Aim: To simulate and analyze the performance of **Rayleigh and Ricean fading channel models** using **MATLAB software.**

Software: MATLAB

Theory:

The probability density function (PDF) of the received signal amplitude in Rayleigh fading is given by:

$$f(r) = \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}}, \quad r \geq 0$$

The Ricean fading model is characterized by the **K-factor**, which is the ratio of the power in the direct LOS path to the power in the scattered multipath components. The K-factor can be expressed as:

$$K = \frac{A^2}{2\sigma^2}$$

The probability density function (PDF) of the received signal in Ricean fading is given by:

$$f(r) = \frac{r}{\sigma^2} e^{-\frac{r^2+A^2}{2\sigma^2}} I_0\left(\frac{Ar}{\sigma^2}\right), \quad r \geq 0$$

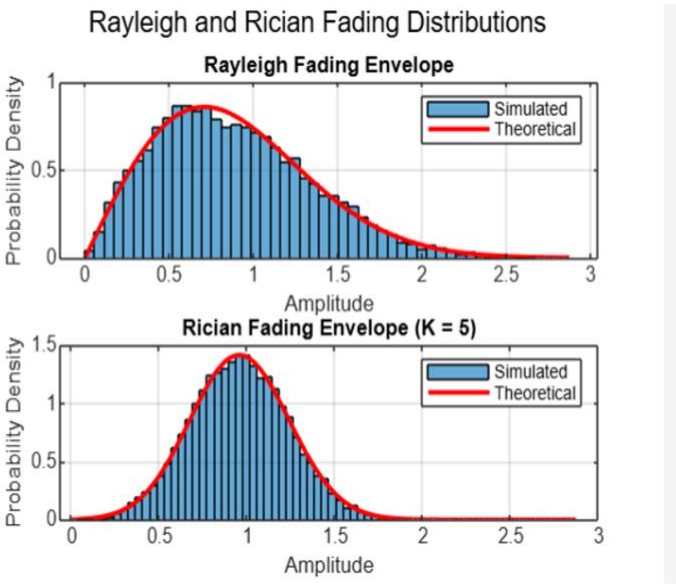
Program:

```
clc;

clear;

close all;

% Number of samples

numSamples = 10000;
```

```
% Generate Rayleigh Fading
rayleighReal = randn(1, numSamples);
rayleighImag = randn(1, numSamples);
rayleighSignal = (rayleighReal + 1i * rayleighImag) / sqrt(2);
rayleighEnvelope = abs(rayleighSignal);
% Generate Rician Fading
K = 5; % Rician K-factor
LOS_Component = sqrt(K / (K + 1));
scattered = sqrt(1 / (K + 1)) * (randn(1, numSamples) + 1i * randn(1, numSamples)) / sqrt(2);
riceanSignal = LOS_Component + scattered;
riceanEnvelope = abs(riceanSignal);
% Plot Histogram of Rayleigh
figure;
subplot(2,1,1);
histogram(rayleighEnvelope, 50, 'Normalization', 'pdf');
hold on;
% Theoretical Rayleigh PDF
r = linspace(0, max(rayleighEnvelope), 100);
sigma = 1/sqrt(2); % standard deviation for normalized power = 1
rayleighPDF = (r / sigma^2) .* exp(-r.^2 / (2 * sigma^2));
plot(r, rayleighPDF, 'r', 'LineWidth', 2);
title('Rayleigh Fading Envelope');
xlabel('Amplitude');
```

```
ylabel('Probability Density');
legend('Simulated', 'Theoretical');
grid on;
% Plot Histogram of Rician
subplot(2,1,2);
histogram(riceanEnvelope, 50, 'Normalization', 'pdf');
hold on;
% Theoretical Rician PDF
ricianPDF = 2*(1+K).*r .* exp(-(1+K)*r.^2 - K) .* besseli(0, 2*r.*sqrt(K*(1+K)));
plot(r, ricianPDF, 'r', 'LineWidth', 2);
title(['Rician Fading Envelope (K = ' num2str(K) ')']);
xlabel('Amplitude');
ylabel('Probability Density');
legend('Simulated', 'Theoretical');
grid on;
% Super title for both plots
sgtitle('Rayleigh and Rician Fading Distributions');
```

OUTPUT:



Result:

The MATLAB code successfully simulated:

- Rayleigh fading by generating complex Gaussian noise without any line-of-sight (LOS) component.
- Rician fading by introducing a deterministic LOS component along with Gaussian multipath components (K-factor = 5).
- Histogram plots of the simulated envelope magnitudes were created and normalized.
- Theoretical PDFs were computed and plotted using the known Rayleigh and Rician distribution formulas.
- The simulated histograms closely matched the theoretical PDFs, validating the correctness of the simulation models.

Aim: To simulate **indoor propagation models** using **MATLAB software** in order to analyze the behavior of radio wave propagation within an indoor environment. The experiment focuses on modeling and simulating indoor propagation model to understand how signals propagate inside buildings and other indoor structures.

Software: MATLAB

Theory:

1. **Free Space Path Loss (FSPL) – Ideal Case**

Ideal line-of-sight (LoS) scenario without any obstructions, used as a baseline.

Mathematical Model:

$$PL_{FS}(d) = 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) \quad \text{or} \quad PL_{FS}(dB) = 20\log_{10}(d) + 20\log_{10}(f) - 147.55$$

2. **Log-Distance Path Loss Model**

✅ **Without Shadowing (Basic Form):**

$$\boxed{PL(d) = PL(d_0) + 10n \cdot \log_{10}\left(\frac{d}{d_0}\right)}$$

✅ **With Shadowing (Log-Normal Fading):**

$$\boxed{PL(d) = PL(d_0) + 10n \cdot \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma}$$

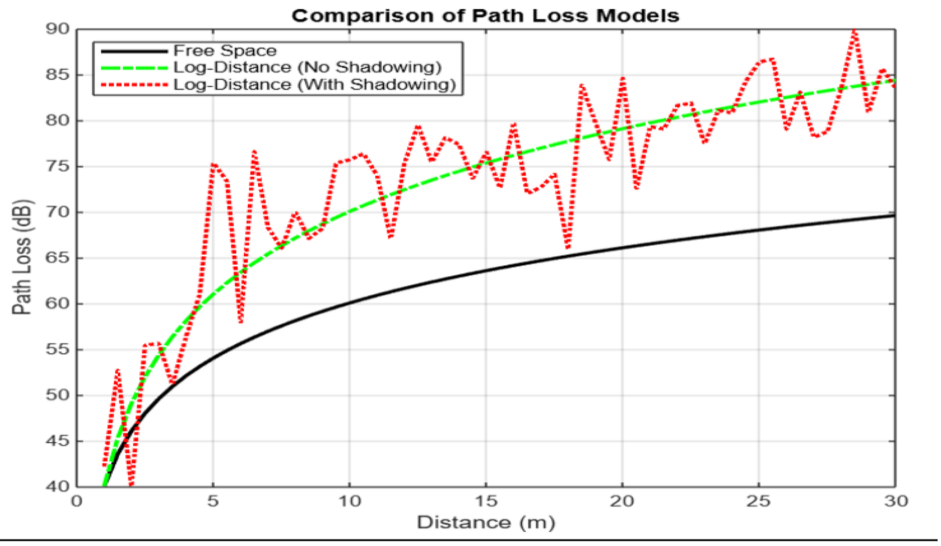
Program:

```
clc;

clear;

close all;
```

```
% Common Parameters
f_Hz = 2.4e9;              % Frequency in Hz
f_MHz = f_Hz / 1e6;        % Frequency in MHz
c = 3e8;                   % Speed of light (m/s)
lambda = c / f_Hz;         % Wavelength
d = 1:0.5:30;              % Distance range in meters
d0 = 1;                    % Reference distance
% 1. Free Space Path Loss
PL_fs = 20 * log10(d) + 20 * log10(f_Hz) - 147.55;

%2. Log-Distance Path Loss
n = 3;                                          % Path loss exponent (e.g., indoor NLOS)
sigma = 4;                                      % Shadowing std deviation (dB)
PL_d0 = 20 * log10(4 * pi * d0 / lambda);  % Reference path loss (FSPL at d0)
PL_log = PL_d0 + 10 * n * log10(d / d0);   % Without shadowing
X_sigma = sigma * randn(size(d));           % Shadowing component
PL_log_shadow = PL_log + X_sigma;          % With shadowing
% Plotting All Models
figure;
plot(d, PL_fs, 'k-', 'LineWidth', 2); hold on;
plot(d, PL_log, 'g-.', 'LineWidth', 2);
plot(d, PL_log_shadow, 'r:', 'LineWidth', 2);
grid on;
xlabel('Distance (m)');
```

```
ylabel('Path Loss (dB)');
title('Comparison of Path Loss Models');
legend('Free Space', ...
    'Log-Distance (No Shadowing)', ...
    'Log-Distance (With Shadowing)', ...
    'Location', 'northwest');
```

Output:



Result:

The MATLAB simulation was performed to compare three widely used wireless path loss models — Free Space Path Loss (FSPL) and the Log-Distance Path Loss Model (with and without shadowing) — over a distance range of 1 to 30 meters at a frequency of 2.4 GHz.

Aim: To simulate and analyze the outdoor path loss propagation models in wireless communication system using MATLAB software

1. Free space 2. Okumura 3. Hata model

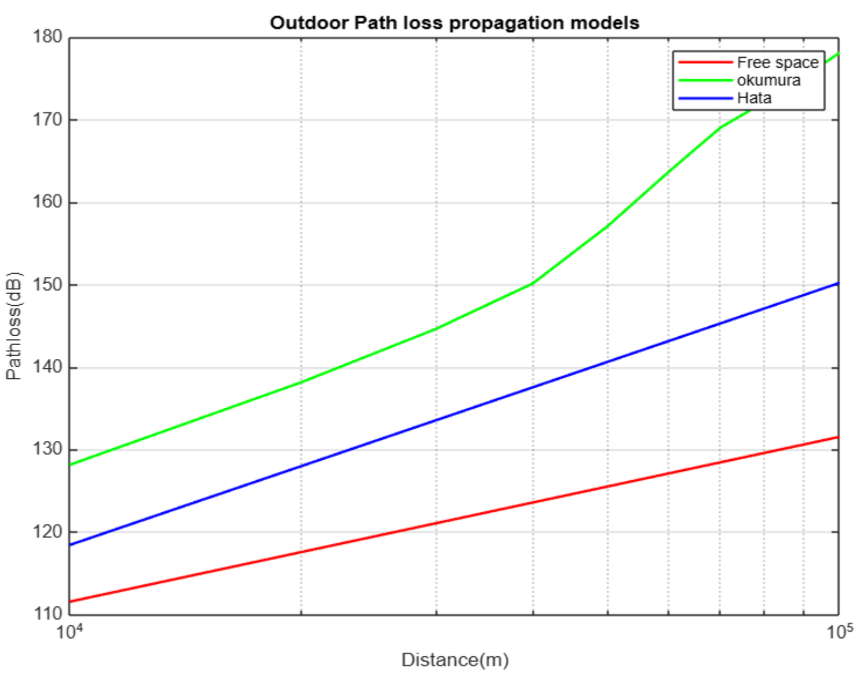
Software:

Program:

```
clc
clear all
close all

frequency=900e6;
d=10000:10000:100e3;
h_tx=100;
h_rx=5;
fspl=(20*log10(d)) + (20*log10(frequency))-147.55;

%Okumura model
Amu=[30 34 37 40 45 50 54 56 58 60];
C=Amu-13.4;
L_okumura = fspl+C;

%Hata Model
a_h = (1.1*log10(frequency)-0.7)*h_rx - ((1.56*log10(frequency))-0.8);
L_hata = 69.55+26.16*log10(frequency/1e6) - 13.82*log10(h_tx)- a_h +(44.9 - 6.55*log10(h_tx))*log10(d/1000);
```

```
%figure
semilogx(d,fspl,'r','LineWidth',1.5); hold on;
semilogx(d,L_okumura,'g','LineWidth',1.5);
semilogx(d,L_hata,'b','LineWidth',1.5);
grid on;
xlabel("Distance(m)");
ylabel("Pathloss(dB)");
title("Outdoor Path loss propagation models")
legend("Free space","okumura","Hata")
hold off;
```

OUTPUT:



Result:

This MATLAB simulation compares outdoor path loss using three models

1. Free space 2. Okumura 3. Hata model

The Okumura model adds urban environment correctors , increasing loss. The hata model further accounts antenna height and frequency, providing more accurate path for real world urban area

Simulation of Handoff performance in mobile communication system.

Aim: To simulate and analyze the handoff performance in mobile communication systems. This involves studying the process of transferring a mobile device's connection from one cell (base station) to another as it moves across different coverage areas.

Software: MATLAB

Program :

```
clc;
clear all;
close all;
cell1 = [0, 0];              % Coordinates of Base Station 1
cell2 = [100, 0];            % Coordinates of Base Station 2
cellRadius = 50;             % Coverage radius of each cell
handoffThreshold = 0.6;      % Trigger handoff if > 60% away from current BS
userPos = [0, 0];            % Starting user position
velocity = [1, 0];           % User moves along +X axis, 1 unit per step
currentCell = 1;             % Keeps track of which BS the user is connected to
handoffOccurred = false;     % Becomes true if a handoff takes place
figure;                      % Opens a new figure window
hold on;                     % Allows multiple plots on same figure
axis equal;                  % Keeps units same along X and Y
xlim([-50 150]);             % Sets visible range on the plot axes
ylim([-50 50]);
title('Simple Handoff Mechanism'); % Add labels to the plot
xlabel('X');
ylabel('Y');
viscircles(cell1, cellRadius, 'Color', 'b');        % Draws coverage circles around each BS
viscircles(cell2, cellRadius, 'Color', 'r');
% Plots BS positions using colored markers
```

plot(cell1(1), cell1(2), 'bo', 'MarkerSize', 10, 'LineWidth', 2);

plot(cell2(1), cell2(2), 'ro', 'MarkerSize', 10, 'LineWidth', 2);

for t = 1:120 **%Runs simulation for 120 time steps.**

userPos = userPos + velocity; **%Moves the user +1 unit along X at each time step.**

dist1 = norm(userPos - cell1); **%Calculates Euclidean distance from user to each BS**

dist2 = norm(userPos - cell2);

% Check handoff condition

- **%Handoff triggers when:**
    - **Distance from cell 1 > 30**
    - **Distance from cell 2 < 50**

if currentCell == 1 && dist1 > handoffThreshold * cellRadius && dist2 < cellRadius

currentCell = 2; **%Once the handoff condition is true, this line updates the user's connection to cell 2**.

handoffOccurred = true;

fprintf('Handoff occurred at time %d seconds. New cell: %d\n', t, currentCell);

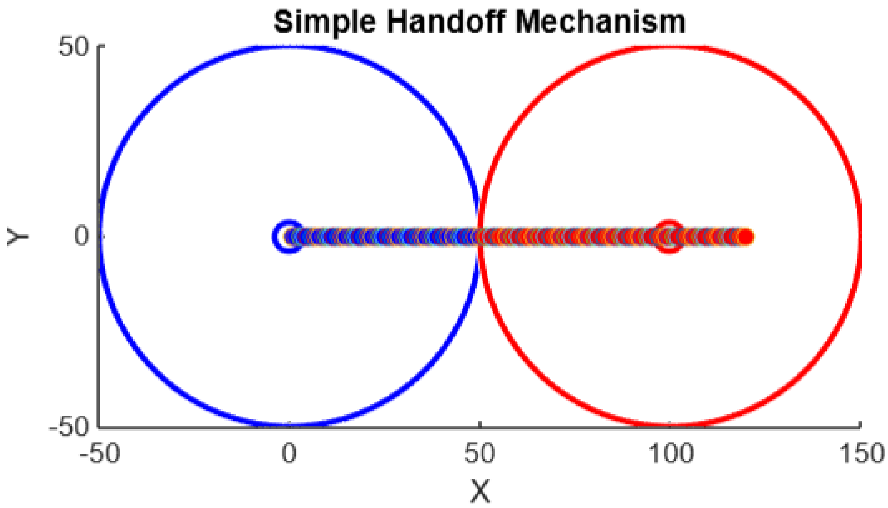
%This prints a message on the Command Window,

end

```
% Plot user position
    if currentCell == 1
        userColor = 'b';
    else
        userColor = 'r';
    end
    plot(userPos(1), userPos(2), 'o', 'MarkerFaceColor', userColor);
    pause(0.05);
end

% Final message
if handoffOccurred
    disp('Handoff simulation completed successfully.');
```

```
else
    disp('No handoff occurred during simulation.');
end
```

Output:



Result:

- The simulation was performed with 10 mobile users moving within a region covered by 3 base stations arranged in a triangular layout.
- Each user was initially associated with the nearest base station based on their position.
- Handoff decisions were triggered when users approached the edge of their serving cell and a nearby base station was closer.
- A total of 37 handoff attempts were observed during the 200 time-step simulation.
- Out of these, 36 handoffs were successful, while 1 handoff failed, based on a 98% success probability.
- The final handoff success rate achieved in the simulation was approximately 97.30%.
- The visualization dynamically displayed user movements, cell coverage, and updated handoff statistics in real-time.

Resource Allocation using frequency reuse in cellular network using MATLAB software.

Aim: To simulate and analyze resource allocation techniques in a cellular network using frequency reuse, with the help of MATLAB software. The experiment focuses on optimizing the use of available frequency spectrum across different cells in a cellular system to maximize network efficiency and minimize interference.

Software: MATLAB

Program :

```
clc;

clear all;

close all;

% Parameters

numCells = 7; % Number of cells in the network

numUsers = 20; % Number of users

frequencyBands = 3; % Number of frequency bands available

cellRadius = 1000; % Radius of each cell in meters

% Generate random user locations

userLocations = rand(numUsers, 2) * cellRadius;

% Frequency reuse pattern

reusePattern = [1 2 3; 2 3 1; 3 1 2];

% Assign frequencies to users based on their cell location

userFrequencies = zeros(numUsers, 1);
```

```
for i = 1:numUsers

    cellIndex = mod(floor(userLocations(i, 1) / (cellRadius / sqrt(numCells))), numCells) + 1;

    userFrequencies(i) = reusePattern(cellIndex, mod(i, frequencyBands) + 1);

end
```

% Display results

```
disp('User Locations:');

disp(userLocations);

disp('Assigned Frequencies:');

disp(userFrequencies);
```

OUTPUT:

```
User Locations:

  113.1938  230.7972

   71.6234  702.4920

  581.0154  309.4704

   10.9610  973.8694

  843.5013  862.4361

   66.0618  619.7731

  484.0970  879.4468

  162.4027  307.1141

  421.9582  626.1958

  863.0600  633.9832
```

Assigned Frequencies:

2

3

2

2

2

1

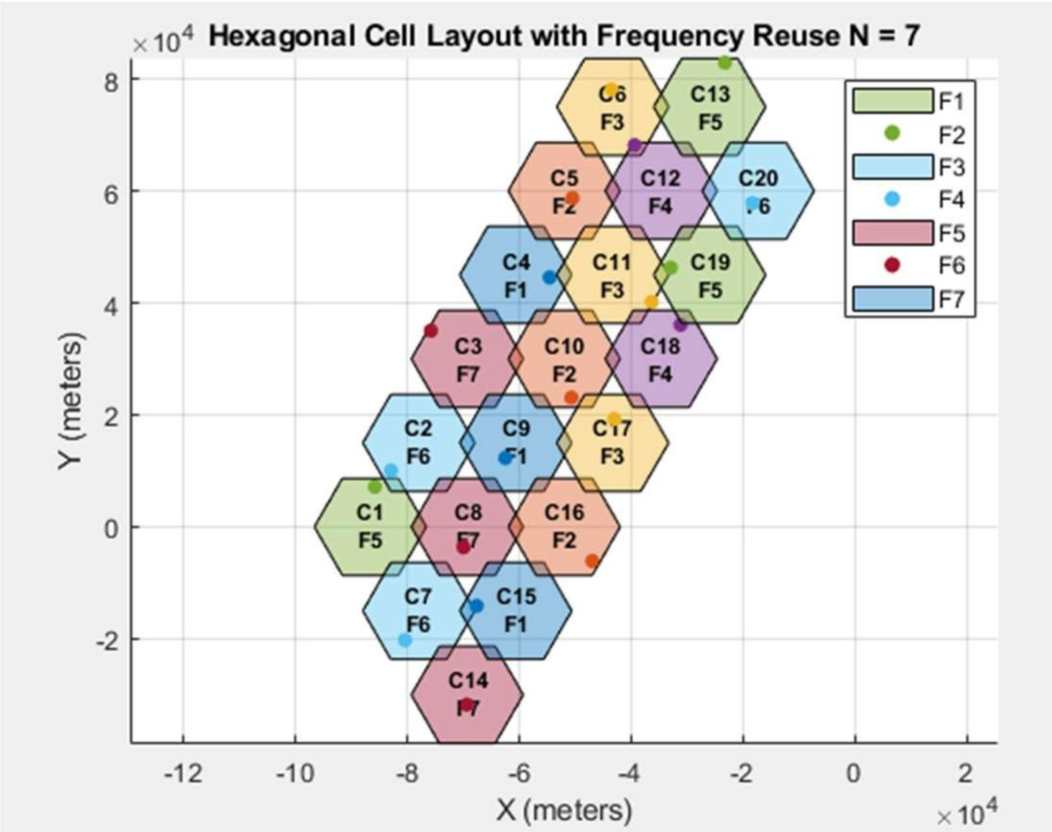
3

3

2

1

1



Result:

1. The simulation successfully generated a hexagonal cellular layout with N = 7 frequency reuse and a total of 20 cells.
2. The calculated reuse distance is: D=45825.76 meters
3. Frequencies were assigned to each cell using the spatial reuse pattern (iShift=2, jShift=1) for N=7.
4. Users per cell = 5, randomly placed inside the hexagonal boundaries of each cell.
5. A user data table was generated, listing:
    - The cell index
    - The assigned frequency
    - The (X, Y) coordinates of each user's location in meters.
6. The plot displayed:
    - Hexagonal cells color-coded by frequency.
    - Users as colored dots matching the cell's frequency.
    - Cell IDs and Frequency IDs labeled at the cell centers.