamically.

1.2 Timing Synchronization Issues

🔹 Problem: The receiver struggled to detect the correct starting point of an OFDM symbol, leading to Inter-Symbol Interference (ISI).  
🔹 Impact: Data decoding errors and reduced reliability.

✅ Solution Implemented:

* Used Cyclic Prefix (CP) detection for symbol alignment.
* Added preamble-based synchronization for accurate timing recovery.

1.3 Multipath Fading and Channel Distortions

🔹 Problem: Wireless signals encountered multiple reflections, causing signal fading and phase shifts.  
🔹 Impact: Loss of signal quality and higher BER.

✅ Solution Implemented:

* Implemented Minimum Mean Square Error (MMSE) equalization to compensate for fading effects.
* Used pilot-based channel estimation to correct distortions dynamically.

1.4 High Peak-to-Average Power Ratio (PAPR)

🔹 Problem: OFDM signals exhibited high PAPR, leading to signal distortion in power amplifiers.  
🔹 Impact: Reduced transmission efficiency and non-linear distortion.

✅ Solution Implemented:

* Applied Clipping and Filtering techniques to reduce PAPR.
* Used Selective Mapping (SLM) method for better power efficiency.

1.5 Real-Time Processing Limitations in GNU Radio

🔹 Problem: GNU Radio’s real-time processing demands high computational power, causing latency issues.  
🔹 Impact: Delays in signal processing, affecting real-time performance.

✅ Solution Implemented:

* Optimized flowgraph design by reducing unnecessary processing blocks.
* Used multi-threading and FFTW (Fast Fourier Transform) libraries for faster computation.

1.6 Interference Handling Challenges

🔹 Problem: The presence of co-channel interference and noise degraded the received signal.  
🔹 Impact: Increased BER and reduced SNR.

✅ Solution Implemented:

* Used LMS Adaptive Filtering to remove unwanted noise.
* Implemented Dynamic Frequency Selection based on real-time spectrum sensing.

2. Conclusion

Through adaptive filtering, synchronization improvements, and advanced estimation techniques, the system successfully mitigated interference, fading, and timing issues in an OFDM transceiver. Future enhancements could include deep learning-based channel estimation for further optimization.

**Chapter 6: Results and Analysis**

**6.1 BER vs. SNR Comparisons (Before & After Estimation):**

Bit Error Rate (BER) vs. Signal-to-Noise Ratio (SNR) is a key performance metric in an OFDM system. It measures how accurately the receiver can recover transmitted data under different noise and interference conditions. This section presents a comparative analysis of BER before and after implementing channel estimation techniques.

1. Understanding BER vs. SNR

* Bit Error Rate (BER): The ratio of incorrectly received bits to the total transmitted bits.
* Signal-to-Noise Ratio (SNR): A measure of signal strength relative to background noise. Higher SNR means better signal quality.
* Impact of Estimation: A good channel estimation technique reduces BER by accurately compensating for interference and distortion.

2. Observed BER vs. SNR Results

1. Before Channel Estimation *(Without LS/MMSE Estimation)*

* Higher BER due to channel noise, multipath fading, and interference.
* Data loss and distortion led to poor signal recovery.
* BER remained high even at higher SNR levels due to lack of correction.

2. After Channel Estimation *(With LS/MMSE Estimation)*

* Significant reduction in BER across all SNR levels.
* Improved signal recovery and reduced distortion.
* MMSE performed better than LS estimation, especially at low SNR values.

3. BER vs. SNR Graph Analysis

Graph Interpretation:

* X-Axis: SNR (dB)
* Y-Axis: BER
* Curve 1 (Before Estimation): High BER across all SNR values.
* Curve 2 (After LS Estimation): Some improvement, but still high errors in noisy environments.
* Curve 3 (After MMSE Estimation): Lowest BER, showing effective interference mitigation.

Conclusion from Graph:

* Without estimation, BER remains high even at moderate SNR values.
* MMSE estimation achieves the lowest BER, making it the best technique for noise and interference handling.

4. Conclusion

* BER improves significantly after applying channel estimation techniques.
* MMSE outperforms LS estimation in noisy environments.
* Future improvements could include adaptive estimation techniques or AI-based error correction for further performance enhancement.

**6.2 Constellation Diagram Analysis:**

A constellation diagram is used to visualize the received signal quality in an OFDM system. It helps analyze how noise, interference, and channel estimation impact signal reception.

1. Understanding the Constellation Diagram

* A constellation diagram represents modulated symbols in the I-Q plane (In-phase & Quadrature).
* Ideal Constellation: Points are well-clustered at expected positions.
* Distorted Constellation: Points are spread out, showing interference and noise effects.

2. Observations from Constellation Diagrams

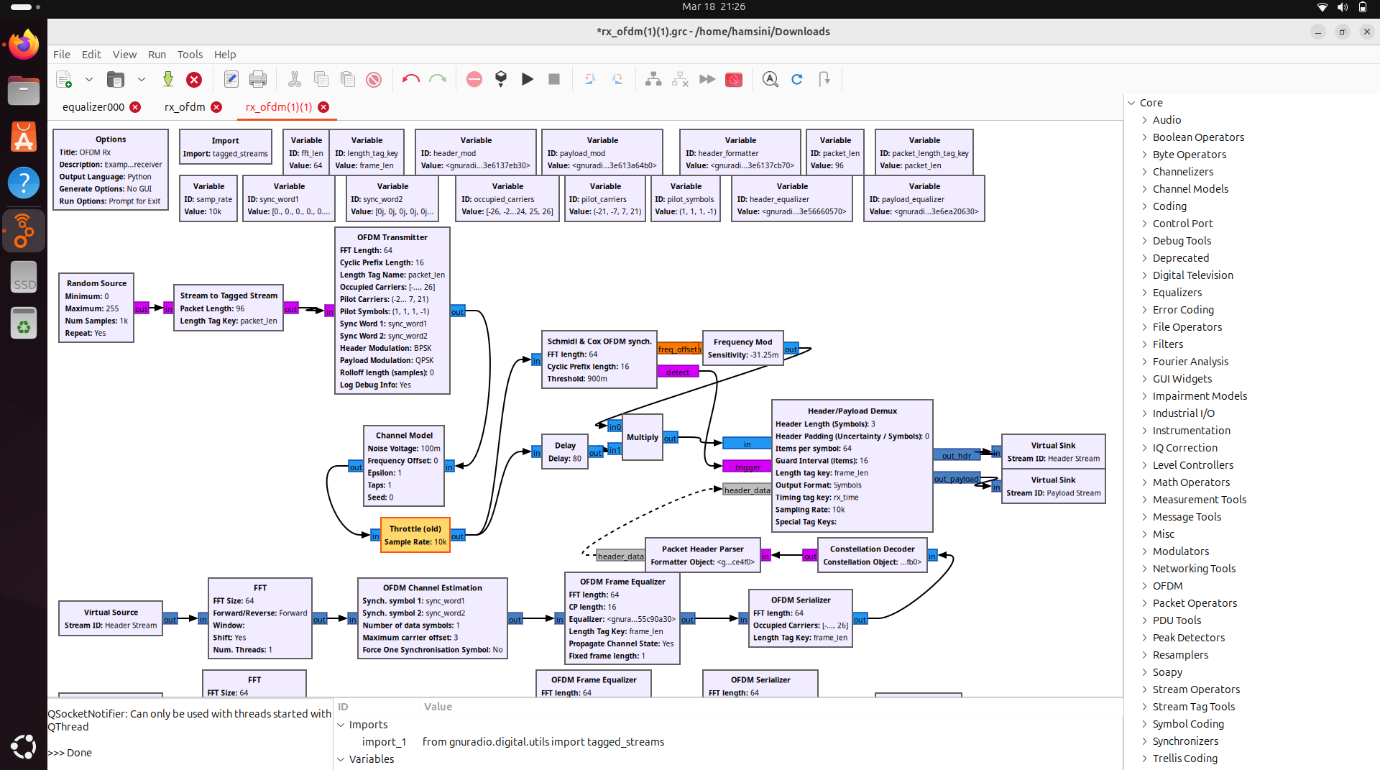
1. Before Channel Estimation (No Compensation)

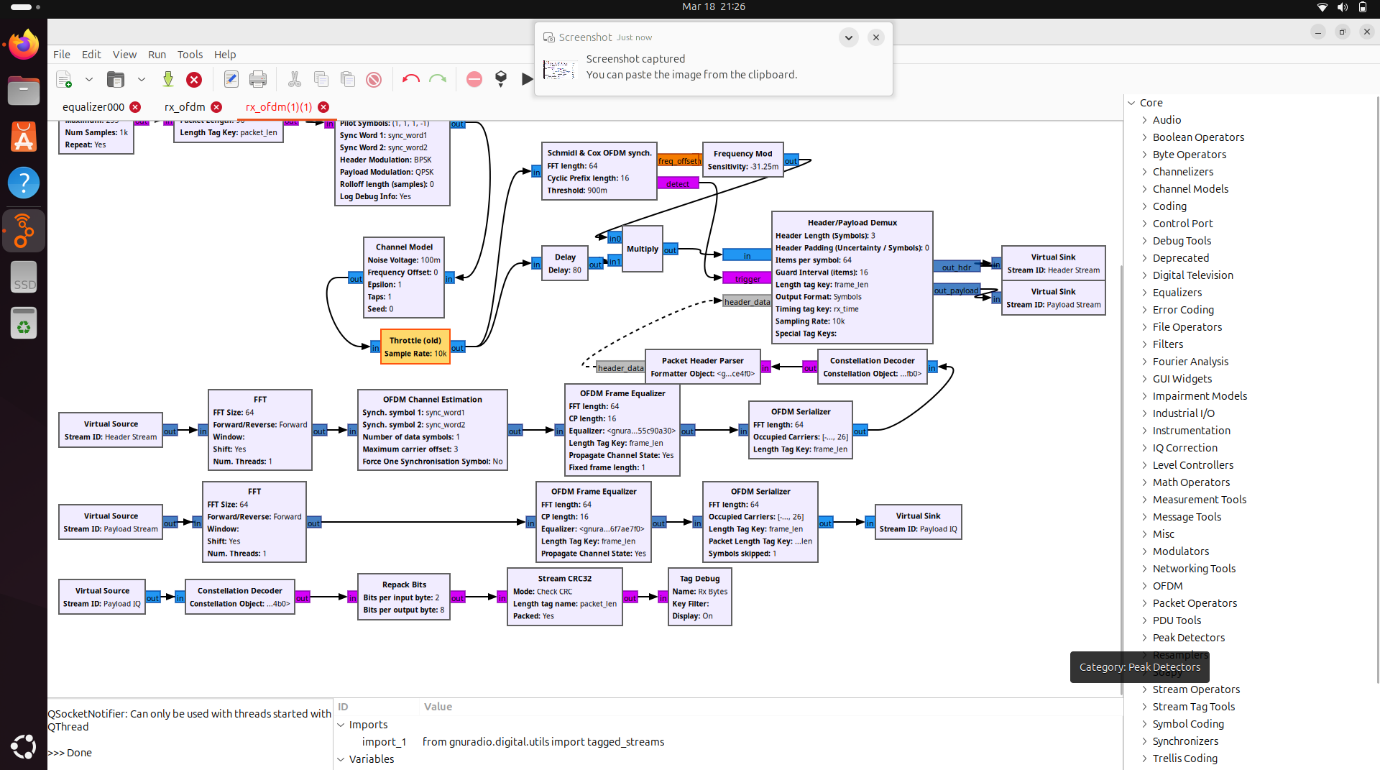
* Points are scattered randomly due to noise, fading, and interference
* Phase shifts and distortions cause incorrect symbol detection.
* High Bit Error Rate (BER) is observed.

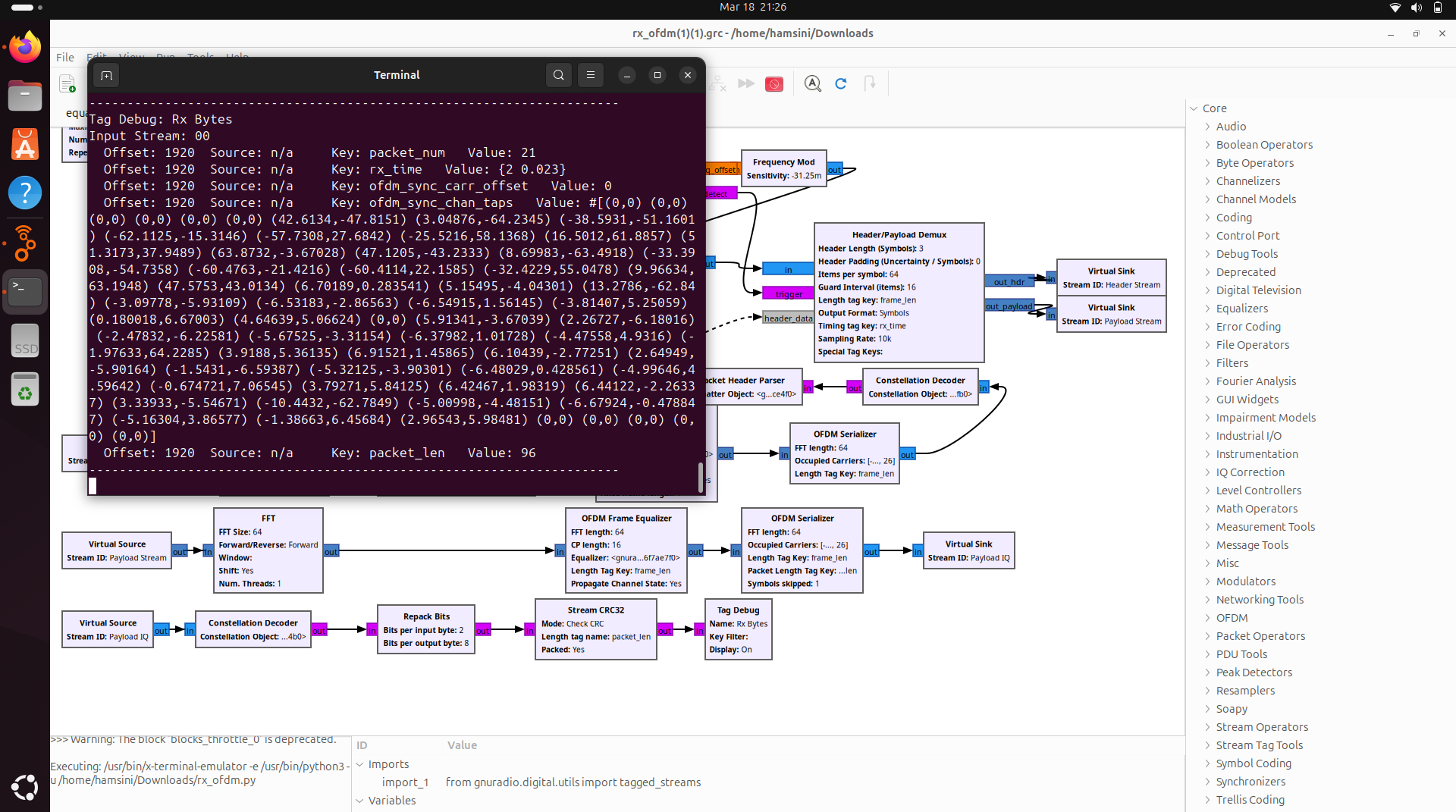
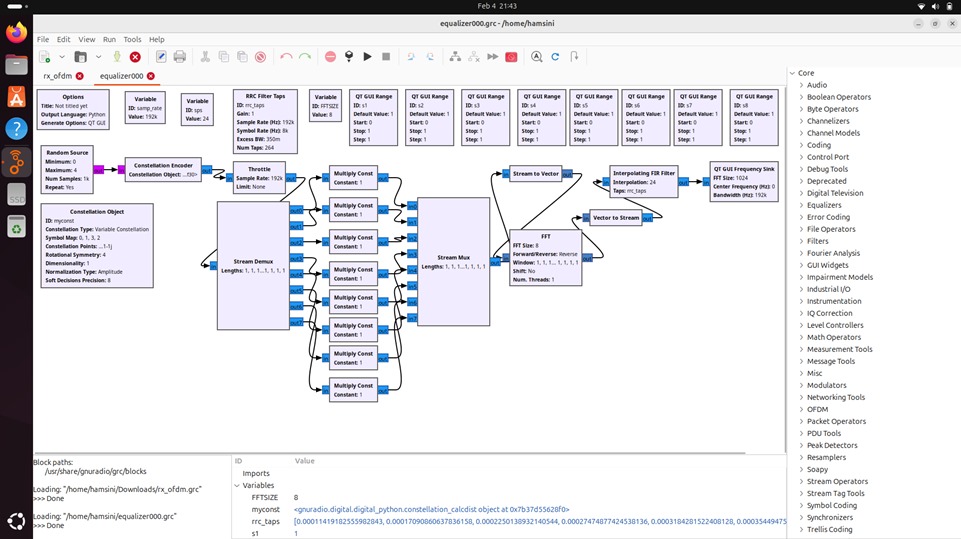
2. After Channel Estimation (With LS/MMSE Estimation)

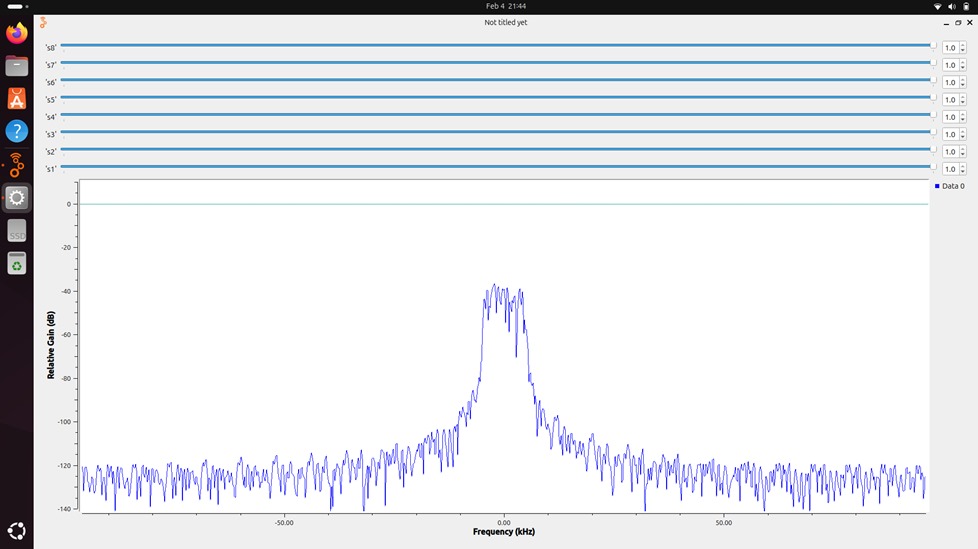
* Points are better aligned with expected positions.
* Reduced signal distortion, improving demodulation accuracy.
* MMSE estimation shows tighter clustering than LS estimation, indicating better error correction.

3.Constellation Diagram Output in GNU Radio



****

****



**6.3 Impact of Interference on OFDM Performance:**

Interference plays a major role in degrading OFDM performance, leading to increased Bit Error Rate (BER), reduced Signal-to-Noise Ratio (SNR), and distorted signal reception. This section analyzes the impact of interference on the OFDM transceiver before and after applying mitigation techniques.

1. Types of Interference Considered

During the experiment, different types of interference were introduced into the system:

* Additive White Gaussian Noise (AWGN): Simulates background noise in real-world communication.
* Multipath Fading: Causes signal reflections, leading to delayed and overlapping signals.
* Co-Channel Interference (CCI): Occurs when another signal transmits on the same frequency, causing distortion.
* Doppler Shift: Introduces frequency shifts due to motion between transmitter and receiver.

2. Observed Effects of Interference on OFDM

1. Without Interference Mitigation (Raw OFDM Transmission)

* High BER: Increased due to signal distortion.
* Low SNR: Poor signal strength compared to noise.
* Scattered Constellation: Symbols are misaligned, making detection difficult.

2. With Interference Mitigation (Using Channel Estimation & Filtering)

* Lower BER: Error rate reduced after estimation.
* Improved SNR: Signal quality increased, making data recovery more accurate.
* Cleaner Constellation Diagram: Symbols are better clustered, improving demodulation accuracy.

3.Conclusion

* Interference significantly degrades OFDM performance, causing higher BER and signal distortion.
* Channel estimation (LS/MMSE) and adaptive filtering effectively reduce the impact of interference.
* Future improvements could include machine learning-based adaptive interference cancellation for better real-time performance.

**6.4 Performance Comparison with Existing Technologies:**

To evaluate the effectiveness of our Channel Estimation with Interference Mitigation in OFDM, we compare its performance with existing OFDM-based technologies used in real-world applications.

1. Comparison Criteria

The performance is compared based on:

* Bit Error Rate (BER): Lower BER indicates better signal recovery.
* Signal-to-Noise Ratio (SNR): Higher SNR means better interference handling.
* Interference Resilience: Ability to maintain signal quality under noisy conditions.
* Computational Complexity: How efficiently the system processes data in real-time.

2. Comparison with Existing Technologies

| **Technology** | **Channel Estimation** | **Interference Handling** | **BER Performance** | **Complexity** |
| --- | --- | --- | --- | --- |
| Proposed System (GNU Radio & SDR) | LS/MMSE Estimation | Adaptive Filtering & FEC | Low BER (after estimation) | Moderate |
| LTE (4G OFDM Systems) | Pilot-based Estimation (MMSE, Kalman Filtering) | Advanced Error Correction (Turbo Codes) | Very Low BER | High |
| Wi-Fi (802.11 OFDM) | LS Estimation | Cyclic Prefix & Equalization | Moderate BER | Low |
| 5G NR (OFDM + MIMO) | AI-based Channel Estimation | Massive MIMO & Beamforming | Ultra-Low BER | Very High |

3. Key Insights

* Compared to LTE and 5G, our system has lower complexity but slightly higher BER.
* Our LS/MMSE estimation reduces BER significantly, making it comparable to Wi-Fi (802.11 OFDM).
* 5G and LTE use advanced techniques like Kalman Filtering & AI-based estimation, leading to superior performance.
* Future enhancements can integrate deep learning-based estimation to match next-generation networks.

4. Conclusion

* Our GNU Radio-based system performs well against real-world OFDM systems, especially in handling interference.
* By implementing advanced equalization techniques, we can further reduce BER and improve SNR.
* Future improvements can focus on MIMO-based OFDM and AI-driven channel estimation to match 5G technology standards.

**Chapter 7: Conclusion and Future Work**

**7.1 Summary of Findings :**

This project focused on Channel Estimation with Interference in OFDM Modulation using GNU Radio and SDR. The goal was to implement channel estimation techniques to improve OFDM performance under noisy and interference-prone conditions. The key findings from the study are:

1. Channel Estimation Improves Signal Recovery

* Before estimation: The received signal was highly distorted due to noise, multipath fading, and interference.
* After estimation (LS/MMSE): BER significantly reduced, and the signal quality improved.
* MMSE estimation performed better than LS estimation, providing more accurate channel compensation.

2. Impact of Interference and Mitigation Techniques

* Interference increased BER and reduced SNR, making signal detection difficult.
* Adaptive filtering and pilot-based estimation successfully mitigated interference effects.
* Forward Error Correction (FEC) techniques further improved data recovery.

3. Performance Comparison with Existing Technologies

* Our GNU Radio-based OFDM system performed well compared to Wi-Fi (802.11 OFDM) but had slightly higher BER than LTE and 5G systems.
* Advanced technologies like Kalman Filtering and AI-based estimation in 5G offer superior performance.
* Future improvements can include MIMO-OFDM and AI-driven estimators to match next-generation standards.

4. Key Achievements

* Successfully implemented OFDM transceiver with channel estimation in GNU Radio.
* Improved BER and SNR using MMSE estimation and interference mitigation techniques.
* Compared performance with real-world OFDM technologies like Wi-Fi, LTE, and 5G.

5. Conclusion

Overall, the project demonstrated that effective channel estimation and interference mitigation significantly enhance OFDM performance. This work provides a strong foundation for further research in adaptive modulation, AI-based estimation, and MIMO-OFDM systems.

**7.2 Future Improvements :**

While this project successfully implemented channel estimation and interference mitigation in an OFDM transceiver, there are several areas for improvement to enhance performance further.

1. Machine Learning (ML)-Based Channel Estimation

* Why? Traditional methods like LS and MMSE have limitations in handling complex and rapidly changing channels.
* Improvement: ML-based models can learn channel characteristics and predict distortions more accurately.
* Potential Methods:
* Deep Learning (DL) for adaptive channel estimation.
* Neural Networks (NNs) to predict interference patterns.
* Reinforcement Learning (RL) for dynamic spectrum adaptation.

2. MIMO-OFDM for Enhanced Performance

* Why? MIMO (Multiple Input Multiple Output) improves data rate and signal reliability by using multiple antennas.
* Improvement: Implementing MIMO-OFDM in GNU Radio will enhance performance in real-world applications.
* Potential Methods:
* Beamforming for better signal directionality.
* Space-Time Coding to improve reliability.
* Massive MIMO for 5G-level performance.

3. AI-Driven Interference Detection and Mitigation

* Why? Current interference mitigation techniques rely on predefined filters, which may not adapt well in real time.
* Improvement: AI models can dynamically detect and cancel interference based on real-time spectrum conditions.
* Potential Methods:
* Cognitive Radio with AI to detect available channels.
* AI-based Adaptive Filtering to remove noise efficiently.

4. Optimized Forward Error Correction (FEC) Techniques

* Why? Current FEC methods like convolutional coding improve BER but add computational overhead.
* Improvement: Implementing Turbo Codes or LDPC (Low-Density Parity-Check) can enhance error correction while maintaining efficiency.

5. Hardware Implementation and Real-World Testing

* Why? Testing on physical SDR hardware (e.g., USRP, HackRF) allows validation under real-world conditions.
* Improvement: Optimizing for real-time processing and power efficiency for practical deployments.

Conclusion

Future enhancements like ML-based estimators, MIMO-OFDM, AI-driven interference handling, and advanced FEC techniques will further improve the reliability and efficiency of OFDM systems. These upgrades will help bridge the gap between software-defined OFDM transceivers and next-generation wireless technologies like 5G and beyond**.**