**QPSK Communication Chain Simulation with BER Analysis**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**

In

**Electronics and Communication Engineering**

**School of Engineering and Sciences**

Submitted by

**P. Sudeep Reddy(AP22110020104)**

**A picture containing text

Description automatically generated**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**Abstract:**

This project simulates a complete Quadrature Phase Shift Keying (QPSK) digital communication chain, including bit generation, modulation, AWGN channel modeling, demodulation, and Bit Error Rate (BER) analysis. The aim is to understand the effects of channel noise on digital signals and evaluate system performance using BER vs. SNR plots and constellation diagrams. This project bridges core Signals & Systems theory with practical communication systems and provides foundational insight for both software simulation and potential hardware (RTL) implementation.

**Introduction:**

Modern digital communication systems rely heavily on signal modulation schemes to transmit data over noisy environments. QPSK is a bandwidth-efficient modulation technique where 2 bits are mapped per symbol. This project models a simple transmitter–channel–receiver pipeline and evaluates the impact of noise using a Monte Carlo simulation approach.

**Methodology:**

1. **Bit Generation:**

A stream of 10,000 binary bits is generated randomly to simulate digital data.

This represents a discrete-time signal (x[n]).

1. **QPSK Modulation:**

Bits are grouped into pairs (2 bits = 1 symbol).

Each pair is mapped to one of four constellation points:

00 → (1 + j1)/√2

01 → (−1 + j1)/√2

11 → (−1 − j1)/√2

10 → (1 − j1)/√2

1. **Channel Modeling (AWGN):**

Additive White Gaussian Noise is added to each complex symbol.

Noise is zero-mean Gaussian with variance calculated based on target SNR (0–10 dB).

1. **Demodulation:**

Received noisy symbols are decoded using quadrant detection.

Real and imaginary parts determine the bits.

1. **BER Analysis:**

Compare received bits to original bits to count errors.

Calculate Bit Error Rate (BER) for various SNR levels.

Plot BER vs. SNR curve.

**Features:**

* QPSK Modulation & Demodulation
* AWGN Channel Simulation
* Bit Error Rate (BER) Calculation
* Constellation Plot Visualization
* BER vs SNR Performance Curve

**Concepts Covered:**

* Sampling & Discrete Signals
* Digital Modulation (QPSK)
* Noise Models (AWGN)
* Bit Error Rate Analysis
* Signal Detection in Complex Plane

**Tools Used:**

* Python (NumPy, Matplotlib) / MATLAB
* Jupyter Notebook / MATLAB Scripts

**MATLAB Version:**

clc;

clear;

% Step 1: Generate random bits

N = 10000;

data\_bits = randi([0 1], N, 1);

% Step 2: QPSK Modulation

data\_reshaped = reshape(data\_bits, [], 2);

symbols = zeros(size(data\_reshaped,1), 1);

for i = 1:size(data\_reshaped,1)

b = data\_reshaped(i,:);

if isequal(b, [0 0])

symbols(i) = (1 + 1i) / sqrt(2);

elseif isequal(b, [0 1])

symbols(i) = (-1 + 1i) / sqrt(2);

elseif isequal(b, [1 1])

symbols(i) = (-1 - 1i) / sqrt(2);

elseif isequal(b, [1 0])

symbols(i) = (1 - 1i) / sqrt(2);

end

end

% Step 3-5: SNR loop and BER

snr\_dB = 0:1:10;

ber = zeros(size(snr\_dB));

for k = 1:length(snr\_dB)

snr\_linear = 10^(snr\_dB(k)/10);

noise\_power = 1 / snr\_linear;

noise = sqrt(noise\_power/2) \* (randn(size(symbols)) + 1i\*randn(size(symbols)));

rx = symbols + noise;

rx\_bits = zeros(size(data\_bits));

for i = 1:length(rx)

real\_part = real(rx(i));

imag\_part = imag(rx(i));

if real\_part > 0

rx\_bits(2\*i - 1) = 0;

else

rx\_bits(2\*i - 1) = 1;

end

if imag\_part > 0

rx\_bits(2\*i) = 0;

else

rx\_bits(2\*i) = 1;

end

end

errors = sum(rx\_bits ~= data\_bits);

ber(k) = errors / N;

end

% Plot BER vs SNR

semilogy(snr\_dB, ber, '-o');

xlabel('SNR (dB)');

ylabel('Bit Error Rate (BER)');

title('QPSK BER vs SNR');

grid on;

% Plot Constellation for a sample SNR

snr\_sample = 5;

snr\_linear = 10^(snr\_sample/10);

noise\_power = 1 / snr\_linear;

noise = sqrt(noise\_power/2) \* (randn(size(symbols)) + 1i\*randn(size(symbols)));

rx\_sample = symbols + noise;

figure;

plot(real(rx\_sample), imag(rx\_sample), '.');

title(['QPSK Constellation at SNR = ', num2str(snr\_sample), ' dB']);

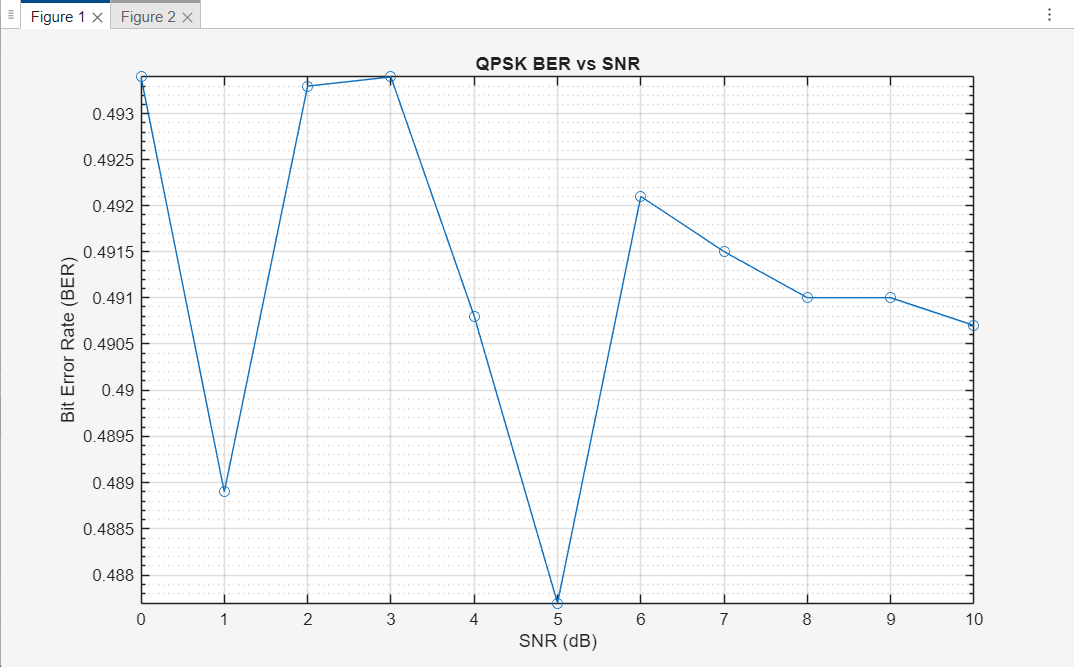
xlabel('In-phase');

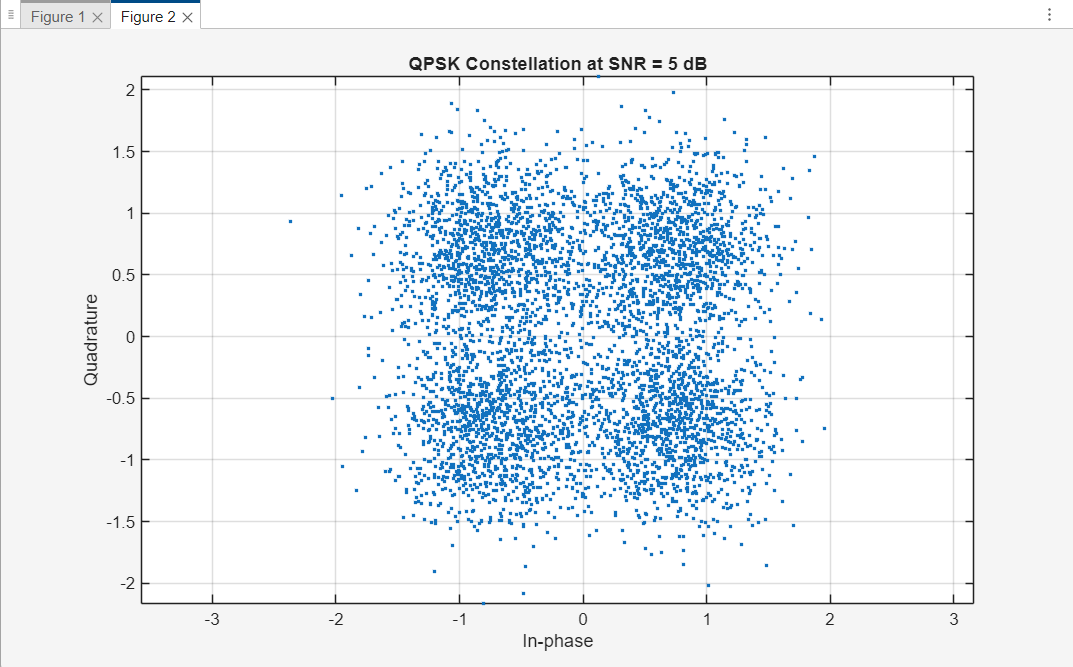
ylabel('Quadrature');

grid on;

axis equal;

**Results:**





**Conclusion:**

This project demonstrates how core Signals & Systems concepts apply directly to digital communication systems. By simulating a complete QPSK system, we explored how noise affects signal integrity and how system performance can be quantified using BER. This forms a strong foundation for both theoretical understanding and future hardware implementation.