O.E.L REPORT

TELECOMMUNICATIONS ENGINEERING

Course Code	EF-101
Course Name	FUNDAMENTAL OF IT APPLICATIONS
Semester	1st
Year	2025
Batch	FETC 2025
Lab Instructor Name	DR. AMIR ZEB
Submission Deadline	24-NOV-2025
Students' Name	ESHAAL RAIS
Roll Number	TC-47
Section	B

ABSTRACT

This laboratory report presents an extensive analysis and implementation of a Python based mini project developed to evaluate and compare the performance of multiple communication links. The core objective of this lab session is to integrate fundamental telecommunication principles with practical programming concepts. The project focuses on calculating Signal-to-Noise Ratio (SNR) and Performance Index for several links using signal power, noise power, and data rate as primary parameters.

Through this experiment, students learn to simulate real-world telecommunication systems in software, understand the impact of noise and signal quality on communication efficiency, and develop logical algorithms to process numerical data. The results provide insight into which communication link provides optimal performance, enhancing understanding of system optimization, decision-making, and engineering problem-solving.

TABLE OF CONTENTS

1)INTRODUCTION

Telecommunication is an essential component of modern digital infrastructure, facilitating the exchange of data, voice, and multimedia information across the globe. The quality and reliability of transmitted signals are influenced by several factors including signal strength, noise interference, channel characteristics, and data transmission rate. In professional telecommunication systems, engineers continuously monitor these parameters to ensure optimal performance and minimize errors.

This laboratory session focuses on using Python programming to simulate and analyze communication links. By modeling the behavior of signals and noise mathematically, students gain practical experience in evaluating link performance. The experiment bridges theoretical telecommunication concepts such as signal integrity, noise analysis, and performance measurement with computational problem-solving skills.

Through this lab, students will not only learn to calculate essential performance metrics but also understand how these metrics guide the design and optimization of communication networks. The integration of software-based simulation enhances learning by allowing experimentation with multiple scenarios without the need for physical hardware.

2)LITERATURE REVIEW

Historically, telecommunication performance evaluation was conducted using physical instruments such as oscilloscopes, spectrum analyzers, and signal generators. These devices provided accurate measurements but were limited in scalability and flexibility. With the advancement of programming and simulation tools, engineers now employ software like MATLAB, Python, and Simulink to model communication systems and perform extensive performance analysis.

Signal-to-Noise Ratio (SNR) remains one of the most critical parameters in determining communication quality. High SNR ensures lower probability of bit errors and high fidelity of transmitted information. Performance Index, which integrates SNR and data rate, serves as a composite metric for evaluating overall efficiency of a communication link. Modern academic and industrial practices emphasize using Python for its simplicity, powerful mathematical libraries, and ability to handle complex data processing tasks.

By using Python, students can experiment with multiple communication scenarios, simulate varying signal and noise conditions, and perform comparative analysis to identify optimal links. This approach also fosters analytical thinking and deep understanding of telecommunication principles.

3)SYSTEM MODEL

The system model comprises multiple communication links characterized by:

- Signal Power (Watts)
- Noise Power (Watts)
- Data Rate (kbps)

Processing Steps:

1. Accept input values for each link.
2. Compute SNR for each link.
3. Calculate Performance Index for each link.
4. Compare performance indices.
5. Identify the best-performing link.
6. Display results clearly in tabular form.

This model closely represents real-world evaluation processes in telecommunication networks, where engineers assess link quality based on signal clarity, noise levels, and data transmission capabilities.

SIGNAL-TO-NOISE RATIO (SNR)

SNR = Signal Power / Noise Power

- High SNR → clear signal, low errors
- Low SNR → distorted signal, high errors

SNR is essential for evaluating communication link performance.

PERFORMANCE INDEX

Performance Index = SNR × Data Rate

This combines signal clarity and transmission speed. Higher values indicate more efficient communication.

QUALITY CLASSIFICATION

- Excellent: SNR > 15
- Good: 8 < SNR ≤ 15
- Fair: 3 < SNR ≤ 8
- Poor: SNR ≤ 3

This helps in selecting reliable communication links for critical applications.

4)METHODOLOGY

The methodology provides a step-by-step explanation of the process:

1. **Input Number of Links:**
    - The user specifies how many links to evaluate, making the program dynamic.
2. **Input Link Parameters:**
    - Users enter Signal Power, Noise Power, and Data Rate for each link.
    - These parameters simulate real-world communication link conditions.
3. **Calculate SNR:**
    - Using SNR = Signal Power / Noise Power, the program evaluates signal clarity.
4. **Compute Performance Index:**
    - Performance Index = SNR × Data Rate, combining signal quality and transmission speed.
5. **Assess Quality:**
    - Links are categorized as Excellent, Good, Fair, or Poor based on SNR.
6. **Store Data:**
    - All link information is stored in a structured list/dictionary for comparison and tabular display.
7. **Compare Links:**
    - The link with the highest Performance Index is identified as the best-performing link.
8. **Display Results:**
    - A formatted table summarizes link parameters, SNR, Performance Index, and quality.

5)PYTHON PROGRAM

```python
from prettytable import PrettyTable

def calculate_snr(signal_power, noise_power):
    return signal_power / noise_power

def calculate_performance_index(snr, data_rate):
    return snr * data_rate

print("=== TELECOMMUNICATION LINK PERFORMANCE SYSTEM ===\n")

num_links = int(input("Enter number of communication links to evaluate: "))
links_data = []

for i in range(1, num_links + 1):
    print(f"\n--- Enter details for Link {i} ---")
    signal_power = float(input("Enter Signal Power (Watts): "))
    noise_power = float(input("Enter Noise Power (Watts): "))
    data_rate = float(input("Enter Data Rate (kbps): "))

    snr = calculate_snr(signal_power, noise_power)
    performance_index = calculate_performance_index(snr, data_rate)

    if snr > 15:
        quality = "Excellent"
    elif snr > 8:
        quality = "Good"
    elif snr > 3:
        quality = "Fair"
    else:
        quality = "Poor"

    links_data.append({
        "link_number": i,
        "signal_power": signal_power,
        "noise_power": noise_power,
        "data_rate": data_rate,
        "snr": snr,
        "performance_index": performance_index,
        "quality": quality
    })

# Display table summary
table = PrettyTable()
table.field_names = ["Link", "Signal Power (W)", "Noise Power (W)", "Data Rate (kbps)", "SNR", "Performance", "Quality"]

best_link = None
max_performance = 0

for link in links_data:
    table.add_row([
        link['link_number'],
        link['signal_power'],
        link['noise_power'],
        link['data_rate'],
        f"{link['snr']:.2f}",
        f"{link['performance_index']:.2f}",
        link['quality']
    ])
    if link['performance_index'] > max_performance:
        max_performance = link['performance_index']
        best_link = link['link_number']

print("\n=== LINK PERFORMANCE SUMMARY ===")
print(table)
print(f"\n>>> The BEST link is Link {best_link} with Performance Index = {max_performance:.2f} <<<")
```

6)OUTPUT

```
=== TELECOMMUNICATION LINK PERFORMANCE SYSTEM ===

Enter number of communication links to evaluate: 3

--- Enter details for Link 1 ---
Enter Signal Power (Watts): 12
Enter Noise Power (Watts): 34
Enter Data Rate (kbps): 56

--- Enter details for Link 2 ---
Enter Signal Power (Watts): 1
Enter Noise Power (Watts): 45
Enter Data Rate (kbps): 76

--- Enter details for Link 3 ---
Enter Signal Power (Watts): 23
Enter Noise Power (Watts): 345
Enter Data Rate (kbps): 465

=== LINK PERFORMANCE SUMMARY ===
+------+------------------+-----------------+------------------+------+-------------+---------+
| Link | Signal Power (W) | Noise Power (W) | Data Rate (kbps) | SNR  | Performance | Quality |
+------+------------------+-----------------+------------------+------+-------------+---------+
|  1   |       12.0       |       34.0      |       56.0       | 0.35 |    19.76    |   Poor  |
|  2   |       1.0        |       45.0      |       76.0       | 0.02 |     1.69    |   Poor  |
|  3   |       23.0       |      345.0      |      465.0       | 0.07 |    31.00    |   Poor  |
+------+------------------+-----------------+------------------+------+-------------+---------+

>>> The BEST link is Link 3 with Performance Index = 31.00 <<<
```

7)CHALLENGES

- Selecting realistic signal and noise values.
- Ensuring floating-point precision.
- Structuring data and output neatly.
- Understanding noise impact on performance.
- Comparing multiple links logically and accurately

8)CONCLUSION

The lab demonstrates the application of Python to simulate telecommunication link performance. Students learned to:

- Calculate SNR and Performance Index
- Classify link quality
- Compare multiple links and identify the best
- Understand practical implications of signal and noise

The experiment enhances computational, analytical, and telecommunication engineering skills.

9)REFERENCES

1. Simon Haykin – Communication Systems
2. Digital Signal Processing Fundamentals
3. Python Documentation – python.org
4. ITFA Course Manual
5. Telecommunication Engineering Principles