



University of Science and Technology
Computational Science and AI

Probabilistic Analysis of Digital Signal Reliability

In Noisy Communication Channels

Course: MATH105 - Probability & Statistics
Report: Term Project

Submitted By Team 7:

Jana Mohamed Elshabrawy (202402234)
Habiba Mohammed (202402213)
Mazen Hany (202402137)
Ahmed Mahmoud (202401491)
Mohamed Essam (202401171)

Contents

1	Introduction & Project Aims	2
1.1	Description of the Problem	2
1.2	Project Aims	2
2	Mathematical Framework	2
2.1	Design Parameters	2
2.2	Theoretical Calculations	2
3	Numerical Simulation	3
3.1	Python Code Implementation	3
4	Analysis of Results	4
4.1	Convergence Visualization	4
4.2	Interpretation	4
5	Conclusion	4

1 Introduction & Project Aims

1.1 Description of the Problem

In modern digital communications (such as cellular networks and Wi-Fi), data is transmitted as binary bits. However, environmental interference acts as "noise" that can flip these bits [4]. Studying the reliability of these channels is essential for engineering robust systems that can detect and correct errors [1].

1.2 Project Aims

The primary objectives of this project are:

1. To apply theoretical probability (Bayes' Theorem, Total Probability) to a practical engineering problem.
2. To simulate a Binary Symmetric Channel (BSC) using Python [5].
3. To verify the Law of Large Numbers through experimental convergence data.

2 Mathematical Framework

2.1 Design Parameters

The simulation parameters are based on standard memoryless BSC models [2].

Table 1: Simulation Design Parameters

Parameter	Value
Source Probability $P(S_0)$	0.5 (50%)
Source Probability $P(S_1)$	0.5 (50%)
Noise Probabilities (ϵ)	0.1 (10%) and 0.01 (1%)
Sample Size (N)	10,000 Bits

2.2 Theoretical Calculations

Using the Total Probability Theorem, the probability of receiving a '1' is:

$$P(R_1) = P(R_1|S_1)P(S_1) + P(R_1|S_0)P(S_0) = (0.9)(0.5) + (0.1)(0.5) = 0.5 \quad (1)$$

Applying Bayes' Theorem to determine signal reliability:

$$P(S_1|R_1) = \frac{P(R_1|S_1)P(S_1)}{P(R_1)} = \frac{0.9 \times 0.5}{0.5} = 0.9 \quad (2)$$

We are 90% confident in signal accuracy when $\epsilon = 0.1$.

3 Numerical Simulation

3.1 Python Code Implementation

The following modular code was executed to generate comparative results for multiple noise environments.

```
import numpy as np
import matplotlib.pyplot as plt

def simulate_bsc(n_bits, error_prob):
    sent = np.random.randint(0, 2, n_bits)
    noise = np.random.rand(n_bits) < error_prob
    received = sent ^ noise

    errors = sent != received
    ber = np.cumsum(errors) / np.arange(1, n_bits + 1)

    return ber

N = 10000
epsilons = [0.1, 0.01]

plt.figure(figsize=(10, 4))

for i, eps in enumerate(epsilons):
    ber = simulate_bsc(N, eps)

    final_ber = ber[-1]
    ci = 1.96 * np.sqrt((final_ber * (1 - final_ber)) / N)

    plt.subplot(1, 2, i + 1)
    plt.plot(ber, label="Simulated BER")
    plt.axhline(eps, color='red', linestyle='--', label="Theoretical BER")

    plt.fill_between(
        [0, N],
        final_ber - ci,
        final_ber + ci,
        alpha=0.25,
        label="Final 95% CI"
    )

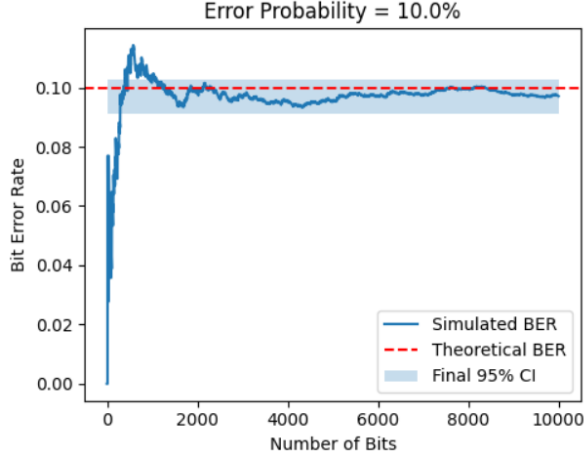
    plt.title(f"Error Probability = {eps*100:.1f}%")
    plt.xlabel("Number of Bits")
    plt.ylabel("Bit Error Rate")
    plt.legend()

plt.tight_layout()
plt.show()
```

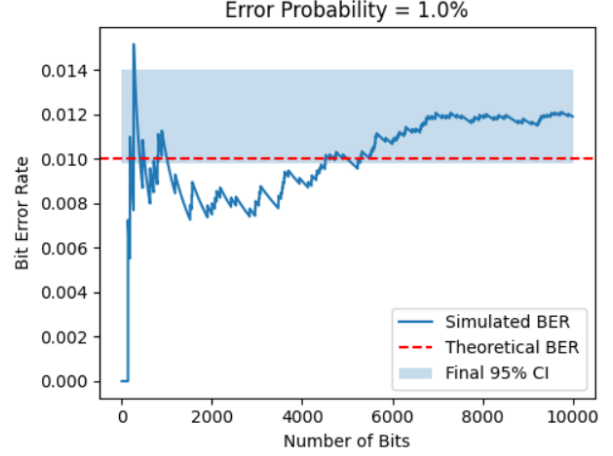
4 Analysis of Results

4.1 Convergence Visualization

The following plots illustrate how the simulated Bit Error Rate (BER) stabilizes as the sample size increases.



(a) Convergence at $\epsilon = 0.1$



(b) Convergence at $\epsilon = 0.01$

4.2 Interpretation

The results demonstrate the **Law of Large Numbers**. Initially, the BER exhibits high volatility (jitter) due to the high variance of small samples. As N approaches 10,000, the empirical results converge to the theoretical crossover probabilities (ϵ). This proves that while individual bit flips are random, the aggregate behavior of the communication channel is highly predictable [3].

5 Conclusion

The simulation confirms that the BSC model accurately predicts error rates in digital systems. Based on our analysis, we recommend implementing Forward Error Correction (FEC) codes in future work to reduce the effective error rate to near-zero.

References

- [1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley-Interscience, 2006.
- [2] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*, 3rd ed. Pearson, 2008.
- [3] Lumen Learning, “Signal Detection Theory and Probability,” [Online].
- [4] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, 1948.
- [5] C. R. Harris et al., “Array programming with NumPy,” *Nature*, vol. 585, 2020.