UNIVERSITY of RWANDA

***Research and Postgraduate Studies (RPGS) Unit***

# Thesis tittle: Evaluating the Impact of Low-resolution DACs on a Downlink Massive MIMO System Performance


**By:**

**Christine Uwamahoro**

**Reg number: 218002132**

College of Science and Technology

School of Information and Communication Technology (SoICT)

Master of Science in Operational Communication.


A dissertation submitted in partial fulfilment of the requirements for the degree of Master of Science in Operational Communication.

July 2025

# DECLARATION

I, Christine Uwamahoro, hereby declare that this project, named "Evaluating the impact of low-resolution DACs on downlink massive MIMO system performance," is my original work and has never been submitted to any school. This was implemented from my effort and is submitted in fulfillment of the requirement for the award of Master of Science in Operational Communication.

Christine Uwamahoro

Reg nº:218002132

Signed: ………………………….. Date: 10/July/2025

# Certificate

This is to certify that the project work entitled ""Evaluating the impact of low-resolution DACs on downlink massive MIMO system performance" is a record of original work done by Christine Uwamahoro with Reg no: 218002132 in partial fulfillment of the requirement for the award masters of science in Operational Communication of College of Science and Technology, University of Rwanda during the academic year 2024-2025.

Signature………………………………. Date 10/July/2025

Student Name: Christine Uwamahoro

**Supervisor**

Names: Dr.Christine MUSANASE

Signature:…………………………………….

**Head of Department**

Names:…………………………… Signature:………………………….

# Acknowledgements

I would like to express our sincere gratitude to my supervisors, Dr Innocent KABANDANA and Dr. Christine MUSANASE, for their invaluable guidance and support throughout this thesis.

# Abstract

This project investigates the impact of low-resolution digital-to-analog converters (DACs) on the performance of downlink massive multiple-input multiple-output (MIMO) systems, emphasizing the trade-offs between power efficiency and system performance, with a specific focus on a Downlink Multi-User Multiple-Input Single-Output (MU-MISO) configuration. DACs are critical components in modern wireless systems, such as millimeter-wave systems, and are significant contributors to power consumption. Reducing the resolution of DACs provides a sustainable approach to lowering power consumption and costs. However, this introduces quantization noise, which degrades system performance metrics such as bit error rate (BER), spectral efficiency (SE), and throughput. A MATLAB-based system-level simulator was developed to model the behavior of massive MU-MISO systems under varying DAC resolution bits. Numerical results demonstrate that 1-bit DAC systems exhibit significantly higher BER due to quantization error. This error limits the system's performance, especially in higher-order modulation schemes. However, advanced techniques like quantization and precoding schemes effectively mitigate these effects, ensuring improved performance for MU-MISO systems under low-resolution DAC constraints.

# Contents

# List of Figures

# List of Tables

# List of Acronyms and Abbreviations

ADC: Analog to Digital converter

AQNM: Additive Quantization Noise Model

AWGN: Additive White Gaussian Noise

BER: Bit Error Rate

BLER: Block Error Rate

BPSK: Binary Phase Shift Keying

BS: Base Station

CSI: Channel State Information

DAC: Digital to Analog converter

DSP: Digital Signal Processing

LS: Least Square

MIMO: Multiple Input Multiple Output

MISO: Multiple Input Single Output

MU-MISO: Multi-user-MISO

MU-MIMO: Multi-user-MIMO

OFDM: Orthogonal Frequency Division Multiplexing

QAM: Quadrature Amplitude Modulation

QPSK: Quadrature Phase Shift Keying

RF: Radio Frequency

SNR: Signal to Noise Ratio

SINR: Signal to Interference and Noise Ratio

UE: User Equipment

ZF: Zero Forcing

# Chapter 1 General Introduction

## 1.1 Introduction

In modern wireless communication systems, the need for high data rates, low latency, and energy efficiency has driven the adoption of advanced technologies such as massive multiple-input multiple-output (MIMO) systems. These systems, which employ a large number of antennas at the base station to serve multiple users simultaneously, promise significant improvements in spectral and energy efficiency [1]. However, their practical implementation introduces new design challenges, especially related to hardware complexity and power consumption. One of the critical hardware components in any communication system is the digital-to-analog converter (DAC), which plays a key role in signal transmission. DACs are used to convert digital signals processed in the baseband into analog signals that can be transmitted over the air. In massive MIMO systems, where each antenna requires a dedicated DAC, the cumulative power consumption and cost of high-resolution DACs become significant. High-resolution DACs (typically 8 bits or above) provide accurate signal conversion but require more power, thereby increasing system cost and reducing scalability[2].

To address these concerns, researchers have focused on low-resolution DACs (1–4 bits), which consume less power and offer a simplified hardware design. While this approach reduces system energy requirements, it introduces considerable quantization noise into the transmitted signal. This noise can degrade key performance metrics such as Bit Error Rate (BER), Spectral Efficiency (SE), and overall system throughput, which are vital for maintaining communication quality and reliability[3]. The trade-off between energy efficiency and system performance has become a key research focus in recent years. In particular, downlink Multi-User MISO (Multiple Input Single Output) systems present an interesting scenario where the base station transmits to multiple users with single antennas. Understanding how low-resolution DACs affect such systems provides critical insights for the practical deployment of power-efficient 5G and future 6G networks.

In response to this challenge, several techniques have been proposed to mitigate the effects of low-resolution quantization. Among these, precoding methods such as Zero-Forcing (ZF) and Maximal Ratio Transmission (MRT) precoding have shown promise in reducing inter-user interference and enhancing system robustness[4]. Additionally, the use of optimized quantization schemes beyond uniform quantization can help preserve signal quality even at lower DAC resolutions. This project builds on this body of research by developing a system-level MATLAB simulator to model the behavior of downlink MU-MISO systems under various DAC resolution settings. The simulator will provide a platform to analyze how different quantization levels affect performance and to test techniques for mitigating quantization noise. The findings will contribute toward designing more sustainable, high-performance wireless systems that balance power efficiency and signal integrity, making them suitable for future high-density deployments.

## 1.2 Motivation

With the rapid expansion of wireless technologies, including 5G and beyond, there is a growing need for systems that are not only high-performing but also energy-efficient and cost-effective. Massive MIMO is a key enabler of this evolution, offering substantial gains in throughput and reliability. However, deploying hundreds of antennas in base stations drastically increases the power and cost burden, with DACs being one of the primary contributors.

Reducing DAC resolution emerges as a promising solution to these challenges, enabling lighter, cheaper, and more sustainable infrastructure. However, this comes at the cost of signal distortion due to quantization. Understanding and solving this trade-off is essential for the future of scalable wireless networks. This project is motivated by the need to bridge this gap, optimizing system design to support both energy efficiency and robust communication performance, contributing to greener and more accessible communication technologies.

## 1.3 Problem Statement

Massive multiple-input multiple-output (MIMO) systems are essential for achieving high data throughput and spectral efficiency in modern cellular networks. However, implementing large-scale antenna arrays presents practical challenges such as power consumption, high cost, and so on. Low-resolution DACs offer a potential solution to reduce power consumption and hardware complexity, but may introduce quantization noise and degrade signal quality. This trade-off presents a challenge: how to maintain acceptable system performance while leveraging the power and cost benefits of low-resolution DACs.

This project addresses this challenge by investigating the performance of downlink Multi-User MISO systems with low-resolution DACs and developing techniques such as advanced precoding and optimized quantization schemes to mitigate the degradation caused by quantization noise. The goal is to provide evidence-based strategies for designing efficient, low-power massive MIMO systems without severely compromising communication quality.

Finally, the results will answer the following question: How can the system performance (increasing spectral efficiency or decreasing bit error rate) be improved in cellular networks with low-resolution DACs?

## 1.4 Study Objectives

### 1.4.1 General Objectives

This project aims to investigate and improve the performance of downlink massive MIMO systems that utilize low-resolution digital-to-analog converters

(DACs). Specifically, the project aims to understand how low-resolution DACs impact key system performance metrics, such as bit error rate (BER) and throughput, and explore solutions to mitigate these effects.

### 1.4.2 Specific Objectives

The main objectives are:

1. Evaluate System Performance with Low-Resolution DACs: Quantify the impact of DAC resolution on key metrics such as bit error rate (BER) and throughput across different resolutions (2-bit to 4-bit DACs).

2. Improving system performance by developing precoding Techniques: Design precoding algorithms that reduce quantization noise and improve system performance in terms of BER and throughput.

3. Build a System-Level Simulator: Create a MATLAB-based system-level simulator that will model the behaviour of downlink massive MIMO systems with low-resolution DACs, providing a foundation for analysing the effectiveness of the proposed techniques.

## 1.5 Hypotheses

This project hypothesizes that although low-resolution digital-to-analog converters (DACs) inherently introduce quantization noise and degrade the performance of downlink massive MIMO systems, manifested through increased bit error (BER) and reduced throughput, these negative effects can be significantly mitigated. Specifically, by implementing advanced precoding techniques and selecting optimal quantization schemes, it is possible to improve system performance even when using low-resolution DACs.

## 1.6 Study scope

This study is limited to the evaluation of downlink Multi-User MISO (Multiple Input Single Output) massive MIMO systems using low-resolution DACs. The project focuses on:

- Simulating system-level performance using MATLAB across various DAC resolutions (e.g., 2-bit to 4-bit).

- Measuring key performance metrics such as Bit Error Rate (BER), Spectral Efficiency (SE), and throughput.

- Investigating different precoding techniques used to reduce quantization noise.

- Comparing the effectiveness of several quantization schemes in mitigating performance loss.
- Ensuring simulation plausibility by aligning results with theoretical models and findings from existing literature.

The study does not extend to hardware implementation or uplink scenarios and will not include advanced MIMO configurations such as full-duplex or OFDM systems due to time limitations.

## 1.7 Significance of the study

While high-resolution DACs offer excellent performance, they come at a cost of increased power consumption and complexity. In modern MISO systems:

- Power Efficiency: Lower-resolution DACs consume significantly less power. In systems with hundreds of antennas, power-saving at each DAC accumulates into substantial overall savings.
- Cost and Size: Higher-resolution DACs are expensive and bulky. Low-resolution DACs enable cost-effective and compact hardware designs.

5G networks and future systems aim for ultra-reliable, low-latency communication while minimizing environmental impact. Low-resolution DACs strike a balance between energy efficiency and system performance.

## 1.8 Organisation of the Study

Chapter 1 introduces the problem and provides background for studying low-resolution DACs in massive MIMO systems. Chapter 2 provides the literature review on downlink massive MIMO systems and DACs, including related work on the topic. It follows Chapter 3, which explains the method used in implementing this study. Chapter 4 provides a detailed description of the simulation system, including the implementation of different modulation schemes, the architecture of the DAC, various quantization schemes, MRT and ZF precoding techniques, and LS channel estimation. Chapter 5 contains a step-by-step analysis of simulated results, focusing on system performance evaluation. Chapter 5 gives the conclusion and recommendations.

# Chapter 2 Literature Review

Several studies have examined the effects of low-resolution DACs on massive MIMO systems, focusing on performance degradation and methods to mitigate it. This section reviews key contributions related to DAC performance, quantization schemes, and precoding techniques.

Tan et al. [4] investigate multiuser precoding techniques for massive MIMO systems, focusing on designing efficient schemes to enhance system performance and analyzing achievable rates. Massive MIMO technology has gained traction due to its potential for dramatically increasing spectral and energy efficiency.

Jacobsson et al. [5] analyzed the impact of low-resolution DACs in massive MU-MIMO systems, focusing on linear precoding techniques such as Zero Forcing (ZF). Using Bussgang's theorem, they demonstrated that systems with 3-4-bit DACs can achieve near-optimal performance in terms of Bit Error Rate (BER) and sum-rate throughput at high Signal-to-Noise Ratio (SNR). However, the study primarily focused on OFDM-based systems.

Liu et al. [6] introduced the Additive Quantization Noise Model (AQNM) to assess the impact of quantization noise resulting from low-resolution DACs and ADCs. Their findings revealed that 2-3-bit DACs incur minimal spectral efficiency (SE) loss of 10–20 percent under high SNR conditions. However, their model is widely applicable; it does not address system-level simulations or precoding techniques.

Sharma et al. [7] evaluate the performance of several digital modulation schemes under an Additive White Gaussian Noise (AWGN) channel, focusing on their Bit Error Rate (BER) characteristics. The study aims to identify the most efficient modulation techniques for digital communication systems by analyzing BER across different SNR levels.

Previous works have also highlighted the need for comprehensive evaluations of quantization schemes such as uniform quantization, A-law, and $\mu$-law companding, as well as a detailed analysis of system-level performance metrics like BER, SNR, and Spectral Efficiency (SE) under practical simulation scenarios.

This study addresses these gaps by developing a MATLAB-based system-level simulator to analyze the effects of varying DAC resolutions on massive MIMO systems. The study incorporates various quantization schemes and evaluates precoding techniques to improve system performance.

# Chapter 3 Research Methodology

## 3.1 Simulation Setup

### 3.1.1 Evolution of the Simulation Setup

The simulation setup has evolved progressively to accommodate increasing system complexity, transitioning from a simple SISO system to a more advanced MU-MISO system. This evolution enables a step-by-step understanding of system behavior and performance under varying conditions.

1. SISO Setup: The initial simulation setup involved a SISO configuration, where a single transmit antenna at the base station communicated with a single user terminal equipped with one receive antenna. This simple setup served as the baseline to evaluate fundamental system performance metrics such as BER and provided insights into quantization effects under ideal conditions.

2. MU-MISO Setup: The simulation extends the system to a MU-MISO configuration. In this setup, the base station employs $M$ transmit antennas to serve $K$ users simultaneously, where each user terminal is equipped with a single receive antenna. This setup introduces the challenge of multi-user interference (MUI), which is mitigated using advanced precoding techniques like ZF and MRT. The MU-MISO setup reflects realistic scenarios in modern wireless communication systems and forms the focus of this study.

The evolution from SISO to MU-MISO highlights the increasing complexity and realism of the simulation setup, providing a foundation for evaluating the impact of low-resolution converters on system performance.

### 3.1.2 Objective of the Simulation

The objective of this simulation is to investigate the impact of low-resolution Digital-to-Analog Converters and Analog-to-Digital Converters on the downlink performance of a multi-user MISO system. This study aims to analyze how the quantization resolution and the choice of quantization scheme (Uniform, A-law, and $\mu$-law) affect key system performance metrics such as Bit Error Rate and Spectral Efficiency under varying Signal-to-Noise Ratio conditions.

The motivation for this analysis lies in the trade-off between hardware complexity, energy consumption, and communication performance. Reducing the resolution of Digital-to-Analog Converters and Analog-to-Digital Converters can significantly reduce power consumption and hardware costs in large-scale multiantenna systems, such as those used in 5G and future wireless networks. However, this reduction in resolution introduces quantization errors that can degrade system performance.

To understand these trade-offs, multiple simulation scenarios are explored. Each scenario combines different quantization schemes (Uniform, A-law, and $\mu$law), Digital-to-Analog Converters and Analog-to-Digital Converters resolutions (2, 4, and 8 bits), and precoding techniques (Zero-Forcing and Maximal Ratio Transmission). By evaluating system performance across these scenarios, this study identifies optimal configurations for efficient and high-performance MUMISO systems with low-resolution converters.

### 3.1.3 Key Parameters

The key parameters used in the simulation are listed in Table 1. These parameters define the environment and constraints of the simulation and serve as a basis for analyzing the impact of quantization schemes, modulation schemes, and precoding techniques.
The simulation environment is characterized by the following key aspects:

**SNR Range:** The SNR is varied from -20 dB to 50 dB to analyze the system's robustness under low, medium, and high SNR conditions.
**DAC/ADC Resolutions:** The system performance is evaluated for DAC/ADC resolutions of 2, 4, and 8 bits. These resolutions represent low, medium, and high-resolution cases, respectively, allowing for a comparative analysis of the performance degradation caused by low-resolution converters.
**Quantization Schemes**: Three quantization methods are considered: Uniform, A-law, and $\mu$-law. The choice of these schemes is motivated by their prevalence in real-world communication systems and their distinct characteristics in handling small and large amplitude signals.
**Number of Base Station Antennas:** The system models a base station with Table 1: Simulation Parameters of the overall simulation setup 64 antennas. This configuration is common in large-scale MIMO systems where massive antenna arrays are employed to achieve spatial multiplexing and enhance system capacity.
**User Equipment (UE):** The system considers single-antenna User Equipment (UE), following a standard downlink multi-user MISO setup.
**Channel Model:** The channel is combined with AWGN noise. This model reflects real-world scenarios where the transmitted signal is subject to multipath propagation and noise.

**Precoders:** Two precoding techniques are used: Zero-Forcing and Maximal Ratio Transmission. Zero-forcing eliminates inter-user interference, while Maximal Ratio Transmission aligns the transmitted signal to maximize the signal-to-noise ratio at the receiver.
**Monte Carlo Iterations:** Multiple iterations are performed for each SNR point to ensure statistical accuracy. This approach [8] enables a more robust analysis of system performance, reducing the variability introduced by random channel realizations.
**Performance Metrics:** The system's performance is evaluated using BER and SE. BER quantifies the error rate of transmitted bits, while SE measures how efficiently the available bandwidth is utilized for data transmission.

*Table 1: Key parameters used in simulation*

| **Parameter** | **Value / Description** |
|---|---|
| SNR Range | From -20 dB to 50 dB |
| DAC/ADC Resolutions | 2, 4, and 8 bits |
| Quantization Schemes | Uniform, A-law, and $\mu$-law |
| Modulation Types | BPSK, QPSK, 16-QAM, and 64-QAM |
| Number of Base Station Antennas | 64 antennas |
| User Equipment (UE) | Single-antenna UEs |
| Channel Model | AWGN |
| Precoders | Zero-Forcing and Maximal Ratio Transmission |
| Monte Carlo Iterations | Multiple iterations for statistical accuracy |
| Performance Metrics | Bit Error Rate (BER) and Spectral Efficiency (SE) |

### 3.1.4 Simulation Process

The simulation process consists of a sequence of operations that convert binary data into modulated symbols, transmit them over a simulated wireless channel, and analyze the system's performance based on BER and SE. The key steps in this process are as follows:

1. Source Signal Generation: Random binary bits are generated as input to the system. The bits are grouped based on the modulation scheme, which in this study is fixed to 64-QAM.

2. Modulation and Normalization: The binary bits are mapped to complex-valued symbols using 64-QAM modulation. The modulated output is then normalized to ensure consistent power across all symbols before further processing.

3. Channel Estimation and Precoding: The channel matrix $H$ is estimated or assumed based on the simulation setup. Accurate CSI is critical for precoding methods such as ZF or MRT to mitigate multi-user interference. Practical channel estimation often relies on techniques like LS or Minimum Mean Square Error (MMSE) estimation [7, 8]. These methods minimize the estimation error and provide robust performance under realistic conditions.

4. DAC: The precoded digital signal is converted into an analog signal using one of the following quantization schemes:

    - Uniform quantization: Divides the signal range into equally spaced levels.
    - A-law quantization: Logarithmic compression improves the resolution of small amplitudes, enhancing performance in low-SNR conditions.
    - $\mu$-law quantization: Similar to A-law, optimized for North American systems.

    The DAC resolution is set to 2, 4, or 8 bits to analyze the impact of quantization on the system's performance.

5. Channel Model: The transmitted analog signal passes through a simulated wireless channel with AWGN. This channel introduces noise to the transmitted signal.

6. ADC: The received analog signal is converted back to a digital signal using the same quantization schemes as the DAC: Uniform, A-law, or $\mu$-law. The ADC resolution is also set to 2, 4, or 8 bits for consistency.

7. Demodulation: The demodulator extracts the binary data from the received symbols. Since 64-QAM is used, the demodulator maps the received signals back to the original binary bit stream.

8. Performance Evaluation: The system's performance is evaluated using two key metrics:

   - BER: The number of bit errors divided by the total number of transmitted bits.
   - SE: The amount of information transmitted over a given bandwidth, calculated as the number of bits per symbol times the bandwidth efficiency.

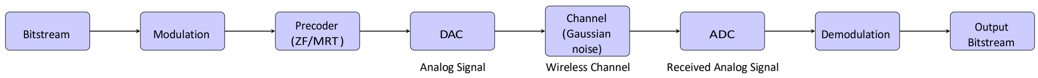


*Figure 1: Simulation process flow from source signal generation to performance evaluation.*

The process includes bitstream generation, modulation, precoding, DAC, channel effects (AWGN), ADC, and demodulation to recover the original bitstream.

# Chapter 4 System Analysis and Design

## 4.1 MISO Systems

### 4.1.1 Overview of MISO

MISO (Multiple Input Single Output) is a wireless communication system where a transmitter uses multiple antennas to transmit data to a receiver with a single antenna[9]. The use of multiple antennas at the transmitter provides spatial diversity and improves signal robustness, particularly in fading environments. Types of Systems Based on Number of Users:

1. Single-User SISO (SU-SISO):

SU-SISO represents a basic communication system where a single transmitting antenna at the base station communicates with a single receiving antenna at the user equipment (UE). This setup serves as the foundation for understanding the fundamental behaviors of modulation, quantization, and noise effects without the added complexities of multiple antennas or users[10].

2. Multi-User MISO (MU-MISO):

MU-MISO refers to a wireless communication setup where a base station equipped with multiple transmit antennas communicates simultaneously with multiple users, each having a single receive antenna[11]. This configuration leverages spatial diversity and beamforming techniques to enhance system capacity and performance while managing inter-user interference.

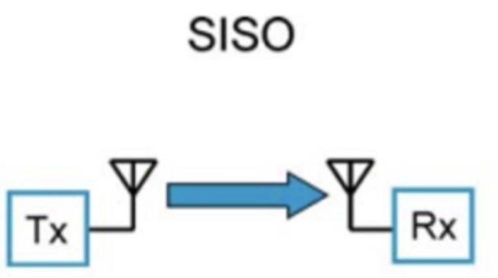


*Figure 2: SISO system [12].*

### 4.1.2 Overview of Single-user SISO

The SU-SISO (Single User Single Input Single Output) configuration represents the simplest form of wireless communication, where a single transmitter at the base station communicates with a single receiver at the user equipment (UE).

In Figure 2, the base station, equipped with one transmitting antenna (Tx), sends a single data stream through a wireless channel to the receiver (Rx), which is also equipped with a single antenna. The base station acts as the source of communication, performing critical functions such as data encoding, modulation, and amplification to prepare the signal for transmission. The transmitting antenna (Tx) is responsible for radiating the electromagnetic signals into the propagation channel, ensuring efficient energy transfer in the direction of the receiver. The propagation channel, which serves as the medium for signal transmission, introduces effects like noise, interference, and attenuation, mimicking real-world conditions and influencing the signal's quality[10]. On the receiving side (Rx), the antenna captures the transmitted electromagnetic waves and converts them back into electrical signals, which are then processed by the receiver. The receiver demodulates and decodes the signal, reconstructing the original data for the end-user. The simplicity of this configuration ensures a clear understanding of how quantization, noise, and channel effects influence wireless communication, enabling effective system optimization[13].

### 4.1.3 Overview of Multi-User MISO

Multi-User Multiple Input Single Output (MU-MISO) is a configuration in wireless communication where a base station with multiple antennas transmits data to multiple user devices, each equipped with a single antenna. This setup is a subset of MIMO (Multiple Input Multiple Output) technology, which is widely used in modern wireless systems like 5G networks. MU-MISO exploits spatial multiplexing and beamforming to improve system capacity and spectral efficiency. Unlike conventional single-user systems, MU-MISO enhances data transmission by simultaneously serving multiple users. It leverages techniques such as precoding to mitigate interference between users, ensuring each user device receives an intended signal with minimal interference[12].

Figure 3 illustrates the working of a Multi-User Multiple-Input Single-Output (MU-MISO) system, where a Base Station (BS) equipped with multiple antennas simultaneously communicates with multiple User Equipments (UEs), each having a single antenna. The BS employs beamforming techniques to focus its transmission power in specific directions, ensuring that each UE receives its intended signal with minimal interference. By utilizing advanced precoding algorithms, the BS can spatially separate the data streams transmitted to different UEs, enabling concurrent communication[14]. This approach improves spectral efficiency by serving multiple users within the same time and frequency resources, while also enhancing signal quality through directional transmission. The curved beams in the figure represent the focused signals directed towards the UEs, showcasing how MU-MISO efficiently manages multi-user communication in wireless networks.

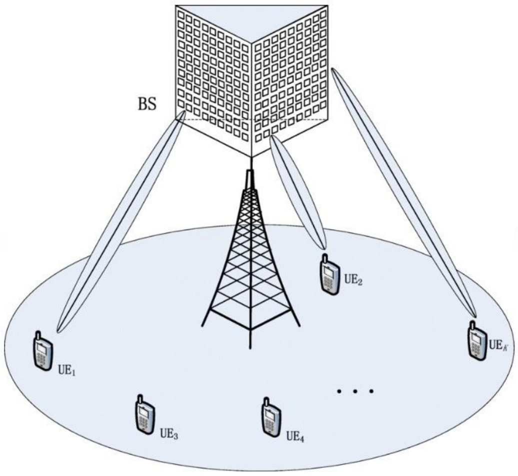


*Figure 3: MU-MISO system [4].*

**Overview of Beamforming:** Beamforming is a critical technique in modern wireless communication systems, particularly in MIMO and MISO configurations, to enhance signal quality and improve spectral efficiency. It involves directing the transmission or reception of signals in specific directions, rather than uniformly in all directions, to achieve better system performance. Beamforming utilizes the spatial diversity of multiple antennas at the transmitter and/or receiver to constructively combine signals at desired locations while suppressing interference at others. This is achieved by adjusting the phase and amplitude of the signals emitted from each antenna to create a focused beam pattern[15].

**Key Components and Techniques:**

- Base Station (BS): Equipped with multiple antennas, the BS transmits distinct signals to multiple users in the same time-frequency resource.
- User Equipment (UE): Each user has a single antenna, making it cost-effective and power-efficient for mobile devices.
- Beamforming: Beamforming directs signals toward specific users, optimizing signal strength and reducing interference.
- Precoding: Precoding techniques like Zero Forcing (ZF) and Maximum Ratio Transmission (MRT) optimize the transmitted signal to improve signal quality and minimize interference.

## 4.2 Digital to Analog Converters (DACs)

DACs are essential components in modern electronic systems, particularly in applications requiring the conversion of digital signals into analog form. A DAC takes a binary input, typically from a microcontroller or a digital signal processor (DSP), and converts it into a corresponding analog output, such as voltage or current. This conversion process is critical in many fields, including audio processing, telecommunications, control systems, and instrumentation.

### 4.2.1 Working Principle

In massive MIMO systems, Digital-to-Analog Converters (DACs) are crucial in converting digital baseband signals into analog signals for radio frequency (RF) transmission. The resolution of the DAC determines the accuracy of this conversion, impacting system performance metrics such as Bit Error Rate (BER) and Spectral Efficiency (SE). DAC maps digital input values (binary representation) to corresponding analog output levels for suitable transmission. It generates stepped waveforms and uses low-pass filters to produce smooth signals for modulation and transmission over radio frequencies[8]. For an N-bit DAC, the analog output voltage $V_{\text{out}}$ is given by:

$$Vout = Vref\,\frac{D}{2}\text{N}$$

where:

- $V_{\text{ref}}$ is the reference voltage,
- $D$ is the digital input code,
- $N$ is the resolution (number of bits).

Conversely, ADCs transform received analog waveforms into digital signals for processing. They sample analog inputs at regular intervals (following the Nyquist theorem), quantize the amplitudes into discrete levels, and encode them as binary values for further demodulation and decoding. Key parameters like sampling rate, resolution, linearity, and speed are critical for ensuring signal fidelity and efficient processing.

This project evaluates the impact of low-resolution DACs on system performance, specifically:

- Bit Error Rate (BER): The error rate between transmitted and received bits,
- Spectral Efficiency (SE): The efficiency of data transmission over the available bandwidth.

A MATLAB-based system-level simulator is used to analyze these metrics under different DAC resolutions and quantization schemes, including uniform quantization, A-law, and $\mu$-law companding.

### 4.2.2 Modeling DACs with Quantization

Quantization is the process of mapping a continuous range of values into a finite set of discrete levels. It is a fundamental step in Digital-to-Analog Conversion (DAC), where the input digital signal is approximated to the nearest discrete level, introducing quantization noise. Quantization plays a crucial role in shaping the performance of low-resolution DACs. Both uniform and nonuniform quantization techniques are essential for evaluating how different DAC designs impact the overall system performance, particularly in massive MIMO and MU-MISO systems.

Uniform Quantization: Uniform quantization divides the input signal range into evenly spaced intervals, and each interval is represented by a unique discrete level. This approach is simple to implement and is widely used in systems where the input signal has a uniform probability distribution. In our study, uniform quantization serves as a baseline model for analyzing the effects of quantization noise on system performance metrics like Bit Error Rate (BER) and Spectral Efficiency (SE)[16]. It provides insight into the limitations of low-resolution DACs under simple, evenly spaced quantization.

Non-uniform quantization: Nonuniform quantization techniques, such as A-law and U-law, are designed to allocate more quantization levels to low-amplitude signals, reducing quantization noise for signals with nonuniform distributions. In our study, non-uniform quantization schemes are applied to mitigate the effects of quantization noise introduced by low-resolution DACs. By experimenting with A-law and U-law quantization, the project evaluates how these techniques can improve BER and SE in our systems[17]. These schemes are particularly relevant when simulating scenarios with varying DAC resolution bits, as they demonstrate how non-uniform quantization can offset the performance degradation caused by reduced resolution.

## 4.3 System Design

### 4.3.1 Source Signal Generation

The system begins with the generation of binary source data. This binary sequence serves as the input data stream, which is subsequently modulated into symbols for transmission. The bitstream is generated randomly, representing a stream of 0s and 1s.

For each simulation scenario, the number of bits generated depends on the modulation scheme used. For instance, modulation schemes like 64-QAM, multiple bits (4 or 6) are grouped to form a single symbol.

This step is critical for enabling a variety of modulation schemes and ensuring that the system is tested under diverse conditions.

### 4.3.2 Modulation and Demodulation

Various modulation schemes are achieved in this project using a Modulator function written in MATLAB, which accepts the input binary data and the modulation type identifier. The function processes the input bit stream according to the modulation scheme and generates a complex modulated signal consisting of the In-phase and Quadrature components. Modulation schemes BPSK, QPSK, 16QAM, and 64QAM are custom-designed for this project.

The general mathematical expression used for design modulation is:

$$s(t) = A_I \cos(2\pi f_c\, t) + A_Q \sin(2\pi f_c\, t)$$

Where:

- $A_I$ is the amplitude of the in-phase component.

- $A_Q$ is the amplitude of the quadrature component.

- $f_c$ is the carrier frequency.

Based on the equation and the modulation scheme chosen with modulation order M, M number of input bits is mapped to each symbol. In BPSK, 1 bit is mapped per symbol following the mapping 0 to -1, 1 to 1, which represents a phase shift of the carrier signal between 0 (for bit '1') and $\pi$ (for bit '0').

In QPSK, 2 bits are mapped per symbol following the phase shift mapping: $\frac{\pi}{4}$ if the bits are 00, $\frac{3\pi}{4}$ if the bits are 01, $\frac{5\pi}{4}$ if the bits are 11, $\frac{7\pi}{4}$ if the bits are 10. For 16QAM, 4 bits are transmitted per symbol by mapping 2 bits to the in-phase component and 2 bits to the quadrature component of each symbol. The bit mapping used in 16QAM is -3 if bits are 00, -1 if bits are 01, +1 if bits are 11, and +3 if bits are 10. 64QAM mapping and implementation design are discussed in detail below, as 64QAM is used for most of the experiments done as part of this project. 6 bits are transmitted per symbol in 64QAM. This enhances spectral efficiency, and each symbol is represented by a unique combination of both amplitude and phase values. The input bit-stream must be a multiple of six to ensure proper grouping. The 64-QAM constellation has 64 possible points.

In 64-QAM, each 6-bit combination is mapped to one of the 64 constellation points. Both $A_I$ and $A_Q$ can take one of 8 amplitude levels, chosen from the set $\{-7,-5,-3,-1,1,3,5,7\}$.

The current implementation uses bit-to-symbol mapping for $A_I$ and $A_Q$ components as below:

- 000: −7
- 001: −5
- 011: −3
- 010: −1
- 110: +1
- 111: +3
- 101: +5
- 100: +7

Each symbol corresponds to a unique pair of amplitudes $A_I$ and $A_Q$, with 8 different amplitude levels on both the in-phase (I) and quadrature (Q) axes. The transmitted 64-QAM signal is a combination of $A_I$ and $A_Q$ components, where $A_I$ is obtained by bit mapping of the first three bits of the 6-bit grouped input data, and $A_Q$ is obtained by bit mapping of the last three bits of the 6-bit grouped input data.

$$s(t) = A_I \cos(2\pi f_c\, t) + A_Q \sin(2\pi f_c\, t)$$

Where:
- $A_I \in \{-7,-5,-3,-1,1,3,5,7\}$
- $A_Q \in \{-7,-5,-3,-1,1,3,5,7\}$

This modulation scheme efficiently transmits 6 bits per symbol, allowing for higher data rates compared to lower-order QAM schemes like 16-QAM.

### 4.3.3 Digital-to-Analog Conversion

The Digital-to-Analog Conversion process transforms the digital modulated symbols into continuous-time analog signals, which are required for wireless transmission. The DAC receives in-phase (I) and quadrature (Q) components of the symbol stream and outputs an analog signal. Three quantization schemes are used for this process: Uniform, A-law, and $\mu$-law. These schemes are widely adopted in telecommunication systems for their efficiency in handling signals with large dynamic ranges. The parameters $A$ = 87.6 and $\mu$ = 255 are standard values defined in modern telecommunication standards [9]. Each scheme follows a distinct approach for signal quantization and compression.

Data Flow and Input/output Signals DAC takes the in-phase (I) and quadrature (Q) components of the symbol stream as input. Each component is processed through one of the quantization methods (Uniform, A-law, or $\mu$-law) to produce a quantized signal. The quantized signal is then converted into a continuous-time analog signal suitable for wireless transmission. The output of the DAC is an analog signal that can be transmitted over a wireless channel.
The entire process can be summarized as follows: 1. Input: I/Q components of the symbol stream (digital signal). 2. Quantization: I and Q components are quantized using Uniform, A-law, or $\mu$-law methods. 3. Conversion: Quantized signals are converted into continuous-time waveforms. 4. Output: A combined analog signal (with I and Q components) ready for transmission.

### 4.3.4 Channel Model

A simple AWGN channel is used in this research. This channel acts as a baseline model to study the degradation of symbols due to random noise. It is described below how the channel model is programmed.
The channel model function takes the modulators' output and the dB SNR value as inputs[18]. The SNR is a critical parameter in wireless communication systems, representing the ratio of signal power to noise power. The relationship between SNR and noise power is expressed as:

$$P_n = \frac{P_S}{10^{\frac{SNR}{10}}}$$

The signal power is calculated by taking the mean of the squared magnitude of the transmitted signal, expressed as:

$$P_S = \frac{1}{N}\sum_{n=1}^{N} |X_n|^2$$

where N is the total number of samples and Xn represents the transmitted signal samples. The noise[19] is generated as a Gaussian random variable with zero mean and a variance equal to the noise power $P_n$. The noise W is defined as:

$$W \sim N\,(0, P_n)$$

In the simulation, the MATLAB randn function generates the noise, which is then scaled by the square root of the calculated noise power.
The received signal at the receiver is obtained by the equation below:

$$Y = X + W$$

This model assumes that the primary distortion in the wireless channel comes from the additive noise and does not include other real-world factors such as fading or interference[20]. To ensure the consistency and reproducibility of the simulation, a fixed random seed is used when generating the AWGN noise. This allows the experiment to be repeatable with the same noise conditions across different runs.

### 4.3.5 Analog-to-Digital Converters

The Analog-to-Digital Conversion process transforms continuous-time analog signals into discrete digital signals, which are required for subsequent processing in the communication system. The ADC receives analog signals from the output of the wireless channel and converts them into quantized symbols using one of three quantization schemes: Uniform, A-law, or $\mu$-law[21].

The entire process can be summarized as follows: 1. Input: Continuous time analog signal (I/Q components) from the channel. 2. Signal Quantization: The signal is quantized using Uniform, A-law, or $\mu$-law quantization methods. 3. Bitstream Conversion: The quantized signal is converted back into binary form, which can be processed by the demodulator.

## 4.4 Various quantization schemes

### 4.4.1 Uniform Quantization scheme

Uniform quantization is one of the simplest forms of quantization. In this method, the range of input signal values is divided into equal-sized intervals. A quantization level represents each interval. Given a signal $x$, the quantized signal $x_q$ can be mathematically expressed as:

$$x_q = Q(x) = \Delta . [\frac{x}{\Delta} + 0.5]$$

where $\Delta$ represents the quantization step size.

### 4.4.2 A-law Companding

A-law companding is a logarithmic compression technique. It compresses low-amplitude signals more significantly than high-amplitude signals, ensuring efficient quantization.

$$F(x) = \begin{cases} \frac{A|x|}{1+\ln(A)}, & 0 \le |x| < \frac{1}{A} \\ \frac{1+\ln(A|x|)}{1+\ln(A)}, & \frac{1}{A} \le |x| \le 1 \end{cases}$$

where A is the compression parameter.

### 4.4.3 μ-law Companding

$\mu$-law companding is another widely used logarithmic compression method that provides similar benefits to A-law. $\mu$-law provides higher compression for small signals, making it particularly suitable for scenarios with a wide dynamic range of signal amplitudes.

$$F(x) = \frac{\ln(1+\mu|x|)}{\ln(1+\mu)}, \quad -1 \le x \le 1$$

where $\mu$ is the compression parameter.

## 4.5 Precoding Techniques

### 4.5.1 ZF Precoding

Zero-forcing precoding is a technique used to eliminate multi-user interference in MISO systems[22]. The ZF precoder attempts to find a beamforming matrix $W$ that satisfies the following condition:

$$W = H^H(HH^H)^{-1},$$

where $W$ is the precoding matrix and $H$ is the channel matrix. Here, $H^H$ denotes the Hermitian transpose of $H$, which is obtained by taking the complex conjugate of each element in $H$ and then transposing the matrix.

For a MISO system with $M$ transmit antennas and $K$ users, $H$ is a $K \times M$ matrix, and $W$ is an $M \times K$ matrix.

One key issue in the ZF design is the potential numerical instability of the matrix inverse $(HH^H)^{-1}$, especially when $H$ is ill-conditioned. To address this, the Moore-Penrose pseudo-inverse is often used instead:

$$W = H^H \cdot \mathrm{pinv}(HH^H),$$

where pinv(·) represents the pseudo-inverse operation. The Moore-Penrose pseudo-inverse provides a more numerically stable solution, particularly when $HH^H$ is not full rank or poorly conditioned, ensuring robust precoding performance.

### 4.5.2 MRT Precoding

MRT maximizes the received SNR for a single-user MISO system[23]. The MRT beamforming matrix $W$ is calculated as:

$$W = \frac{H^H}{\|H^H\|_F}$$

where $\|H^H\|_F$ represents the Frobenius norm of $H^H$. The weights are applied to the transmitted signal to maximize the power of the received signal at the intended user.

MRT is ideal for single-user systems since it maximizes SNR. However, for multi-user systems, it introduces interference among users, making it less efficient than ZF in multi-user scenarios.

### 4.5.3 Key Performance Parameters

Several parameters define the performance of a DAC, including:

- Resolution: The number of bits representing the digital input determines the DAC's resolution. Higher resolution allows for finer granularity in the output, translating to a more accurate analog signal.
- Sampling Rate: The rate at which the DAC converts digital data to an analog signal. It defines the speed of conversion and is crucial in applications where real-time data processing is required.
- Linearity: This parameter defines how closely the DAC output follows the expected analog value. Good linearity ensures that the DAC output is proportional to the digital input across its entire range.
- Settling Time: The time it takes for the DAC output to stabilize to its final value after a digital code change. Fast settling times are essential in high-speed systems.

### 4.5.4 Challenges of Low-Resolution DACs

While low-resolution DACs offer benefits, their use introduces significant challenges:

- Quantization Noise: In low-resolution DACs, fewer quantization levels are available, leading to quantization noise. This noise results from the approximation error introduced during the conversion process. Quantization noise distorts the transmitted signal, which can degrade key performance metrics such as BER and SE, particularly in high-order modulation schemes like 16-QAM and 64-QAM. Reduced resolution increases quantization errors, leading to higher Bit Error Rates (BER) and degraded signal fidelity.

- Non-Linear Distortions: The non-linear characteristics of low-resolution DACs cause distortions, particularly affecting higher-order modulations like 16-QAM and 64-QAM.
- Out-of-Band Emissions: Low-resolution DACs contribute to unwanted spectral leakage, reducing spectral efficiency and causing interference.
- Precoding Complexity: Techniques such as Zero Forcing (ZF) and MRT must adapt to the imperfections introduced by low-resolution DACs.

## 4.6 Channel Estimation

In wireless communication systems, the channel estimation technique is vital for evaluating the performance of the MIMO system. Furthermore, channel estimation is responsible for obtaining the CSI, where CSI provides information about channel properties of the wireless communication link estimated at the receiver side, and the transmitter is fed back with this information[24]. This study focuses on the study of performance of Least Square (LS) compared to the ideal channel matrix H. The goal of the channel least square estimator [25]is to minimize the square distance between the received signal and the original signal. The least squares (LS) estimates of the channel at the pilot subcarriers are given by:

$$Y = X * H + N$$

can be obtained by the following equation:

$$H_{ls} = X^{-1} * Y$$

where $H_{ls}$ represents the least squares (LS) estimate obtained over the pilot subcarriers.

## 4.7 Performance metrics used to evaluate the impact of DACs

In this section, key performance metrics are described, which are used to evaluate the impact of DAC resolution bits on the performance of massive MISO systems. These metrics provide insight into how different DAC resolutions affect the accuracy and efficiency of signal transmission.

### 4.7.1 Bit Error Rate (BER)

BER is a critical metric for evaluating the performance of DACs in communication systems. BER is defined as the ratio of bits received incorrectly to the total number of bits transmitted. A lower BER indicates better performance, as fewer errors occur during transmission. The DAC's resolution, noise levels, and signal distortion contribute to the overall BER, which helps assess the DAC's ability to accurately convert digital signals into analog form without introducing significant errors. In high-performance systems, achieving a low BER is essential for maintaining the integrity of the transmitted data.

### 4.7.2 Signal-to-Noise Ratio (SNR)

SNR is a measure of the strength of the desired signal relative to background noise. A higher SNR indicates that the signal is stronger than the noise, which improves communication quality. The performance of a DAC can significantly impact SNR as higher-resolution DACs tend to produce less quantization noise, thereby improving SNR. Evaluating the DAC's effect on SNR helps determine its suitability for applications that require precise signal reproduction, such as high-fidelity audio and data transmission.

### 4.7.3 Spectral Efficiency (SE)

SE is a critical performance metric in wireless communication systems, particularly in MISO and MIMO systems. It measures how efficiently the available bandwidth is utilized to transmit information. When evaluating the impact of a Digital-to-Analog Converter (DAC) in a MU-MISO system, spectral efficiency provides insights into how the DAC's resolution, quantization error, and non-linearities affect system capacity and data transmission efficiency.

# Chapter 5 Results and Analysis

## 5.1 System performance with different modulation schemes

This section discusses the evaluation of different modulation schemes like BPSK, QPSK, 16QAM, and 64QAM. SISO setup is used for this evaluation, and the configuration used is listed in Table 4.1:

*Table 2: Simulation parameters used to evaluate modulation schemes*

| **Parameter** | **Value / Description** |
|---|---|
| numSymbols | 1e5 |
| SNR range (dB) | -20:2:30 |
| Modulation Scheme | BPSK, QPSK, 16-QAM, and 64-QAM |
| Number of BS Antennas | 1 |
| UE | 1 (single antenna) |
| Channel Model | AWGN |
| Performance Metric | BER |

The obtained constellation diagram for each modulation scheme with the SISO setup under test is shown in Figure 4.

### 5.1.1 Constellation diagram

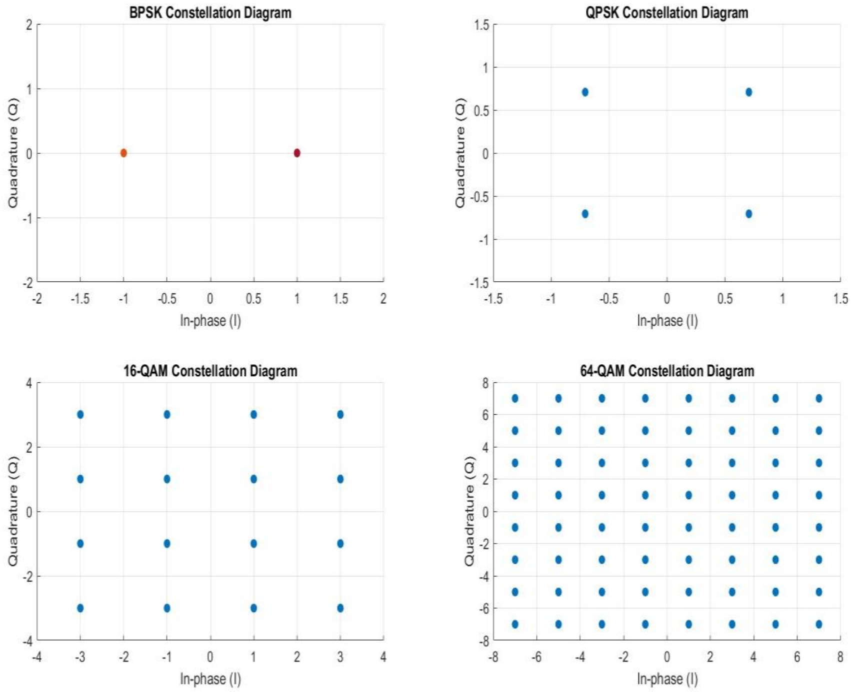


*Figure 4: Constellation Diagram for different modulation schemes*

Detailed analysis of each constellation diagram provides insights into the modulation behavior and signal robustness in the system:

1. BPSK Constellation Diagram:
   - Description: The BPSK constellation diagram (top left plot in figure 4) displays two distinct points along the in-phase (I) axis, with values at +1 and -1. There are no components along the quadrature (Q) axis.
   - Interpretation: The BPSK modulation scheme uses two phases to indicate the binary values 0 and 1. It provides high noise immunity owing to the large spacing between the two points. this condition reduces the possibility of an error due to noise.
   - Performance impact: With only two possible symbols, BPSK is the most robust against noise but has the lowest spectral efficiency because it transmits only 1 bit per symbol.

2. QPSK Constellation Diagram:

- Description: The QPSK constellation diagram (top right plot in figure 4) consists of four points, symmetrically arranged at equal distances in the quadrants of the in-phase (I) and quadrature (Q) axes. Each point represents a unique combination of 2 bits.
- Interpretation: The constellation points are positioned at ±0.707 on both the I and Q axes. QPSK doubles the data rate compared to BPSK by transmitting 2 bits per symbol. The Gray-coded scheme minimizes the bit error rate by ensuring that adjacent symbols differ by only one bit.
- Performance impact: QPSK strikes a good balance between data rate and noise immunity. The distance between the constellation points is sufficient to provide noise resilience, although not as robust as BPSK.

3. 16-QAM Constellation Diagram:

- Description: The 16-QAM constellation diagram (bottom left plot in figure 4) displays 16 distinct points arranged in a 4x4 grid with equal spacing between them. The I and Q values take discrete amplitude levels of -3, -1, +1, and +3.
- Interpretation: Each point in the diagram corresponds to a 4-bit symbol. The higher number of points increases the data rate by allowing 4 bits to be transmitted per symbol. However, the points are closer together than in QPSK, making the modulation scheme more sensitive to noise and interference.
- Performance Impact: 16-QAM offers a higher spectral efficiency compared to QPSK and BPSK. However, it requires a higher SNR to maintain the same BER because the points are closer, making it more prone to errors in noisy conditions.

4. 64-QAM Constellation Diagram:

- Description: The 64-QAM constellation (bottom right plot in figure 4) shows 64 distinct points in an 8x8 grid, with the I and Q components taking integer values from -7 to +7. The points are much more densely packed compared to lower-order modulation schemes.
- Interpretation: Each point represents a 6-bit symbol, providing the highest data rate among the four modulation schemes. However, the proximity of the points to each other means that even small noise disturbances or signal distortions can lead to incorrect decoding.
- Performance Impact: 64-QAM offers the highest spectral efficiency, transmitting 6 bits per symbol, but it is the most vulnerable to noise and distortion. It requires a very high SNR to operate effectively, making it suitable for environments with excellent signal quality.

### 5.1.2 BER Vs. SNR curve for different modulation schemes

Figure 5 shows the performance of the custom-built modulation schemes using the SISO simulator for different modulation schemes like BPSK, QPSK, 16QAM, and 64QAM compared to the inbuilt modulation code blocks of MATLAB. The curve shows how error rates change as the signal quality improves for different modulation schemes and how it looks compared to theoretical BER, which is generated using inbuilt modulation and demodulation blocks. Comparing theoretical performance with the performance of custom-built modulator and demodulator, it can be seen that all modulation schemes perform close to theoretical performance except for BPSK. BPSK (1 bit per symbol) BER plot using inbuilt modulation and demodulation has the lowest error rate because the signal is very distinct, making it easier to correctly decode even in noisy conditions. Custom-built BPSK performs a bit worse than inbuilt BPSK because of the mapping used in custom BPSK modulation code. QPSK (2 bits per symbol) performs well but slightly worse than BPSK because it uses more closely spaced symbols, so noise can cause more confusion between them. In 16-QAM and 64-QAM, even more bits are packed into each symbol (4 bits and 6 bits, respectively), but this means the signal points are very close to each other. As a result, it becomes harder to distinguish between them when noise is present, causing more errors unless the signal is strong (high SNR). This explains why higher modulation schemes have higher error rates for the same SNR—they trade off noise resistance for higher data rates.

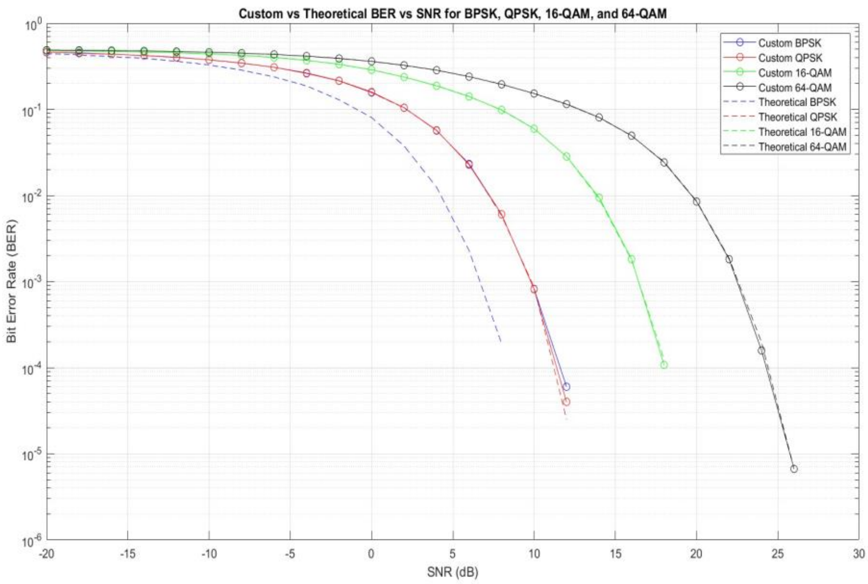


*Figure 5: Plot BER vs. SNR using SU-SISO setup: custom and theoretical modulation*

The above figure is used to compare the performance between the inbuilt modulation blocks of MATLAB and custom-built modulator and demodulator for a perfect CSI channel.

### 5.1.3 Summary of modulation schemes performance

The constellation diagrams highlight the trade-offs between spectral efficiency and noise immunity. As the modulation order increases (from BPSK to 64-QAM), the number of bits per symbol increases, enhancing the data rate. However, the points in the constellation become more closely packed, making the higher-order modulation schemes more susceptible to noise. In practice, lower-order modulation schemes like BPSK and QPSK are preferred in noisy environments or when signal quality is poor due to their robustness. Higher-order schemes like 16-QAM and 64-QAM are ideal for high-quality channels where maximizing throughput is crucial but require a stronger SNR to maintain reliability. This analysis of the constellation diagram and BER performance curve provides a comprehensive view of how each modulation scheme performs in terms of data rate, noise resilience, and spectral efficiency. Based on the above observations, the 64QAM modulation scheme is selected for future experiments with the enhanced simulation setup.

## 5.2 System performance with DAC

For preliminary understanding, the SU-SISO simulation setup as described in Table 3, is used to evaluate DAC resolution bits' impact on system performance. For 64-QAM modulation, the performance of the system with different resolution bits and, in the ideal case, without a DAC/ADC is characterized. The uniform quantization scheme means that a range of input digital values falls into a corresponding analog value with equally spaced intervals. The plot in Figure 5 show the BER vs. SNR curves for the 64-QAM with different resolutions of DAC/ADC, namely 1-bit, 2-bit, 4-bit, and 8-bit, compared to the ideal curve without a DAC/ADC. Higher resolutions, like 8-bit, result in performances very close to the ideal case. For lower resolutions, this will introduce large errors into the system due to significant quantization noise. These results show the influence of the resolution of DAC/ADC on the overall system performance.

*Table 3: Simulation parameters used to evaluate performance of DAC*

| Parameter | Value / Description |
|---|---|
| numSymbols | 1e5 |
| SNR range (Db) | -20:2:30 |
| DAC/ADC Resolution bits | 2, 4, and 8 bits |
| Quantization Scheme | Uniform quantization |
| Modulation Type | 64QAM |
| Number of BS Antennas | 1 |
| Number of Ues | Single User (single antenna) |
| Channel Model | AWGN |

| | |
|---|---|
| Performance Metrics | BER and SE |

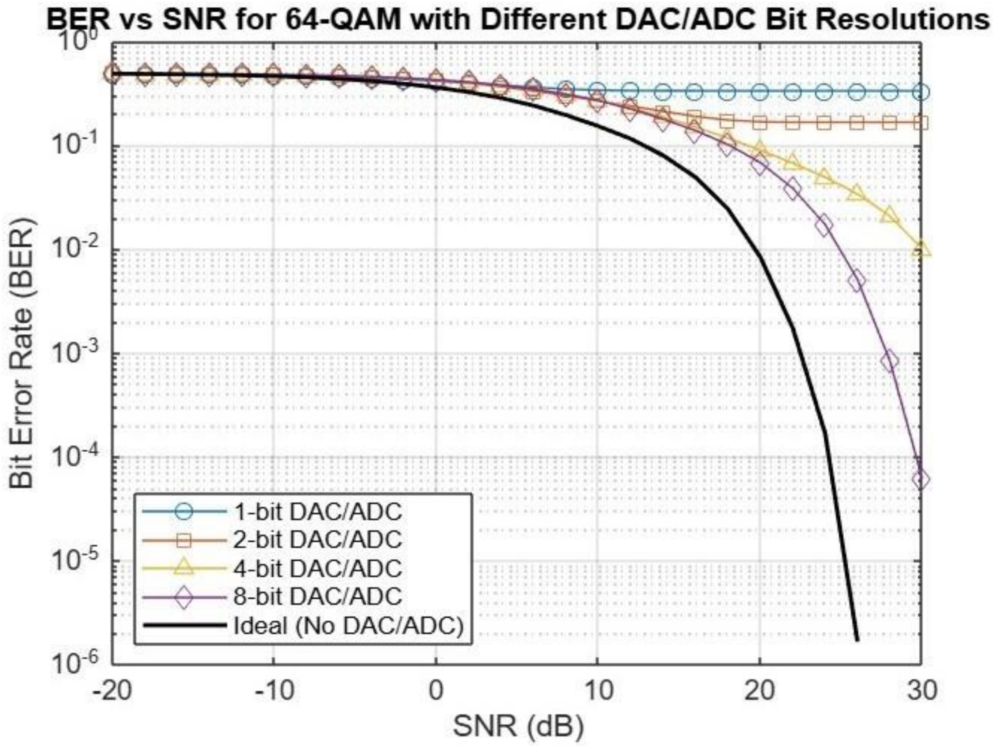


*Figure 6: BER vs. SNR for SU-SISO setup: uniform quantization*

This figure compares the performance of uniform DAC for different resolution bits with the ideal case, with no DAC and perfect CSI for 64QAM. Furthermore, to see the effect of DAC on the existing simulation setup, the uniform quantization scheme is replaced by non-uniform quantization schemes like A-law and $\mu$-law implementations in the simulation parameters of in Table 3, and performance curves are obtained for this system to see how non-uniform quantization schemes affect the overall performance.

First, the A-law quantization scheme is tested for different A values to see which value of A gives a maximum improvement in performance, followed by evaluating the $\mu$-law quantization scheme for different $\mu$values. Plots 7 show the performance of the A-law quantization scheme for different A values of 10, 87.6, and 100 for a 2-bit DAC. Similarly, plots 8 show the performance of the A-law quantization scheme for different A values of 10, 87.6, and 100 for an 8-bit DAC. The plots show that A = 10 is better for DAC with higher resolution bits, and for lower resolution bits, any A value gives similar performance. Since the rest of the experimentation focuses on low-bit DACs, to align with the theoretical expectations of A-law quantization scheme performance, 87.6 is used for future experiments since the proposed A value is 87.6 according to the G.711 Standard for European systems

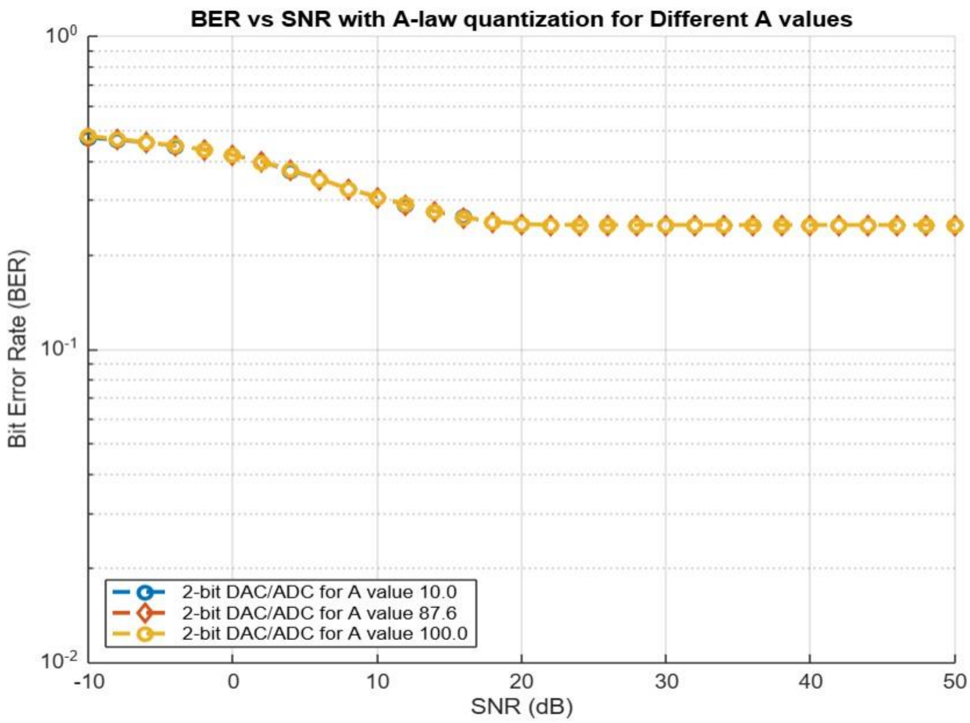


*Figure 7: BER vs. SNR for SU-SISO setup: A-law quantization for 2-bit 64-QAM modulation*

The above figure compares performance by replacing uniform quantization with A-law quantization for different A values for 2-bit resolution and modulation order M=64.

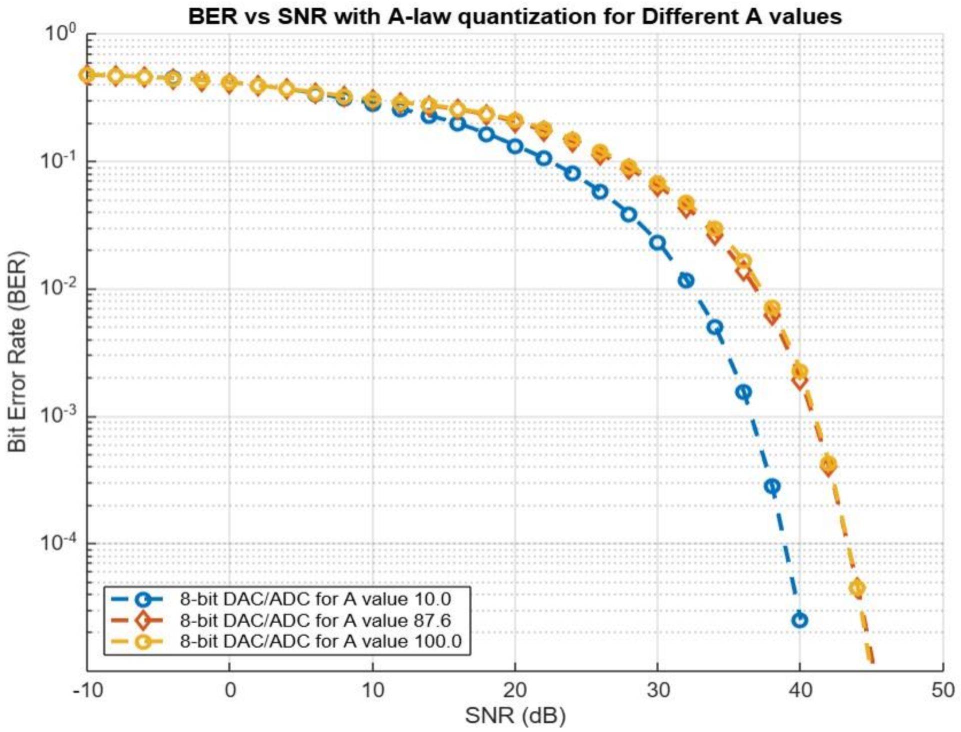


*Figure 8: BER vs. SNR for SU-SISO setup: A-law quantization for 8-bit 64-QAM modulation*

A similar evaluation is done for the $\mu$-law quantization scheme for different $\mu$values, such as 128, 255, and 300. The performance curves for these values are as shown in Figures 9 and 10. From these plots, it's seen that $\mu=128$ gives better performance for DAC with 8-bit resolution; and for 2 bits, all $\mu$values have similar performance. Thus, $\mu=255$ is used for future experiments as the expectation is to evaluate the impact of different quantization schemes on performance for low-resolution DAC, and also because $\mu=255$ is the recommended value for $\mu$-law quantization scheme as per ITU-T G.711 standard in North America and Japan.

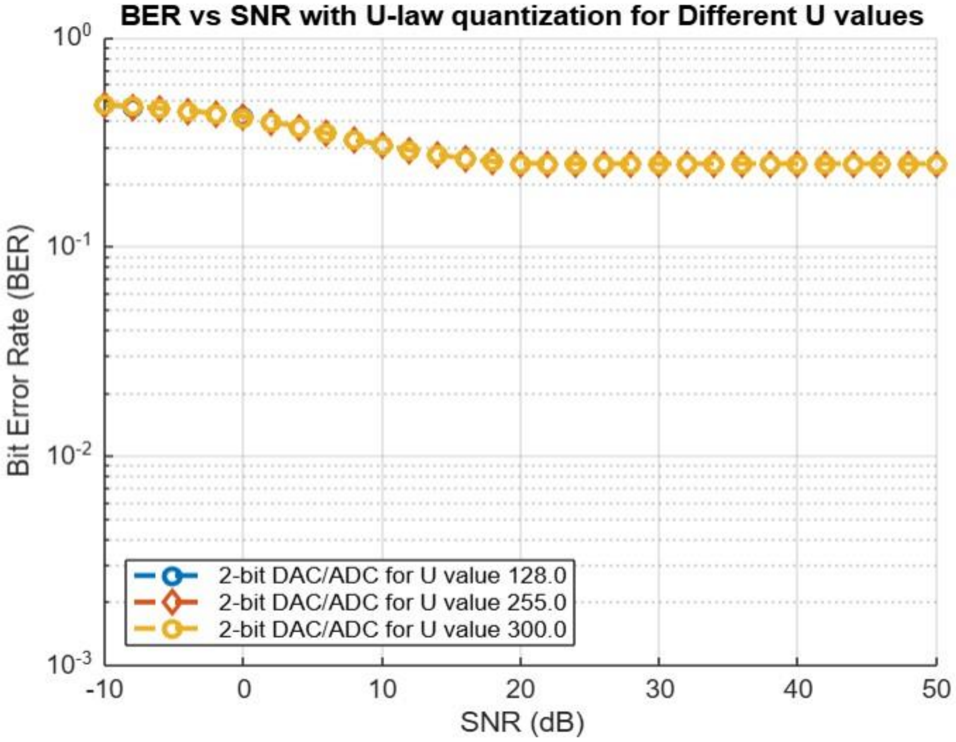


*Figure 9: BER vs. SNR for SU-SISO setup: μ-law quantization for 2-bit 64-QAM modulation*

Figure 9: BER vs. SNR for SU-SISO setup by replacing uniform quantization with $\mu$-law quantization for different U values for 2-bit resolution and modulation order M=64.

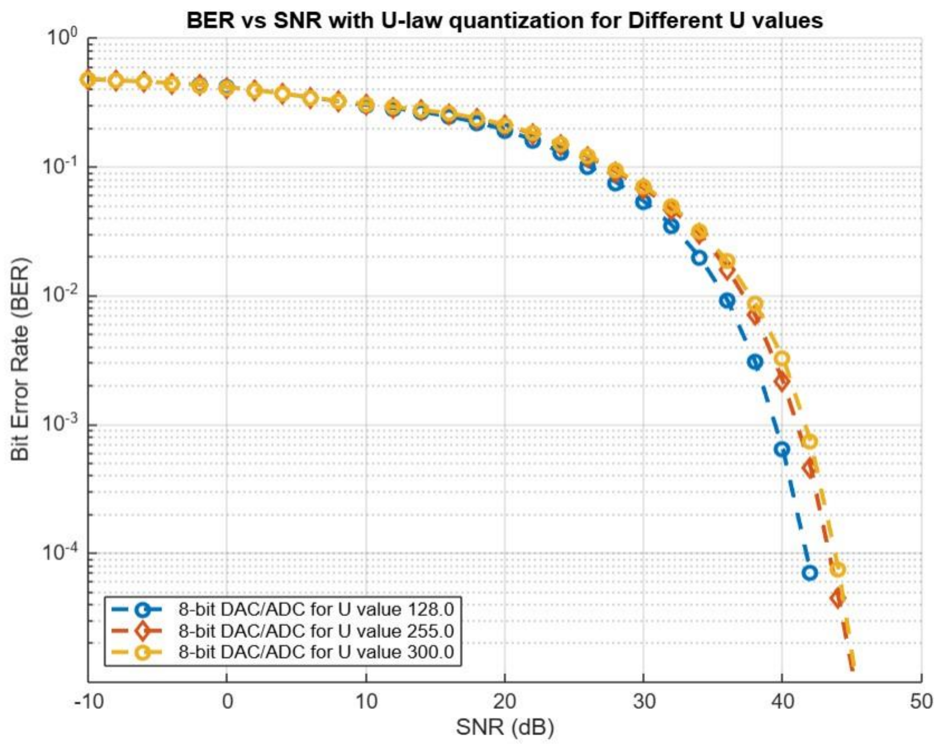


*Figure 10: BER vs. SNR for SU-SISO setup: μ-law quantization for 8-bit 64-QAM modulation*

Figure 10: BER vs. SNR for SU-SISO setup by replacing uniform quantization with $\mu$-law quantization for different U values for 8-bit resolution and modulation order M=64.

Based on the above experiment, uniform and non-uniform quantization schemes are now compared with each other, using an A-value of 87.6 for the A-law quantization scheme and a $\mu$ value of 255 for the $\mu$-law quantization scheme. The BER vs. SNR curve in Figure 11 and the SE vs. SNR curve in Figure 12 provide the output of uniform, A-law, and $\mu$-law quantization comparisons for 8 and 2-bit DAC resolution. The purpose of choosing 8 and 2 bits is to discuss the impact of high-resolution and low-resolution bits on system performance, respectively. Key observations:

- 8-bit $\mu$-law and A-law perform best, achieving the lowest BER and thus correspondingly high SE. This is because $\mu$-law and A-law at higher resolutions (8-bit) improve small-signal precision due to their non-linear compression.
- At 2-bit resolution, uniform quantization outperforms $\mu$ and A-law in terms of BER and SE because non-linear companding at low resolution causes

errors for larger amplitudes. This is because uniform quantization performs better than non-uniform methods in this case, as the input signal distribution is uniform and there is no precoder in between the data processed by the DAC, ensuring the output remains uniform.

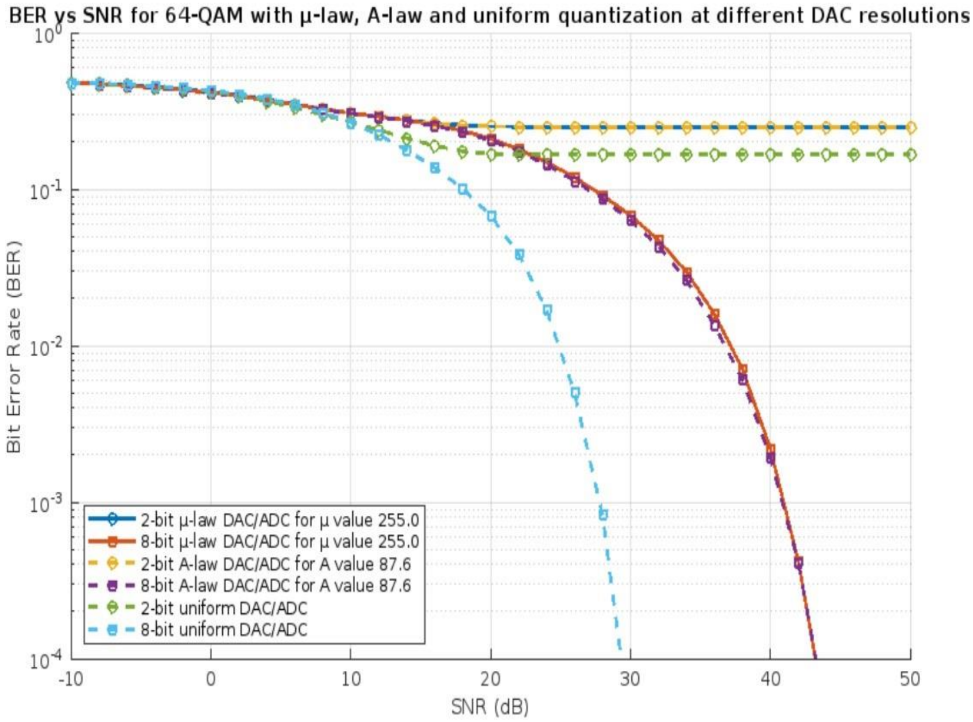


*Figure 11: Plot that compares BER VS SNR for A law, U law, and Uniform quantization at 2 and 8-bit resolution for the SU-SISO setup for 64-QAM*

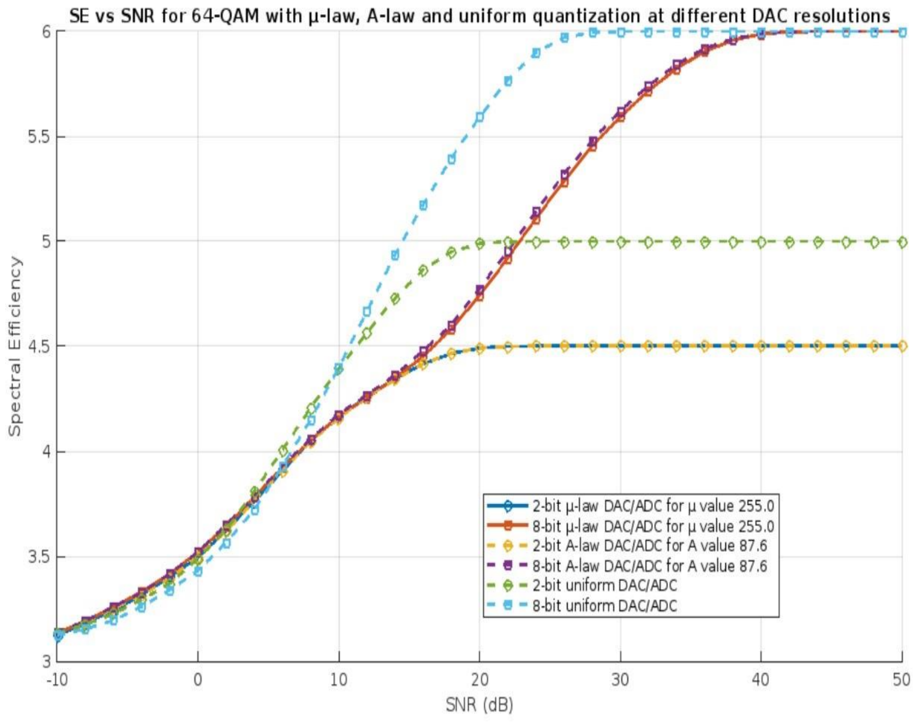


*Figure 12: Plot that compares SE VS SNR for A law, U law, and Uniform quantization at 2 and 8-bit resolution for the SU-SISO setup with 64QAM.*

## 5.3 System performance with precoders and channel estimation

In this section, the performance evaluation of the simulation setup with precoders and channel estimation will be discussed. The different precoding methods under study are MRT and ZF; the channel estimation method used is LSThe setup used for this evaluation has the configurations described in Table 4.

*Table 4: Parameters used to evaluate the performance of DAC with precoders*

| Parameter | Value / Description |
|---|---|
| SNR range (dB) | -30:2:30 |
| DAC/ADC Resolution bits | 2 |
| Quantization Schemes | Uniform, A-law, and $\mu$-law |
| Modulation Types | 64-QAM |
| Number of BS Antennas | 64 antennas |
| Number of UEs | 6 single antenna UEs |
| Channel estimation | Ideal H and LS |
| Precoders | MRT and ZF |

| | |
|---|---|
| Monte Carlo Iterations | 1000 |
| Performance Metrics | BER and SE |

### 5.3.1 Key observations on performance curves

Figure 13 and Figure 14 represent the performance metrics curves (BER and SE) generated for configuration: 6 UEs in a MU-MISO simulation setup with 2-bit DAC resolution using MRT precoding and ZF precoding, respectively. The comparison includes three quantization schemes: A-law, $\mu$-law, and uniform quantization, with and without channel estimation (LS-based and ideal H).
The following sections discuss the effects of different quantization schemes applied along with the precoders on the system performance.

#### 5.3.1.1 BER vs SNR (Top Plot in 13 and 14):

- A-law and $\mu$-law quantization achieve the best BER performance across all SNR values for both precoding schemes individually. This is because A-law and $\mu$-law are non-linear companding schemes that allocate more quantization levels to the small-amplitude signals, thereby enhancing the precision where the signal values are more concentrated. In the 2-bit DAC resolution case, this allocation significantly reduces quantization noise, leading to better BER performance compared to uniform quantization.

- Uniform quantization has the highest BER at all SNRs because, using a 2-bit resolution DAC, uniform quantization loses its accuracy in small amplitudes, which would eventually cause a higher BER compared to A-law and $\mu$-law.

- Even though LS-based estimation causes a slight degradation overall, compared to the ideal H, A-law, and $\mu$-law, still reduce quantization noise better than uniform quantization. The cause for degradation with LS addition is that channel estimation introduces estimation errors, which could lead to higher BER.

#### 5.3.1.2 SE vs SNR (Bottom Plot in 13 and 14):

- $\mu$-law quantization achieves the highest spectral efficiency at higher SNR values.

- A-law quantization follows closely behind $\mu$-law and outperforms uniform quantization in SE.

- Introducing LS channel estimation causes a slight degradation in SE similar to the BER trend. This is due to estimation errors introduced by channel estimation.

5.3.1.3 Conclusion of Performance Curves

Both MRT and ZF precoding schemes enhance the performance of non-uniform quantization schemes (A-law and $\mu$-law) compared to uniform quantization for both ideal H and estimated H by effectively addressing the limitations of 2-bit DAC resolution, providing improvements in BER and spectral efficiency. The improved performance of non-uniform quantization schemes over uniform quantization is because, with precoding, the output becomes non-uniform, which aligns better with the non-uniform quantization techniques, leading to improved performance.

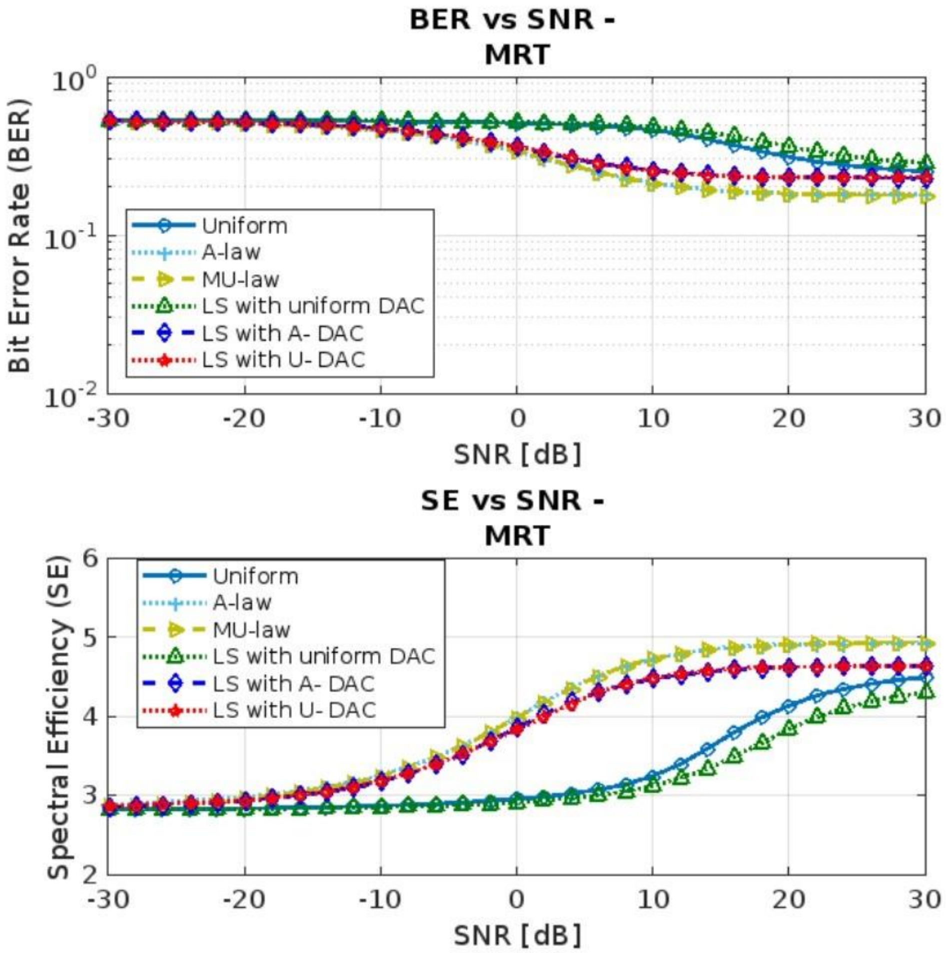


*Figure 13: Plot that compares the performance of 2-bit uniform and non-uniform DACs with MRT precoding with and without channel estimation for par.mod = 64QAM, par.U = 6, and par.B = 64*

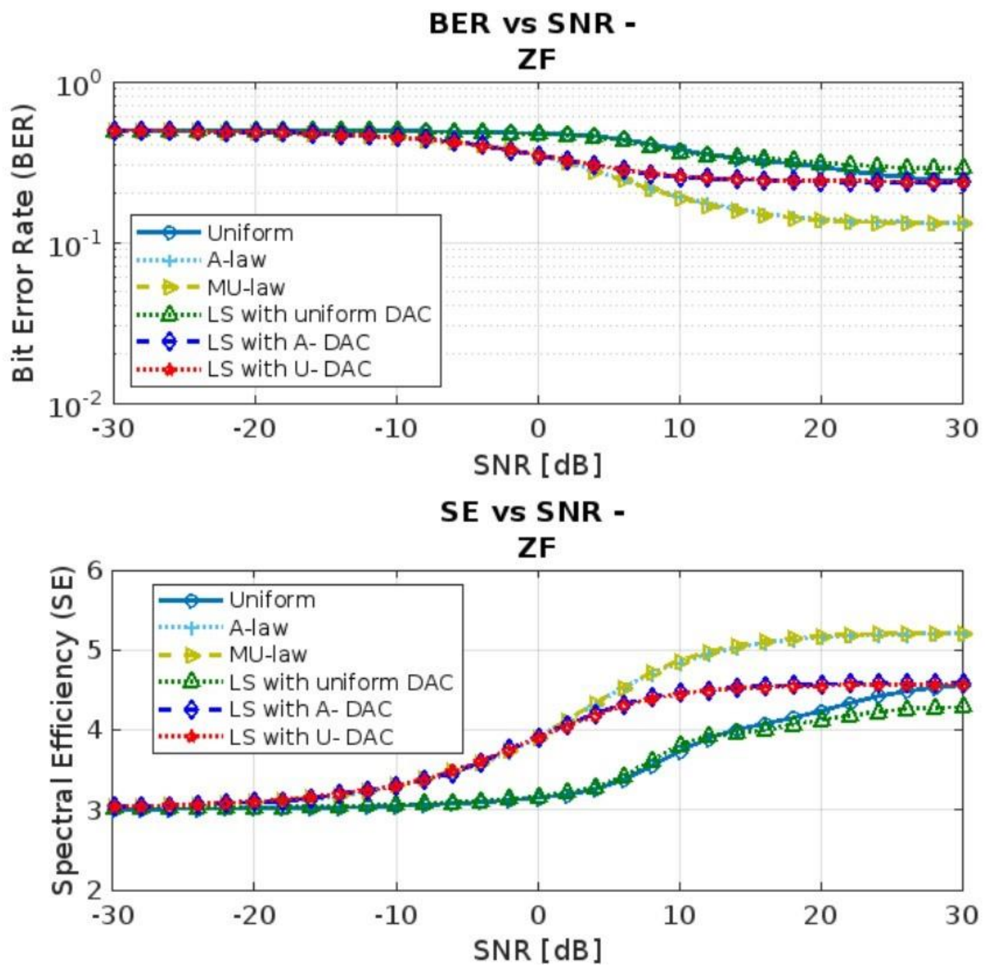


*Figure 14: Plot that compares the performance of 2-bit uniform and non-uniform DACs with ZF precoding with and without channel estimation for par.mod = 64QAM, par.U = 6, and par.B = 64*

### 5.3.2 Numerical results to compare MRT and ZF performance

Key Insights and Performance Analysis: Table 4.4 and 4.5 are the numerical values of BER and SE, respectively, for uniform quantization, A law, and $\mu$ law for MRT and ZF precoders.
In a multi-user scenario with 6 users, it is observed in Tables 4.4 and 4.5 that ZF outperforms MRT for all quantization schemes at every SNR value. This can be explained by the key reasons listed below.

Underlying factors of MRT and ZF precoder performance

1. Interference Suppression: ZF eliminates inter-user interference by ensuring the orthogonality of transmitted signals using the precoding matrix. In contrast, MRT maximizes individual signal power but does not suppress interference.

2. Multi-User Trade-off: As the number of users increases (e.g., 6 users), inter-user interference dominates system performance. MRT suffers from

significant interference, while ZF minimizes this interference at the cost of power efficiency.

*Table 5: BER Values for different SNRs*

| SNR | Uni_ZF | Uni_MRT | A_ZF | A_MRT | U_ZF | U_MRT |
|---|---|---|---|---|---|---|
| -30 | 0.4164 | 0.4701 | 0.4174 | 0.4696 | 0.4152 | 0.4688 |
| -25 | 0.4157 | 0.4694 | 0.4164 | 0.4675 | 0.4146 | 0.4652 |
| -20 | 0.4145 | 0.4685 | 0.4149 | 0.4636 | 0.4143 | 0.4612 |
| -15 | 0.4125 | 0.4666 | 0.4143 | 0.4549 | 0.413 | 0.4522 |
| -10 | 0.4091 | 0.4626 | 0.4044 | 0.436 | 0.4052 | 0.4336 |
| -5 | 0.4021 | 0.456 | 0.3828 | 0.396 | 0.3824 | 0.3891 |
| 0 | 0.39 | 0.445 | 0.3309 | 0.3335 | 0.3288 | 0.3277 |
| 5 | 0.3726 | 0.4297 | 0.2536 | 0.2719 | 0.2522 | 0.2674 |
| 10 | 0.3689 | 0.421 | 0.1895 | 0.2266 | 0.1885 | 0.2253 |
| 15 | 0.3578 | 0.3739 | 0.1543 | 0.2051 | 0.1528 | 0.2046 |
| 20 | 0.315 | 0.3045 | 0.1391 | 0.1971 | 0.14 | 0.1989 |
| 25 | 0.2646 | 0.2586 | 0.1334 | 0.1942 | 0.1358 | 0.1974 |
| 30 | 0.2309 | 0.2379 | 0.132 | 0.1939 | 0.1339 | 0.1961 |

*Table 6: SE values for different SNRs*

| SNR | Uni_ZF | Uni_MRT | A_ZF | A_MRT | U_ZF | U_MRT |
|---|---|---|---|---|---|---|
| -30 | 3.5017 | 3.1793 | 3.4958 | 3.1823 | 3.5088 | 3.187 |
| -25 | 3.506 | 3.1835 | 3.5018 | 3.195 | 3.5123 | 3.2087 |
| -20 | 3.513 | 3.189 | 3.5107 | 3.2182 | 3.5145 | 3.2328 |
| -15 | 3.5248 | 3.2002 | 3.5143 | 3.2705 | 3.5222 | 3.2827 |
| -10 | 3.5457 | 3.2245 | 3.5735 | 3.3842 | 3.5687 | 3.3982 |
| -5 | 3.5875 | 3.2642 | 3.7032 | 3.6238 | 3.7053 | 3.6652 |
| 0 | 3.6598 | 3.3298 | 4.0143 | 3.9992 | 4.0273 | 4.0338 |
| 5 | 3.7642 | 3.4217 | 4.4787 | 4.3688 | 4.4867 | 4.3957 |
| 10 | 3.7867 | 3.4742 | 4.8632 | 4.6403 | 4.8692 | 4.6482 |
| 15 | 3.8532 | 3.7563 | 5.0743 | 4.7695 | 5.0832 | 4.7727 |
| 20 | 4.1098 | 4.1732 | 5.1653 | 4.8173 | 5.1598 | 4.8063 |
| 25 | 4.4122 | 4.4487 | 5.1997 | 4.8348 | 5.1853 | 4.8153 |
| 30 | 4.6145 | 4.5727 | 5.2078 | 4.8367 | 5.1963 | 4.8233 |

# Chapter 6 Conclusion and Recommendation

## 6.1 Conclusion

This research thoroughly examined the impact of low-resolution DACs on the performance of downlink MU-MISO systems with 6UEs and 64 base station antennas. By employing a MATLAB-based simulation framework, we analyzed key performance metrics, such as BER, SNR, and SE across various modulation schemes and DAC configurations. The results highlight the trade-offs between power efficiency and performance degradation introduced by low-resolution DACs. Lower-resolution DACs were found to increase quantization noise, adversely affecting BER and SE; however, the use of quantization schemes such as A-law and $\mu$-law played an important role in reducing these effects significantly. In our system, various quantization schemes were evaluated for the 64-QAM modulation scheme, where with uniform quantization, the system performance results were almost the same as the ideal case when no DAC/ADC was used. While with A-law and μ-law, 8-bit performs better and achieves the lowest BER and high SE. This is because μ-law and A-law at higher resolutions (8-bit) improve small-signal precision due to their non-linear compression. For 2-bit resolution, uniform quantization performed better than μ-law and A-law both in terms of BER and SE because non-linear companding at low resolution causes errors for larger amplitudes. The performance of our system was again evaluated with precoders and channel estimation.

The key findings of the performance of ZF and MRT with or without LS channel estimation show that MRT precoding enhances the performance of non-uniform quantization schemes (A-law and μ-law) compared to uniform quantization for both ideal H and estimated H by effectively addressing the limitations of 2-bit DAC resolution and provides improvements in both BER and SE. However, ZF performs better than the MRT precoder due to its ability to eliminate multi-user interference. This study provides valuable insights for designing more energy-efficient wireless systems, especially for future broadband networks, such as 5G and beyond, that demand high spectral efficiency with constrained power budgets. The primary goal of this thesis was to evaluate the impact of low-resolution DACs on downlink MU-MISO system performance. We aimed to analyze how quantization schemes (Uniform, A-law, and $\mu$-law) influence BER, SNR, and spectral efficiency. Additionally, this study explored how channel estimation methods (LS) and precoding techniques (ZF and MRT) could mitigate performance loss. These goals have been achieved, as evidenced by this thesis's simulation results and analysis

## 6.2 Recommendation

Although the study successfully addressed the fundamental challenges of low-resolution DACs in MU-MISO systems, several areas remain open for further exploration, including advanced compensation methods, such as machine learning-based precoding, which could further reduce performance losses associated with low-resolution DACs. Exploring the integration of Reconfigurable Intelligent Surfaces (RIS) to enhance signal quality and compensate for hardware limitations is an exciting research avenue. By addressing these directions, future research can extend this study's findings to develop robust, energy-efficient, and secure wireless communication systems tailored for next-generation networks.

## 6.3 Sustainable Development

This research demonstrates how low-resolution DACs when properly optimized, contribute to reducing the energy demands of large-scale communication systems. By lowering power consumption, these systems can significantly minimize their carbon footprint, supporting global efforts to combat climate change. Furthermore, incorporating advanced precoding and channel estimation techniques ensures that performance is not overly compromised, achieving a balance between sustainability and system efficiency.

## 6.4 Ethical Consideration

Ethical aspects of deploying low-resolution DAC systems merit further study. This includes ensuring equitable access to high-performance communication technologies and addressing potential disparities in service quality. Future work should also investigate the implications of system downgrades on user privacy and security, ensuring that sustainable practices do not unintentionally compromise these essential aspects.

## 6.5 Cost analysis

In this project, the cost is minimal because all simulations are conducted using personal computers and MATLAB software. Since no additional hardware, cloud services, or paid software licenses were required, the financial cost is negligible. However, in larger and more complex projects, cost analysis becomes a critical consideration. For instance, if real-time hardware testing were necessary, expenses would be incurred for devices such as DACs, ADCs, FPGA boards, and specialized signal processing hardware.

# List of References


[1] H. Q. Ngo, *Massive MIMO: Fundamentals and System Designs*, 1st ed. Linkopings Universitet, 2014.

[2] J. Choi, G. Lee, A. Alkhateeb, A. Gatherer, N. Al-Dhahir, and B. L. Evans, "Advanced Receiver Architectures for Millimeter-Wave Communications with Low-Resolution ADCs," *IEEE Commun. Mag.*, vol. 58, no. 8, pp. 42–48, Aug. 2020, doi: 10.1109/MCOM.001.2000122.

[3] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency," *FNT in Signal Processing*, vol. 11, no. 3–4, pp. 154–655, 2017, doi: 10.1561/2000000093.

[4] W. Tan, W. Huang, X. Yang, Z. Shi, W. Liu, and L. Fan, "Multiuser precoding scheme and achievable rate analysis for massive MIMO system," *J Wireless Com Network*, vol. 2018, no. 1, p. 210, Dec. 2018, doi: 10.1186/s13638-018-1223-1.

[5] "Linear precoding with Low-Resolution DACs."

[6] J. Liu, Z. Luo, and X. Xiong, "Low-Resolution ADCs for Wireless Communication: A Comprehensive Survey," *IEEE Access*, vol. 7, pp. 91291–91324, 2019, doi: 10.1109/ACCESS.2019.2927891.

[7] N. Sharma, D. Jain, K. Bhatt, and M. Thomas Themalil, "Performance Comparison of Various Digital Modulation Schemes based on Bit Error Rate under AWGN Channel," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India: IEEE, Apr. 2021, pp. 619–623. doi: 10.1109/ICCMC51019.2021.9418396.

[8] J. Xu, W. Xu, J. Zhu, D. W. K. Ng, and A. Lee Swindlehurst, "Secure Massive MIMO Communication With Low-Resolution DACs," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3265–3278, May 2019, doi: 10.1109/TCOMM.2019.2895023.

[9] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A Deep Learning Framework for Optimization of MISO Downlink Beamforming," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866–1880, Mar. 2020, doi: 10.1109/TCOMM.2019.2960361.

[10] J. Mietzner, R. Schober, L. Lampe, W. Gerstacker, and P. Hoeher, "Multiple-antenna techniques for wireless communications - a comprehensive literature survey," *IEEE Commun. Surv. Tutorials*, vol. 11, no. 2, pp. 87–105, 2009, doi: 10.1109/SURV.2009.090207.

[11] E. Björnson and Ö. T. Demir, *Introduction to Multiple Antenna Communications and Reconfigurable Surfaces*. Now Publishers, 2024. doi: 10.1561/9781638283157.

[12] D. Bala, G. M. Waliullah, M. A. Hena, I. Abdullah, and M. A. Hossain, “Study the Performance of Capacity for SISO, SIMO, MISO and MIMO in Wireless Communication”.

[13] M. Zhou, Y. Zhang, X. Qiao, M. Xie, L. Yang, and H. Zhu, “Multigroup Multicast Downlink Cell-Free Massive MIMO Systems With Multiantenna Users and Low-Resolution ADCs/DACs,” *IEEE Systems Journal*, vol. 16, no. 3, pp. 3578–3589, Sep. 2022, doi: 10.1109/JSYST.2021.3077765.

[14] S. Jacobsson, U. Gustavsson, G. Durisi, and C. Studer, “Massive MU-MIMO-OFDM Uplink with Hardware Impairments: Modeling and Analysis,” Jul. 17, 2019, *arXiv*: arXiv:1812.02078. doi: 10.48550/arXiv.1812.02078.

[15] I. Ahmed *et al.*, “A Survey on Hybrid Beamforming Techniques in 5G: Architecture and System Model Perspectives,” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 3060–3097, 2018, doi: 10.1109/COMST.2018.2843719.

[16] B. Liu, “The perceptual impact of different quantization schemes in G.719”.

[17] Q. Guo, Y. Wang, and X. Cui, “Integer-Only Neural Network Quantization Scheme Based on Shift-Batch-Normalization,” May 28, 2021, *arXiv*: arXiv:2106.00127. doi: 10.48550/arXiv.2106.00127.

[18] Van de Beek, J.-J., Edfors, O., Sandell, M., Wilson, S. K., & Börjesson “Enhanced channel estimation. IEEE - Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/VETEC.1995.504981”

[19] Jae Hong Lee, “Digital transmission through AWGN channel.” Prentice Hall, 2002

[20]“Performance_analysis_of_transmission_modes_in_4G_LTE_LTE-A_downlink_systems.”

[21] B. Haji-saeed, J. Khoury, W. D. Goodhue, and C. L. Woods, “A-law/μ-law Dynamic Range Compression Deconvolution”.

[22] M. Chaitanya and I. S. P. Reddy, “System Performance of MIMO MRT/MRC System under Perfect CSI”.

[23] T. Younas *et al.*, “Performance analysis of ZF and RZF in low-resolution ADC/DAC massive MIMO systems,” *China Commun.*, vol. 21, no. 8, pp. 115–126, Aug. 2024, doi: 10.23919/JCC.ja.2022-0702.

[24] “Channel_Estimation_using_LS_and_MMSE_Channel_Estimation_Techniques_for_MIMO-OFDM_Systems.”

[25] Q. Guo, and Y. Wang “LS channel estimation.” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 3060–3097, 2018, doi: 10.1109/COMST.2018.2843719.

# Appendix A

## MATLAB Code Used

### BER vs SNR Evaluation

```
% Parameters
SNR_dB = -20:2:30; % SNR values in dB
numSymbols = 1e5; % Number of symbols to transmit

% Modulation orders
modulationOrders = [2, 4, 16, 64]; % BPSK, QPSK, 16-QAM, 64-QAM

% Pre-allocate space for BER results
ber_custom = zeros(length(modulationOrders), length(SNR_dB));
ber_theoretical = zeros(length(modulationOrders), length(SNR_dB));

% Loop over each modulation scheme for custom and theoretical BER
for modIdx = 1:length(modulationOrders)
    M = modulationOrders(modIdx); % Modulation order
    k = log2(M); % Bits per symbol

    % Generate random bits (same bits will be used for both custom and
theoretical parts)
    bits = randi([0 1], numSymbols * k, 1);

    % Custom BER Calculation
    % Modulate the bits using custom function
    if M == 2
        % BPSK modulation
        txSymbols_custom = phase1Modulator(bits, 1);
    elseif M == 4
        % QPSK modulation
        txSymbols_custom = phase1Modulator(bits, 2);
    elseif M == 16
        % 16-QAM modulation
        txSymbols_custom = phase1Modulator(bits, 3);
    elseif M == 64
        % 64-QAM modulation
        txSymbols_custom = phase1Modulator(bits, 4);
    end

    % Ensure txSymbols is a column vector for consistency
    if size(txSymbols_custom, 1) == 1
        txSymbols_custom = txSymbols_custom';
    end

    % Theoretical BER Calculation
    % Modulate the bits using MATLAB's built-in function
    if M == 2
        % BPSK modulation
        txSymbols_theoretical = pskmod(bits, M);
    else
```

```
        % QPSK, 16-QAM, 64-QAM modulation
        txSymbols_theoretical    =    qammod(bits,    M,    'InputType',    'bit',
'UnitAveragePower', true);
    end

    % Loop over SNR values for both custom and theoretical BER
    for snrIdx = 1:length(SNR_dB)

        % Custom BER computation
        rxSymbols_custom = awgn(txSymbols_custom, SNR_dB(snrIdx), 'measured');
        if M == 2
            rxBits_custom = phase1Demodulator(rxSymbols_custom, 1);
        elseif M == 4
            rxBits_custom = phase1Demodulator(rxSymbols_custom, 2);
        elseif M == 16
            rxBits_custom = phase1Demodulator(rxSymbols_custom, 3);
        elseif M == 64
            rxBits_custom = phase1Demodulator(rxSymbols_custom, 4);
        end
        if size(rxBits_custom, 1) == 1
            rxBits_custom = rxBits_custom';
        end
        numErrors_custom = sum(bits ~= rxBits_custom);
        ber_custom(modIdx, snrIdx) = numErrors_custom / length(bits);

        % Theoretical BER computation
        rxSymbols_theoretical  =  awgn(txSymbols_theoretical,  SNR_dB(snrIdx),
'measured');
        if M == 2
            rxBits_theoretical = pskdemod(rxSymbols_theoretical, M);
        else
            rxBits_theoretical     =     qamdemod(rxSymbols_theoretical,     M,
'OutputType', 'bit', 'UnitAveragePower', true);
        end
        if size(rxBits_theoretical, 1) == 1
            rxBits_theoretical = rxBits_theoretical';
        end
        numErrors_theoretical = sum(bits ~= rxBits_theoretical);
        ber_theoretical(modIdx, snrIdx) = numErrors_theoretical / length(bits);
    end
end

% Plot the BER vs SNR curves for both custom and theoretical
figure;

% Custom BER plots
semilogy(SNR_dB, ber_custom(1, :), 'b-o', 'DisplayName', 'Custom BPSK');
hold on;
semilogy(SNR_dB, ber_custom(2, :), 'r-o', 'DisplayName', 'Custom QPSK');
semilogy(SNR_dB, ber_custom(3, :), 'g-o', 'DisplayName', 'Custom 16-QAM');
semilogy(SNR_dB, ber_custom(4, :), 'k-o', 'DisplayName', 'Custom 64-QAM');

% Theoretical BER plots
semilogy(SNR_dB,  ber_theoretical(1,  :),  'b--',  'DisplayName',  'Theoretical
BPSK');
```

```
semilogy(SNR_dB, ber_theoretical(2, :), 'r--', 'DisplayName', 'Theoretical QPSK');
semilogy(SNR_dB, ber_theoretical(3, :), 'g--', 'DisplayName', 'Theoretical 16-QAM');
semilogy(SNR_dB, ber_theoretical(4, :), 'k--', 'DisplayName', 'Theoretical 64-QAM');

% Formatting the plot
xlabel('SNR (dB)');
ylabel('Bit Error Rate (BER)');
title('Custom vs Theoretical BER vs SNR for BPSK, QPSK, 16-QAM, and 64-QAM');
legend show;
grid on;
hold off;
```

## DAC Integration

```
% Parameters
SNR_dB = -20:2:30; % SNR values in dB
numSymbols = 1e5;  % Number of symbols to transmit

% Modulation order for 64-QAM
M = 64;
k = log2(M); % Bits per symbol

% DAC and ADC bit resolutions to test
bit_resolutions = [1, 2, 4, 8];
ber = zeros(length(bit_resolutions) + 1, length(SNR_dB)); % +1 for the ideal case

% Generate random bits
bits = randi([0 1], numSymbols * k, 1);

% Loop over each DAC/ADC resolution
for bitIdx = 1:length(bit_resolutions)
    dac_bits = bit_resolutions(bitIdx);
    adc_bits = dac_bits;

    % 64-QAM modulation
    txSymbols = phase1Modulator(bits, 4);

    % Ensure modulator output is a column vector
    if size(txSymbols, 1) == 1
        txSymbols = txSymbols.';
    end

    % Convert to DAC format
    [DACOutput_real, DACOutput_imag] = DAC_function(txSymbols, dac_bits);

    % Loop over SNR values
    for snrIdx = 1:length(SNR_dB)
        % Pass the DAC output through an AWGN channel (only once)
        rxSymbols_real = awgn(DACOutput_real, SNR_dB(snrIdx), 'measured');
        rxSymbols_imag = awgn(DACOutput_imag, SNR_dB(snrIdx), 'measured');
```

```
        % Convert the received symbols back to ADC format
        ADCOutput = ADC_function(rxSymbols_real, rxSymbols_imag, adc_bits);

        % 64-QAM demodulation
        rxBits = phase1Demodulator(ADCOutput, 4);

        % Ensure demodulator output is a column vector
        if size(rxBits, 1) == 1
            rxBits = rxBits';
        end

        % Calculate the number of bit errors
        numErrors = sum(bits ~= rxBits);

        % Compute the BER
        ber(bitIdx, snrIdx) = numErrors / length(bits);
    end
end

% Ideal case: no DAC and ADC quantization
txSymbols_ideal = phase1Modulator(bits, 4);

for snrIdx = 1:length(SNR_dB)
    % Pass the ideal transmitted symbols through an AWGN channel
    rxSymbols_ideal = awgn(txSymbols_ideal, SNR_dB(snrIdx), 'measured');

    % 64-QAM demodulation for ideal case
    rxBits_ideal = phase1Demodulator(rxSymbols_ideal, 4);

    % Ensure demodulator output is a column vector
    if size(rxBits_ideal, 1) == 1
        rxBits_ideal = rxBits_ideal';
    end

    % Calculate the number of bit errors for the ideal case
    numErrors_ideal = sum(bits ~= rxBits_ideal);

    % Compute the BER for the ideal case
    ber(end, snrIdx) = numErrors_ideal / length(bits);
end

% Plot 64-QAM BER vs SNR curves for different DAC/ADC bit resolutions
figure;
markers = {'-o', '-s', '-^', '-d', '-x', '-*'}; % Different markers for each
resolution

for bitIdx = 1:length(bit_resolutions)
    semilogy(SNR_dB,    ber(bitIdx,    :),    markers{bitIdx},    'DisplayName',
sprintf('%d-bit DAC/ADC', bit_resolutions(bitIdx)));
    hold on;
end

% Add the ideal plot (no DAC/ADC quantization)
semilogy(SNR_dB, ber(end, :), '-k', 'LineWidth', 1.5, 'DisplayName', 'Ideal (No
DAC/ADC)');
```

```
% Formatting the plot
xlabel('SNR (dB)');
ylabel('Bit Error Rate (BER)');
title('BER vs SNR for 64-QAM with Different DAC/ADC Bit Resolutions');
legend('Location', 'southwest'); % Move legend to the bottom left
grid on;
```

Precoding Techniques Integrated with Quantization Schemes

```
function precoder_sim_uni_A_U_comp(varargin)
% -- set up default/custom parameters

    if isempty(varargin)
        disp('using default simulation settings and parameters...');
        % set default simulation parameters
        par.U = 6; % number of UEs
        par.B = 64; % number of BS antennas
        par.trials = 1e3; % number of Monte-Carlo trials (transmissions)
        par.relerr = 0; % relative channel estimate error
        par.SNRdB_list = -50:2:10; % list of SNR [dB] values to be simulated
        par.mod        =        '64QAM';        %        modulation        type:
'BPSK','QPSK','8PSK','16QAM','64QAM'
        par.precoder = {"MRT"}; % select precoding scheme(s) to be evaluated
        par.dac = {"uni","a","u"};
        par.bps = 6; % number of bits per symbol
        par.dac_bits = 8;
        par.adc_bits = 8;
    else
        disp('use custom simulation settings and parameters...')
        par = varargin{1}; % load custom simulation parameters
    end

    % initialize result arrays
    res.BER_uni  = zeros(length(par.precoder),length(par.SNRdB_list));
    res.SE_MRT_uni = zeros(length(par.precoder),length(par.SNRdB_list));
    res.BER_A  = zeros(length(par.precoder),length(par.SNRdB_list));
    res.SE_MRT_A = zeros(length(par.precoder),length(par.SNRdB_list));
    res.BER_U  = zeros(length(par.precoder),length(par.SNRdB_list));
    res.SE_MRT_U = zeros(length(par.precoder),length(par.SNRdB_list));

    % -- start simulation
    % generate random bit stream
    b = randi([0 1],par.U,par.bps);

    % trials loop
    for tt=1:par.trials

        for u = 1:par.U
          % generate transmit symbols
            s(u, :) = phase1Modulator(b(u, :),4);
            %normalize the modulator output
            s(u, :) = s(u, :) / 9.98;
        end
```

```
        % generate iid Gaussian channel matrix & noise vector
        n = sqrt(0.5)*(randn(par.U,1)+1i*randn(par.U,1));
        H = sqrt(0.5)*(randn(par.U,par.B)+1i*randn(par.U,par.B));

        % channel estimation error
        if par.relerr > 0
            Hhat              =               sqrt(1-par.relerr)*H              +
sqrt(par.relerr/2)*(randn(par.U,par.B)+1i*randn(par.U,par.B));
        else
            Hhat = H;
        end

        % algorithm loop
        for pp=1:length(par.precoder)

            % noise-independent precoders
            switch (par.precoder{pp})
                case 'MRT_inf' % MRT precoding (inf. res.)
                    [x, beta, ~] = MRT(s,Hhat);
                case 'MRT' % MRT precoding (quantized)
                    [z, beta, ~] = MRT(s,Hhat);
                    [x,~] = DAC_function_uni_precoding(z.',par.dac_bits);
                    x=x.';
                case 'ZF_inf' % ZF precoding (inf. res.)
                    [x, beta] = ZF(s,Hhat);
                case 'ZF' % ZF precoding (quantized)
                    [z, beta] = ZF(s, Hhat);
                    [x,~] = DAC_function_uni_precoding(z.',par.dac_bits);
                    x=x.';
            end

            % SNR loop
            for k=1:length(par.SNRdB_list)

                % set noise variance
                N0 = 10.^(-par.SNRdB_list(k)/10);

                % transmit data over noisy channel
                Hx = H*x;
                y = Hx + sqrt(N0)*n;

                % pass received signal through ADC
                switch (par.precoder{pp})
                     case {'MRT', 'ZF'}
                        y = ADC_function_uni_precoding(y.',par.adc_bits);
                        y=y.';
                 end

                % scale received signal at the UEs (not needed for PSK)
                shat = beta*y;

                % demodulation of scaled received signal
                for u = 1:par.U
                    %de-normalize the received signal before passing to
                    %demodulator
                    shat(u, :) = shat(u, :) * 9.98;
```

```
                bhat(u, :) = phase1Demodulator(shat(u, :),4);
            end

            % BER calculation
            res.BER_uni(pp,k)          =          res.BER_uni(pp,k)          +
sum(sum(b~=bhat))/(par.U*par.bps); % bit error rate

        end % SNR loop

    end % algorithm loop

end % trials loop

% normalize results
res.BER_uni = res.BER_uni/par.trials;
res.SE_MRT_uni = log2(64)*(1-res.BER_uni);


% trials loop
for tt=1:par.trials

    for u = 1:par.U
      % generate transmit symbols
        s(u, :) = phase1Modulator(b(u, :),4);
        %normalize the modulator output
        s(u, :) = s(u, :) / 9.98;
    end

    % generate iid Gaussian channel matrix & noise vector
    n = sqrt(0.5)*(randn(par.U,1)+1i*randn(par.U,1));
    H = sqrt(0.5)*(randn(par.U,par.B)+1i*randn(par.U,par.B));

    % channel estimation error
    if par.relerr > 0
        Hhat             =              sqrt(1-par.relerr)*H             +
sqrt(par.relerr/2)*(randn(par.U,par.B)+1i*randn(par.U,par.B));
    else
        Hhat = H;
    end

    % algorithm loop
    for pp=1:length(par.precoder)

        % noise-independent precoders
        switch (par.precoder{pp})
            case 'MRT_inf' % MRT precoding (inf. res.)
                [x, beta, ~] = MRT(s,Hhat);
            case 'MRT' % MRT precoding (quantized)
                [z, beta, ~] = MRT(s,Hhat);
                [x,max_value]                                                  =
DAC_function_A_law_precoding(z,par.dac_bits);

            case 'ZF_inf' % ZF precoding (inf. res.)
                [x, beta] = ZF(s,Hhat);
            case 'ZF' % ZF precoding (quantized)
                [z, beta] = ZF(s, Hhat);
```

```
                    [x,max_value]                                         =
DAC_function_A_law_precoding(z,par.dac_bits);

            end

            % SNR loop
            for k=1:length(par.SNRdB_list)

                % set noise variance
                N0 = 10.^(-par.SNRdB_list(k)/10);

                % transmit data over noisy channel
                Hx = H*x;
                y = Hx + sqrt(N0)*n;

                % pass received signal through ADC
                switch (par.precoder{pp})
                     case {'MRT', 'ZF'}
                        y     =     ADC_function_A_law_precoding(y,par.adc_bits,
max_value);

                 end

                % scale received signal at the UEs (not needed for PSK)
                shat = beta*y;

                % demodulation of scaled received signal
                for u = 1:par.U
                    %de-normalize the received signal before passing to
                    %demodulator
                    shat(u, :) = shat(u, :) * 9.98;
                    bhat(u, :) = phase1Demodulator(shat(u, :),4);
                end

                % BER calculation
                res.BER_A(pp,k)          =          res.BER_A(pp,k)          +
sum(sum(b~=bhat))/(par.U*par.bps); % bit error rate

            end % SNR loop

        end % algorithm loop

    end % trials loop

    % normalize results
    res.BER_A = res.BER_A/par.trials;
    res.SE_MRT_A = log2(64)*(1-res.BER_A);

    % trials loop
    for tt=1:par.trials

        for u = 1:par.U
          % generate transmit symbols
            s(u, :) = phase1Modulator(b(u, :),4);
            %normalize the modulator output
            s(u, :) = s(u, :) / 9.98;
```

```
        end

        % generate iid Gaussian channel matrix & noise vector
        n = sqrt(0.5)*(randn(par.U,1)+1i*randn(par.U,1));
        H = sqrt(0.5)*(randn(par.U,par.B)+1i*randn(par.U,par.B));

        % channel estimation error
        if par.relerr > 0
            Hhat             =             sqrt(1-par.relerr)*H             +
sqrt(par.relerr/2)*(randn(par.U,par.B)+1i*randn(par.U,par.B));
        else
            Hhat = H;
        end

        % algorithm loop
        for pp=1:length(par.precoder)

            % noise-independent precoders
            switch (par.precoder{pp})
                case 'MRT_inf' % MRT precoding (inf. res.)
                    [x, beta, ~] = MRT(s,Hhat);
                case 'MRT' % MRT precoding (quantized)
                    [z, beta, ~] = MRT(s,Hhat);
                    [x,max_value]                                           =
DAC_function_U_law_precoding(z,par.dac_bits);

                case 'ZF_inf' % ZF precoding (inf. res.)
                    [x, beta] = ZF(s,Hhat);
                case 'ZF' % ZF precoding (quantized)
                    [z, beta] = ZF(s, Hhat);
                    [x,max_value]                                           =
DAC_function_U_law_precoding(z,par.dac_bits);

            end

            % SNR loop
            for k=1:length(par.SNRdB_list)

                % set noise variance
                N0 = 10.^(-par.SNRdB_list(k)/10);

                % transmit data over noisy channel
                Hx = H*x;
                y = Hx + sqrt(N0)*n;

                % pass received signal through ADC
                switch (par.precoder{pp})
                     case {'MRT', 'ZF'}
                        y    =    ADC_function_U_law_precoding(y,par.adc_bits,
max_value);

                 end

                % scale received signal at the UEs (not needed for PSK)
                shat = beta*y;
```

```
                % demodulation of scaled received signal
                for u = 1:par.U
                    %de-normalize the received signal before passing to
                    %demodulator
                    shat(u, :) = shat(u, :) * 9.98;
                    bhat(u, :) = phase1Demodulator(shat(u, :),4);
                end

                % BER calculation
                res.BER_U(pp,k)          =          res.BER_U(pp,k)          +
sum(sum(b~=bhat))/(par.U*par.bps); % bit error rate

            end % SNR loop

        end % algorithm loop

    end % trials loop

    % normalize results
    res.BER_U = res.BER_U/par.trials;
    res.SE_MRT_U = log2(64)*(1-res.BER_U);

% -- show results
    % marker style and color
    marker_style = {'o-','s--','v-.','+:','<-','>--','x-.','^:','*-','d--','h-
.','p:'};
    marker_color = [...
        0.0000    0.4470    0.7410;...
        0.8500    0.3250    0.0980;...
        0.9290    0.6940    0.1250;...
        0.4940    0.1840    0.5560;...
        0.4660    0.6740    0.1880;...
        0.3010    0.7450    0.9330;...
        0.6350    0.0780    0.1840;...
        0.7500    0.7500    0.0000;...
        0.7500    0.0000    0.7500;...
        0.0000    0.5000    0.0000;...
        0.0000    0.0000    1.0000;...
        1.0000    0.0000    0.0000];

    % legends
    precoder_legend = par.dac;
    for pp = 1:length(par.dac)
        if strcmpi(precoder_legend{pp}, 'uni')
            precoder_legend{pp} =  'Uniform';
        elseif strcmpi(precoder_legend{pp}, 'a')
            precoder_legend{pp} = 'A-law';
        elseif strcmpi(precoder_legend{pp}, 'u')
            precoder_legend{pp} = 'MU-law';
        end
    end

    % Create a figure for BER and SE subplots
figure('name', 'BER and SE Subplots');

% Subplot 1: BER
```

```
subplot(2, 1, 1); % Two rows, one column, first plot
hold on;
for pp = 1:length(par.precoder)
    semilogy(par.SNRdB_list+20, res.BER_uni(pp,:), marker_style{pp}, 'color', marker_color(pp,:), 'LineWidth', 2);
    semilogy(par.SNRdB_list+20, res.BER_A(pp,:), marker_style{pp+3}, 'color', marker_color(pp+5,:), 'LineWidth', 2);
    semilogy(par.SNRdB_list+20, res.BER_U(pp,:), marker_style{pp+5}, 'color', marker_color(pp+7,:), 'LineWidth', 2);

end
grid on; box on;
xlabel('SNR [dB]', 'FontSize', 12);
ylabel('Bit Error Rate (BER)', 'FontSize', 12);
set(gca, 'YScale', 'log');
legend(precoder_legend, 'FontSize', 10, 'location', 'southwest');
set(gca, 'FontSize', 12);
if length(par.SNRdB_list) > 1
     axis([min(par.SNRdB_list+20) max(par.SNRdB_list+20) 1e-2 1]);
end
title(['BER vs SNR - ' par.precoder{1}]);

% Subplot 2: SE
subplot(2, 1, 2); % Two rows, one column, second plot
hold on;
for pp = 1:length(par.precoder)
    semilogy(par.SNRdB_list+20,     res.SE_MRT_uni(pp,:),     marker_style{pp}, 'color', marker_color(pp,:), 'LineWidth', 2);
    semilogy(par.SNRdB_list+20,     res.SE_MRT_A(pp,:),     marker_style{pp+3}, 'color', marker_color(pp+5,:), 'LineWidth', 2);
    semilogy(par.SNRdB_list+20,     res.SE_MRT_U(pp,:),     marker_style{pp+5}, 'color', marker_color(pp+7,:), 'LineWidth', 2);

end
grid on; box on;
xlabel('SNR [dB]', 'FontSize', 12);
ylabel('Spectral Efficiency (SE)', 'FontSize', 12);
legend(precoder_legend, 'FontSize', 10, 'location', 'southwest');
set(gca, 'FontSize', 12);
if length(par.SNRdB_list) > 1
     axis([min(par.SNRdB_list+20) max(par.SNRdB_list+20) 2 6]);
end
title(['SE vs SNR - ' par.precoder{1}]);

fprintf('\nsimulation has finished!\n\n');

end
```