LDPC CODES FOR ENHANCED RELIABILITY IN 5G NR NETWORKS

Mid-Review 1

AY 2021-25

GITAM (Deemed-to-be) University

Major Project
Project ID: C3

Department of Electrical Electronics and Communication Engineering

Project Team:

- DWARAKANATH [0090]
- SIVA CHANRAN [400]
- HEMANTH KUMAR [457]

Project Mentor:
- DR AMBAR BAJPAI

Project In-charge:
- DR ARUN KUMAR

AGENDA

- OBJECTIVE AND GOALS
- GHANT CHART
- ABOUT LDPC
- FLOW CHART
- ML MODELS
- SIMULATION
- CONCLUSION

Objective and Goals

Phase 1: LDPC Implementation

✓ Implement LDPC Encoding & Decoding – Add redundancy to data (encoding) and use it to correct errors (decoding).

Phase 2: LDPC Performance Testing & ML-Based Improvement

✓ Test LDPC Codes in Noisy Environments – Analyze LDPC performance under Rayleigh (dense areas) and Rician (open areas) fading channels.

✓ Use Machine Learning to Improve Signals – Apply a regression-based ML model to remove noise and enhance signal quality.

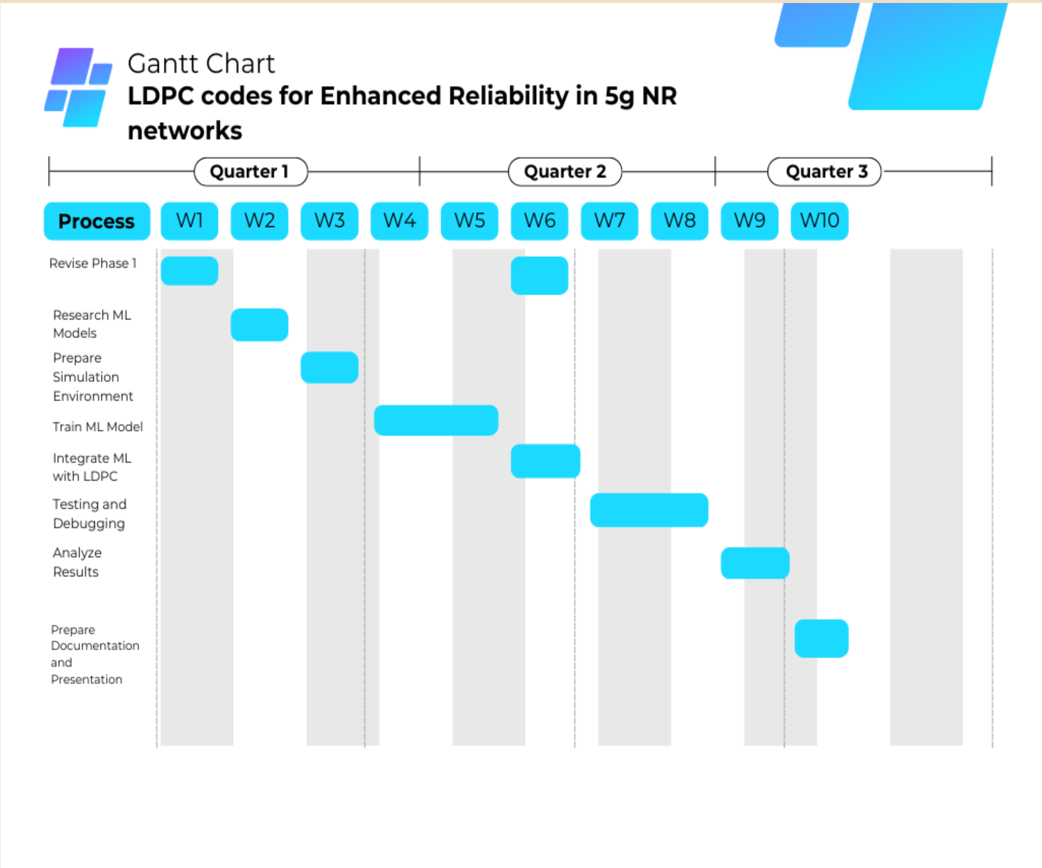
✓ Simulate & Fix Noisy Communication – Transmit encoded messages through noisy channels, apply ML for noise reduction, and recover the original message.

✓ Prove Better Performance – Show that combining LDPC with ML improves communication reliability.

Project Plan (Clearly mention milestone for objectives under each reviews)

Gant Chart - Milestones and Activities

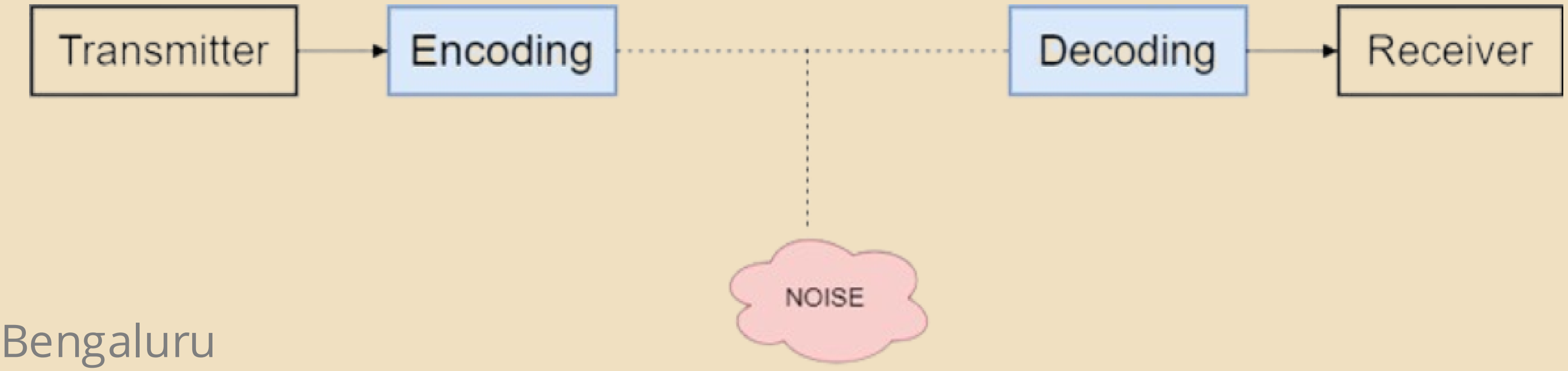
Resources : https://www.officetimeline.com/gantt-chart/how-to-make/excel & https://www.teamgantt.com/



WEEK	TASK
1	Revise Phase 1: Review and refine the implemented LDPC encoding and decoding in MATLAB
2	Research ML Models: Explore machine learning models suitable for channel estimation.
3	Prepare Simulation Environment: Set up MATLAB for integrating ML models and simulate Rayleigh and Rician channels.
4-5	Train ML Model: Train the ML model to map noisy signals to clean signals.
6	Integrate ML with LDPC: Combine the trained ML model with the LDPC system.
7-8	Testing and Debugging: Test the integrated system under various noise conditions and fix issues.
9	Analyze Results: Compare the performance of LDPC with and without ML in terms of accuracy and reliability.
10	Prepare Documentation and Presentation: Write the final report, create a presentation, and prepare for submission.

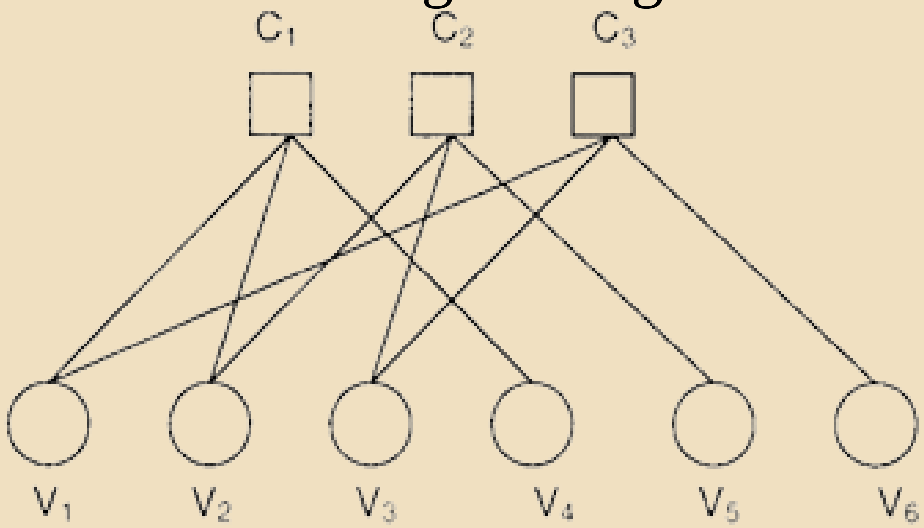
ABOUT LDPC:

- **Low Density parity check matrix**
- **Low density**: The number of “1s” in a matrix used for encoding is small compared to the number of “0s”.This matrix helps in error correction and having fewer “1s”.
- **Parity check**: Parity refers to checking whether the number of “1s” in a set of bits is odd or even.it helps in detecting and correcting errors in data.
- **Definition**: LDPC (Low-Density Parity-Check) is a type of error-correcting code used to detect and fix mistakes in data transmission. It works by adding extra bits to the original data, which helps in checking if any errors occurred during transmission. The "low-density" part means that the connections between these extra bits and the original data are sparse, making the code efficient and fast to decode.



KEY TERMS

- **Tanner graph**: A Tanner graph is a special type of graph used to represent error-correcting codes, like LDPC codes. It is made up of two types of nodes: variable nodes (representing data bits) and check nodes (representing parity checks). The connections (edges) between these nodes show how the data bits are checked for errors.
- **Sparse matrix**: A sparse matrix is a matrix (a grid of numbers) where most of the elements are zero. Only a small number of the entries have non-zero values.
- **Generative matrix**: A generator matrix is a special matrix used in coding theory to create codewords (which are the data plus extra bits for error correction) from the original data bits. It defines how to take the original data and turn it into a longer string of bits that can be sent or stored with error-correction capabilities.

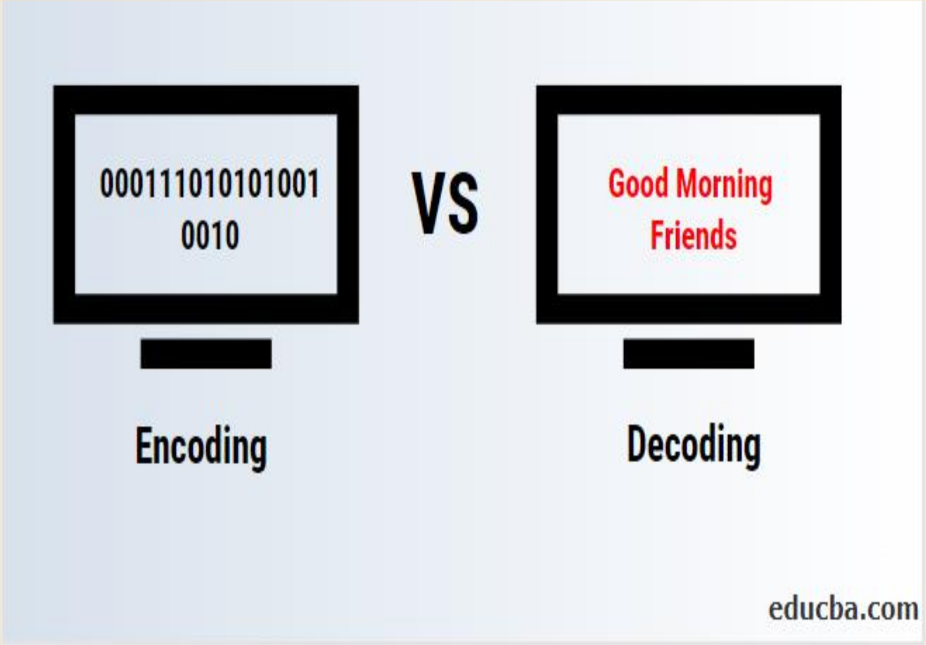


•**Encoding**: The process of adding extra bits (redundancy) to the original message so that errors can be detected and corrected during transmission.

•**Modulation**: The process of converting digital data (0s and 1s) into a signal that can travel through a communication channel, like radio waves or cables. Example: BPSK (Binary Phase Shift Keying) changes the phase of a signal to represent bits.

•**Message Passing Algorithm**: A method used in LDPC decoding where information is exchanged between different nodes (bits and checks) to gradually correct errors and recover the original message.

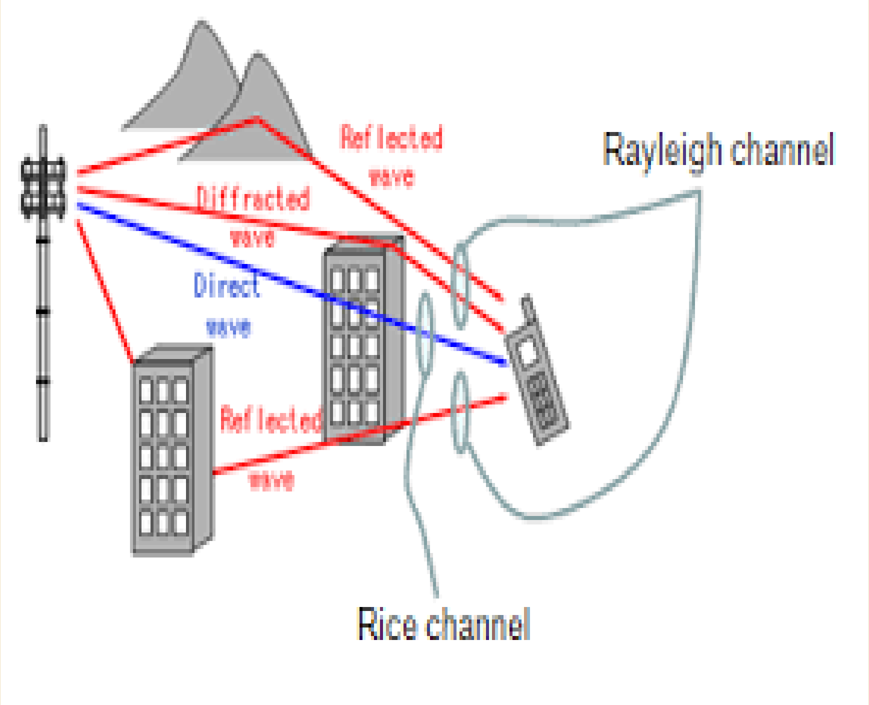
•**Decoding**: The process of detecting and correcting errors in the received message to retrieve the original transmitted data. LDPC decoding uses iterative algorithms like message passing to improve accuracy.



FADING CHANNELS

Rayleigh Fading channel:This occurs when there is **no direct line-of-sight (LOS**) between the transmitter and receiver. The signal reaches the receiver after reflecting off **multiple obstacles like buildings, trees**, or mountains, causing multiple indirect paths. The varying path lengths and interactions lead to constructive and destructive interference, resulting in rapid changes in signal strength.

Real-Life Example: Imagine you're in a dense urban area using your mobile phone. The signal from the cell tower doesn't have a clear path to your device due to numerous buildings. Instead, it bounces off various structures, causing fluctuations in the signal strength you experience.

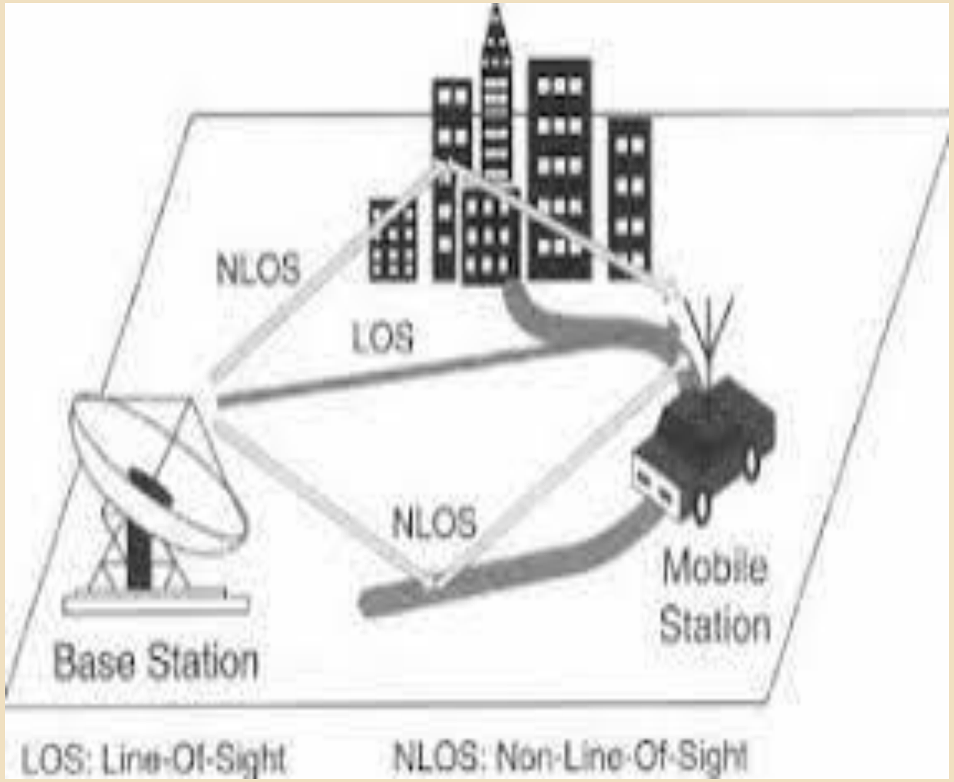


RICIAN FADING:

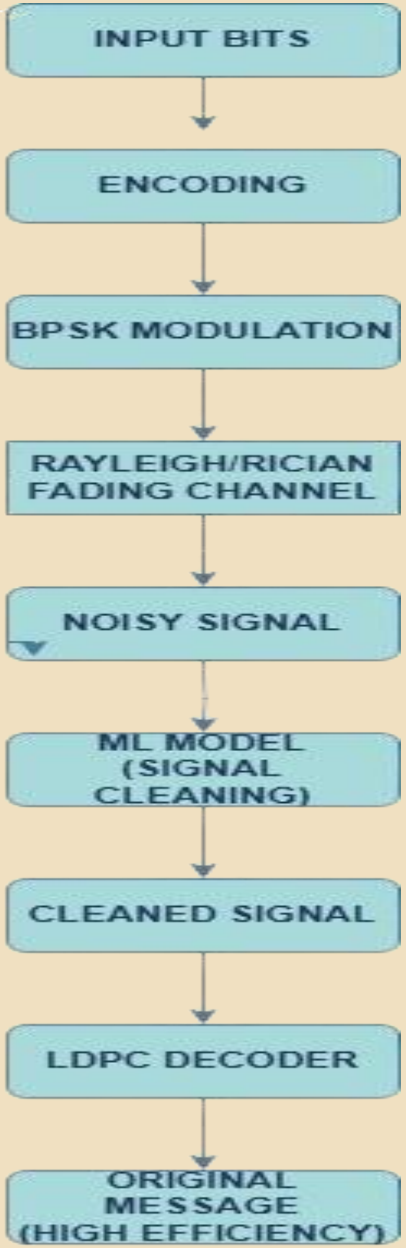
This occurs when there is a direct **LOS path** between the transmitter and receiver, **along with multiple indirect paths due to reflections**. The direct path provides a dominant signal component, while the indirect paths cause additional variations. The presence of the strong direct signal makes the fading less severe compared to Rayleigh fading.

Real-Life Example:

Consider using a GPS device in an open area with a clear view of the sky. The device receives signals directly from satellites (direct path) and also some signals reflected off nearby structures or the ground (indirect paths). While the direct signals are strong, the reflected ones can cause slight variations in the overall signal quality.



FLOW CHART:


INPUT BITS
ENCODING
BPSK MODULATION
RAYLEIGH/RICIAN FADING CHANNEL
NOISY SIGNAL
ML MODEL (SIGNAL CLEANING)
CLEANED SIGNAL
LDPC DECODER
ORIGINAL MESSAGE (HIGH EFFICIENCY)

ML MODELS:

Regression:

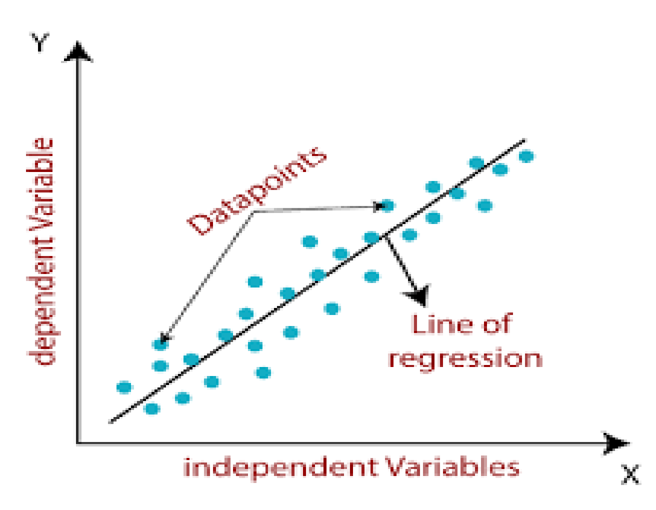
Regression is a machine learning technique used to find relationships between input and output variables. It helps predict a continuous value based on given data.

Example: Predicting house prices based on factors like area, number of rooms, and location.

Linear Regression

Linear regression is a type of regression where the relationship between input and output is modeled as a straight line. It predicts the output using the equation **Y = mX + C**, where **m** is the slope and **C** is the intercept.

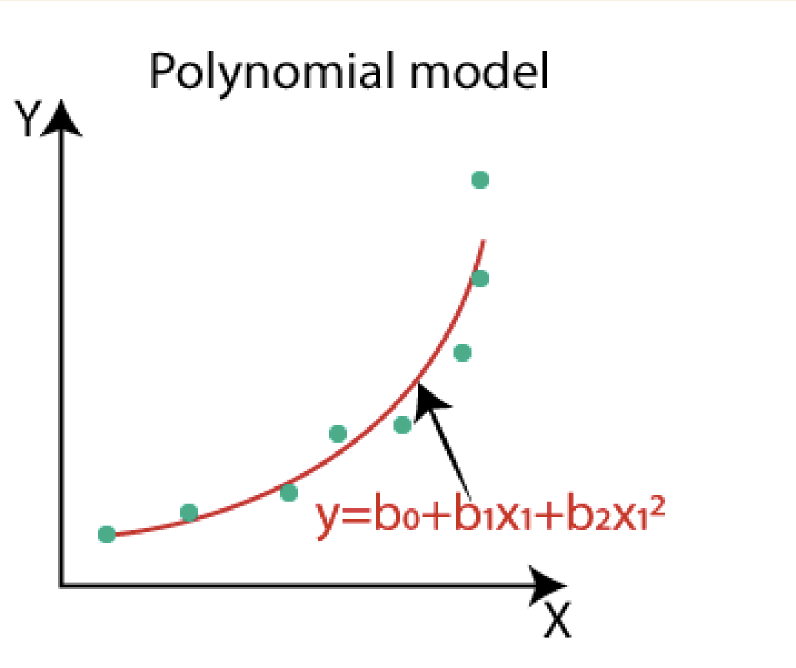
Example: Predicting a person's salary based on years of experience. If experience increases, salary also increases in a straight-line manner.



Polynomial Regression

Polynomial regression is an extension of linear regression where the relationship is modeled using a polynomial equation. It is useful when data shows a curved trend instead of a straight line.

Example: Predicting the speed of a car based on time. A car starts slow, speeds up, and then slows down, forming a curved pattern.



Fig(b) – Polynomial regression

DIFFERENCE BETWEEN LINEAR REGRESSION and POLYNOMIAL REGRESSION

Feature	Linear Regression	Polynomial Regression
Nature	Straight-line relationship	Curved relationship (higher-degree polynomial)
Equation Form	y=mx+c	$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$
Best For	Simple trends in data	Complex patterns in data
Flexibility	Less flexible, assumes linearity	More flexible, captures non-linear patterns
Overfitting Risk	Low	Higher if degree is too high

Why is Regression Used for Noise Removal?

- Regression helps **identify patterns** in data and removes random variations (noise).
- It predicts **expected values** by smoothing out unwanted distortions.
- Linear or polynomial regression is commonly used because **they approximate signals effectively**.

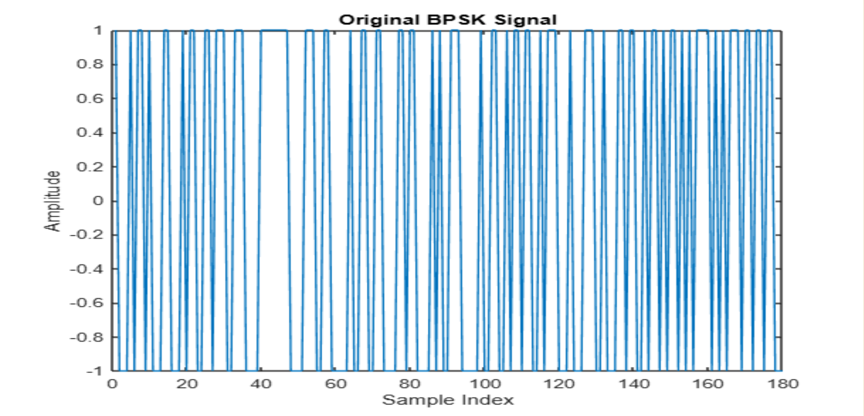
SIMULATION:

- **Encoded Output –** The input binary data was successfully encoded using LDPC codes, adding redundimprove error correction

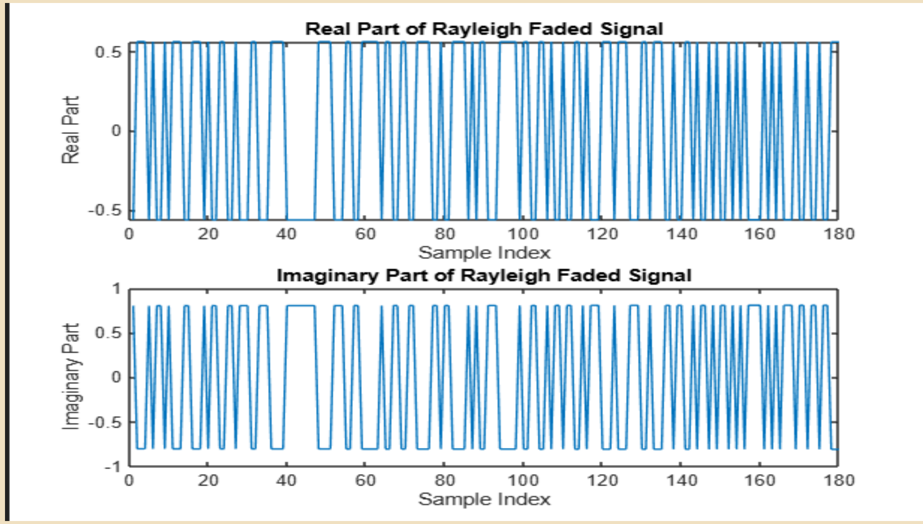
```
>> completecoderay1
Generated LDPC Codeword:
 Columns 1 through 36

   1   0   0   0   1   0   1   1   0   1   0   0   0   1   1   0   0   0   1   0   1   1   0   0   1   1   0   1   1   1   0   0   1   1   1   0

 Columns 37 through 72

   0   0   0   1   1   1   1   1   1   1   1   0   0   0   0   1   1   1   0   0   1   1   0   0   0   0   0   1   0   0   1   1   0   0   1   1

 Columns 73 through 108

   0   0   0   0   1   1   0   1   1   0   0   0   0   1   0   1   0   0   1   1   1   0   0   0   0   0   1   0   0   1   1   0   0   1   0   1

 Columns 109 through 144

   1   0   1   1   0   0   1   0   1   1   1   0   0   0   1   0   0   0   1   1   1   0   0   1   0   0   0   1   1   0   1   1   0   0   1   0

 Columns 145 through 180

   1   1   0   1   0   1   1   0   1   0   1   0   1   1   1   1   0   1   0   1   0   1   1   1   0   1   1   0   1   1   0   1   1   0   0   0
```

Fig(1) - codeword

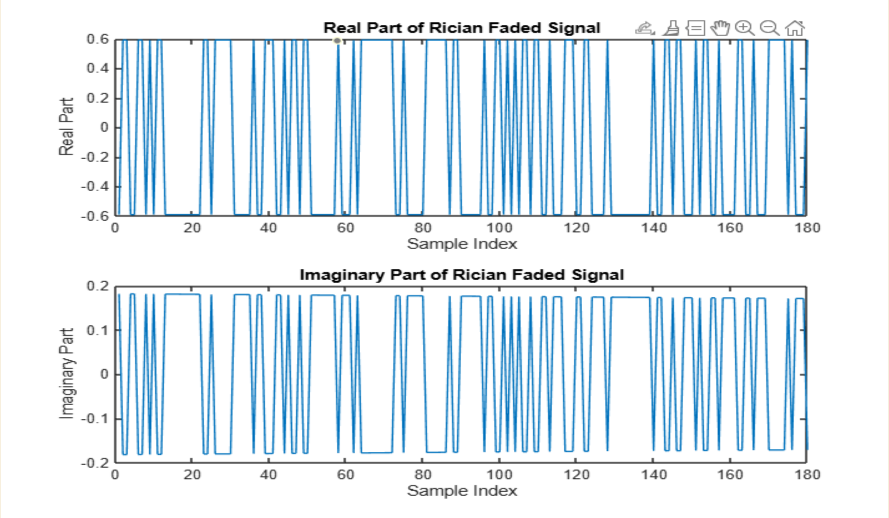
- **BPSK Modulation Output –** The encoded bits were modulated using BPSK, converting them into a modulated signal for transmission.



Noisy Signal Representation – After passing through Rayleigh and Rician fading channels, the signal was distorted, and the real and imaginary parts of the noisy signal were visualized in graphical form.



Fig(iii) – Rayleigh output

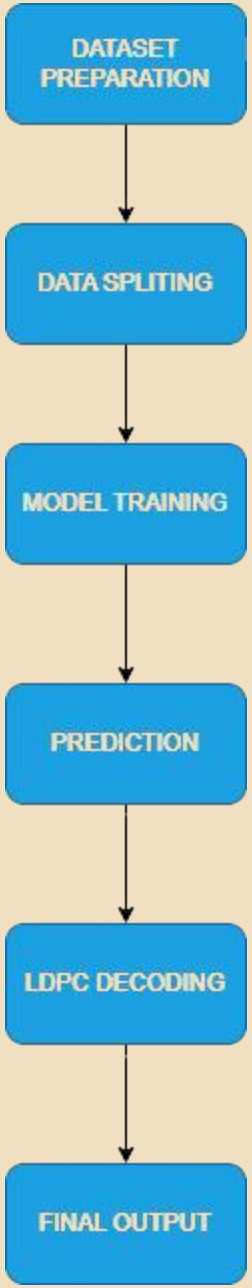


Fig(iv) – Rician output

Differences between Rayleigh and rician outputs:

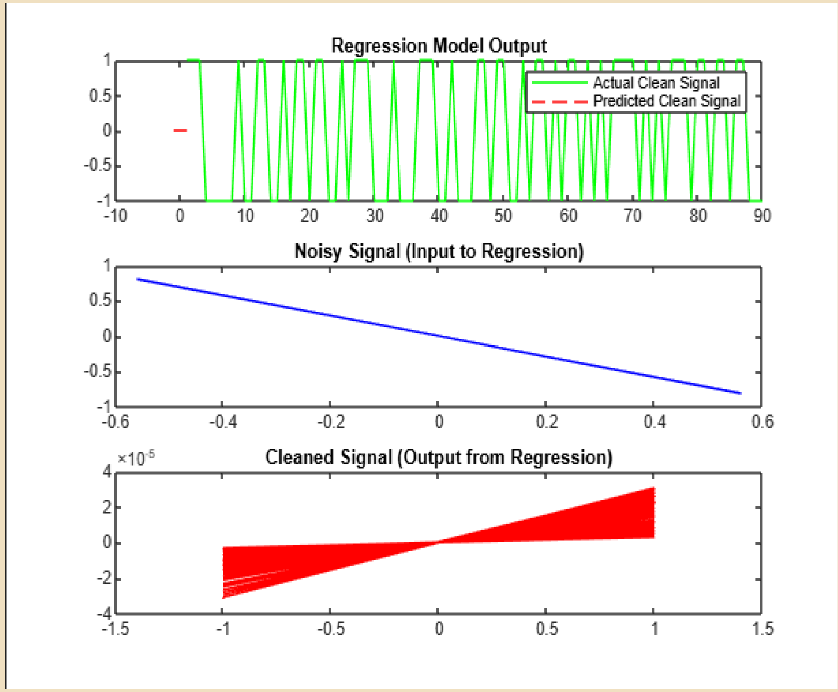
Feature	Rayleigh Fading	Rician Fading
Real Part Variation	Varies randomly between -0.5 to 0.5	Slightly biased, varies between -0.6 to 0.6
Imaginary Part Variation	More evenly spread across -1 to 1	Less variation, ranges between -0.2 to 0.2
LOS (Line-of-Sight) Component	Absent (pure multipath fading), More random, deep fades visible	Present (direct signal + multipath), Less severe fades due to LOS presence.

ML MODEL EXECUTION :

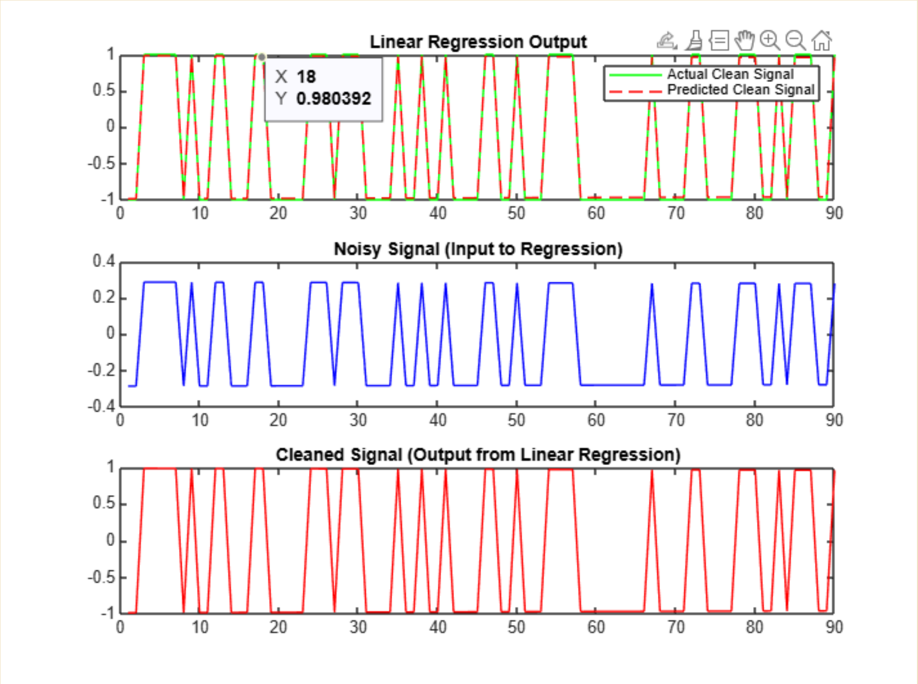



```
% Split Data for Training and Testing
train_ratio = 0.5;  % Adjusted for better test data
num_samples = length(X_clean);
train_size = round(train_ratio * num_samples);

X_train = Y_noisy(1:train_size);
Y_train = X_clean(1:train_size);
X_test = Y_noisy(train_size+1:end);
Y_test = X_clean(train_size+1:end);

% Train Regression Model
degree = 3;
p = polyfit(X_train, Y_train, degree);
```

ML Model Output – The regression-based ML model processed the noisy signal and provided a **cleaner version**, reducing distortion and improving signal quality.



Fig(v) – polynomial output



Fig(vi) – Linear output

Feature	Polynomial Regression	Linear Regression
Curve Shape	Highly fluctuating (overfitting)	Smoother and stable
Noise Handling	Captures noise, causing overfitting	Removes noise better
Predicted Signal	More complex, follows signal closely	Simpler, avoids unnecessary variations
Best For	When the relationship is truly curved	When the relationship is linear

Final Decoded Output – The LDPC message-passing algorithm decoded the cleaned signal, successfully recovering the original transmitted message with improved accuracy.

Sample decoded output:

```
Decoded Message:
 Columns 1 through 54

  1   0   1   1   0   1   1   1   1   1   1   0   1   0   1   0   0   1   0   1   0   0   1   1   1   1   1   0   1   1   0   1   1   0   1   0   0   0   1   0   0   0   1   0   1   1   1   1   0   1   0   0   0   0

 Columns 55 through 108

  0   1   1   0   1   0   0   1   0   0   1   0   0   1   0   0   0   0   0   1   0   1   0   1   1   0   0   1   1   0   1   0   0   0   0   0   1   0   0   0   0   0   0   1   0   1   0   1   1   0   1   1   1   1

 Columns 109 through 162

  0   0   1   1   1   1   1   1   1   0   1   0   0   1   0   1   0   1   0   0   1   1   1   1   0   0   0   1   0   0   1   0   1   1   1   0   0   0   0   0   1   0   0   0   0   1   1   0   1   1   0   0   0   1

 Columns 163 through 180

  1   0   1   0   1   1   1   0   0   1   1   0   0   0   1   0   0   1
```

Fig(vii) – final output

CONCLUSION

- Implemented **LDPC codes** for error correction in communication systems.
- Simulated performance under **Rayleigh and Rician fading channels**.
- Applied **message passing algorithm** for efficient decoding.
- Enhanced performance using **machine learning-based noise removal**.
- Highlighted LDPC’s importance in **modern communication systems like 5G**.

THANK YOU

Have a Great Day !