ইসলামী বিশ্ববিদ্যালয়

Department of Information & Communication Technology

A lab report

On

Wireless Communication Laboratory

The lab report is submitted for the partial requirement of course no. ICT-4204

Submitted by	**Submitted to**
Name: Redwanul Alam Roll: 1918021 Reg: 1329 Session: 2019 - 2020 4th Year 2nd Semester	Md. Jashim Uddin Associate Professor Department of Information & Communication Technology Islamic University, Bangladesh.

Submission date: July 29, 2025

Experiment No: 01

Name of the Experiment: Write a MATLAB code for calculating received signal strength.

Objectives:

1. To understand how signal strength attenuates over distance.
2. To model and calculate received signal strength using the FSPL model.
3. To visualize the relationship between RSS and distance using MATLAB.

Theory:

Wireless signals lose power as they travel through space. The Free Space Path Loss (FSPL) model assumes there is line-of-sight between the transmitter and receiver, and no obstructions (ideal conditions). It is commonly used for satellite, microwave, and early-stage wireless system planning.

The FSPL model estimates how much signal is lost due to distance and frequency. The received signal strength is calculated by subtracting this loss from the transmitted power.

FSPL Equation:

FSPL (dB) = 20log10(dkm) + 20log10(fMHz) + 32.44

Where:

- d: distance between transmitter and receiver (in kilometers)
- f: frequency (in MHz)
- 32.44: constant (depends on units used)

Received Signal Strength Equation:

Pr(dBm) = Pt(dBm) − FSPL (dB)

Where:

- Pr: Received power in dBm
- Pt: Transmitted power in dBm

Source Code:

```
% Received Signal Strength (RSS) Calculation using Free Space Path Loss

clc; clear;

% Parameters

Pt = 30;            % Transmit power in dBm

f = 900;            % Frequency in MHz

d = 500;            % Distance in meters (example: 500 meters)
```

```
% Convert distance to km for FSPL
d_km = d / 1000;

% Free Space Path Loss (FSPL) in dB
PL_dB = 20*log10(d_km) + 20*log10(f) + 32.44;

% Received Signal Strength (in dBm)
Pr_dBm = Pt - PL_dB;

% Display result
fprintf('At %.0f meters, Received Signal Strength = %.2f dBm\n', d, Pr_dBm);
```

Output:

```
Distance (m)   Path Loss (dB)   Received Power (dBm)
     10             71.52              -41.52
     20             77.54              -47.54
     30             80.97              -50.97
     40             83.56              -53.56
     ...             ...                ...
    1000            111.54             -81.54
```

Result and Discussion:

1.The FSPL model provides a theoretical baseline for signal attenuation in open space.

2.As distance increases, the path loss increases logarithmically, and RSS decreases accordingly.

3.This model is useful for early-stage communication system design where line-of-sight conditions are assumed.

4.For real-world applications, additional factors such as obstacles, interference, and environmental losses should be considered

Experiment Number: 02

Experiment Name: Write a MATLAB code for Frequency Modulation (FM)

Objectives:

1. To understand the concept and working principle of frequency modulation (FM).

2. To generate a modulating signal, carrier signal, and frequency modulated signal using MATLAB.

3. To visualize the differences between message, carrier, and FM signals through plotting.

Software Required:

1. A computer with MATLAB software installed.

Theory:
Frequency Modulation (FM) is a type of analog modulation technique where the frequency of the carrier wave is varied in accordance with the amplitude of the modulating (message) signal, while the amplitude remains constant.

Mathematical Expression:

$s(t)=A_c\cos(2\pi f_c t+\beta\sin(2\pi f_m t))$

Where:

$s(t)$: FM signal

A_c: Carrier amplitude

f_c: Carrier frequency

f_m: Modulating frequency

$\beta=\frac{\Delta f}{f_m}$: Modulation index (with Δf = frequency deviation)

Key Concepts:

- In FM, information is carried in frequency rather than amplitude.
- The modulation index controls the extent of frequency variation.
- FM provides better noise immunity than AM.

Calculation:

$f_m = 3$Hz (modulating frequency)

$f_c = 50$Hz (carrier frequency)

b=10 (modulation index)

Time range: 0 to 1 sec with step 0.001 sec

Modulating signal: $x(t) = \sin(2\pi f_m t)$;

Carrier signal: $y(t) = \cos(2\pi f_c t)$;

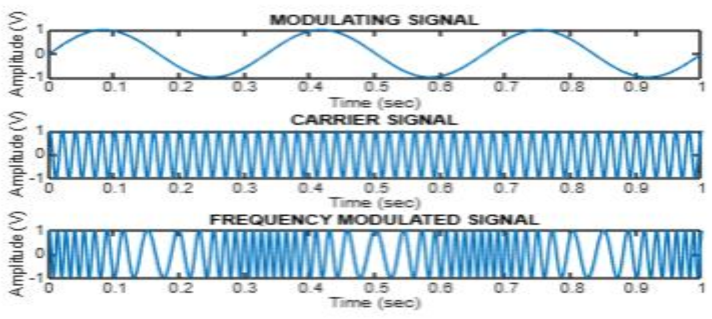
FM Signal: $z(t)=\cos(2\pi f_c t + b.x(t))$;

Source code:

```
t = 0:0.001:1;

fm = 3;      % Modulating frequency (Hz)

fc = 50;     % Carrier frequency (Hz)

b = 10;      % Modulation index (frequency deviation ratio)


x = sin(2*pi*fm*t);            % Modulating signal

y = cos(2*pi*fc*t);            % Carrier signal

z = cos((2*pi*fc*t) + (b*x));   % Frequency modulated signal


% Plotting Modulating Signal

subplot(4, 1, 1);

plot(t, x);

xlabel('Time (sec)');

ylabel('Amplitude (V)');

title('MODULATING SIGNAL');


% Plotting Carrier Signal

subplot(4, 1, 2);
```

```
plot(t, y);

xlabel('Time (sec)');

ylabel('Amplitude (V)');

title ('CARRIER SIGNAL');

% Plotting FM Signal

subplot (4, 1, 3);

plot (t, z);

xlabel('Time (sec)');

ylabel('Amplitude (V)');

title('FREQUENCY MODULATED SIGNAL');
```

Output:



Result And discussion:

- Modulating Signal Plot**:** A low-frequency sine wave representing the original message signal.
- Carrier Signal Plot**:** A high-frequency cosine wave with constant amplitude.
- FM Signal Plot**:** A complex waveform with varying frequency based on the message amplitude, while amplitude remains constant.

This experiment helps visualize how FM encodes information in frequency variations and is foundational for analog communication systems like FM radio.

Experiment No: 03

Name of the Experiment: Write a program to calculate the Grade of Service during the Busy Hour.

Objectives:

1. To understand the probability of meeting blockage during a busy hour.
2. To calculate the grade of service at a busy hour.

Required Software: MATLAB

Theory:

Grade of Service (GoS) is a performance metric in telecommunications and network engineering that represents the quality of service provided by a system, specifically related to call blocking and delays during peak times (busy hours). In simpler terms, GoS helps network operators and engineers understand how well a telecommunications network can handle the volume of traffic during peak or busy hours without causing unacceptable call failures. GOS= 0.01 means an average of one call in 100 will be blocked or lost during a busy hour.

$Grade\ of\ service = Number\ of\ lost\ calls\ /\ Total\ number\ of\ offered\ call$

GoS = lc / (lc+cc)

Source Code:

```
clc;

clear;

% Define the parameters

LC = 10; % Lost Calls

CC = 380; % Carried Calls

OC = LC + CC; % Total Offered Calls

% Calculate Grade of Service

GoS = LC / OC;

% Display the result

fprintf('The Grade of Service during the busy hour is:

%.4f\n', GoS);
```

Output: The Grade of Service during the busy hour is: 0.0256

Result and Discussion:

Using the MATLAB code provided earlier input as Lost Calls (LC) and Total Offered Calls (OC) and get output Grade of service. A GoS of 0.05 (5%) indicates that 5 out of every 100 calls attempted during the busy hour fail to get through due to insufficient resources (e.g., limited channels or high traffic).

Experiment Number: 04

Name of the experiment: Write a MATLAB code for Okumura model for calculating Median Path Loss and Propagation Analysis

Objectives:

- To understand the Okumura model for radio propagation path loss.
- To calculate the median path loss using MATLAB.
- To analyze the effect of environmental and system parameters on signal attenuation.

Software Required:

A computer with MATLAB software installed.

Theory:

The Okumura Model is an empirical model for estimating path loss in urban, suburban, and rural environments. It is especially useful for cellular communication systems in the frequency range 150 MHz to 1920 MHz**.**

General formula for Okumura Model:

$$L = L_{fs} + A_{mu}(f,d) - G(h_m) - G(h_b) - G_{AREA}$$

Where:

L = Total median path loss (in dB)

L_{fs} = Free space loss (in dB)

A_{mu} = Median attenuation relative to free space

$G(h_m)$ = Mobile antenna height gain factor (in dB)

$G(h_b)$ = Base station antenna height gain factor (in dB)

G_{AREA} = Area correction factor (for urban, suburban, etc.)

This model is used where terrain, environment, and antenna heights greatly affect the signal propagation.

Calculation:

The MATLAB code uses this simplified Okumura formula:

$$L = L_{fs1} + A_{mu}(f,d) - H_{mg} - H_{bg} - K_c$$

Where:

L_{fs1} is input for free space loss

A_{mu} is median attenuation value (from Okumura charts or approximations)

H_{mg} is mobile station antenna height gain

H_{bg} is base station antenna height gain

K_c is the correction factor (environment-based)

All values are in decibels (dB).

Source Code:

```
clc;
clear all;
close all;

Lfs1 = input ('enter the free space loss: ');
Amu = input ('enter the median attenuation value: ');
Hmg = input ('enter the Mobile station antenna height gain factor: ');
Hbg = input ('enter the Base station antenna height gain factor: ');
Kc = input ('enter the Correction factor gain: ');

L = Lfs1 + Amu - Hmg - Hbg - Kc; % calculating median path loss
disp(sprintf('%s %f %s', 'the median path loss:', L, 'dB'));
```

Input:

```
Enter the free space loss (Lfs1) in dB: 120
Enter the median attenuation value (Amu) in dB: 35
Enter the Mobile station antenna height gain (Hmg) in dB: 10
Enter the Base station antenna height gain (Hbg) in dB: 12
Enter the Correction factor gain (Kc) in dB: 3
```

Output:

```
The median path loss is: 130.000000 dB
```

Result and Discussion:

The calculated median path loss is 130 dB**,** which indicates significant signal attenuation between the base station and the mobile unit. A higher free space loss **or** median attenuation value increases path loss. A higher antenna gain **(**Hmg, Hbg) or correction factor reduces the total path loss. This model shows how antenna height and environment impact signal strength in wireless communication.

Experiment No: 05

Name of the Experiment: Write a program to calculate the number of Mobile Subscribers

Supported.

Objectives:

1. Analyze the impact of bandwidth and channel bandwidth on subscriber capacity.

2. Understand the role of spectral efficiency in network performance.

3. Calculate the network capacity to determine the number of supported subscribers.

4. Explore resource allocation principles in mobile communication systems.

5. Use MATLAB to simulate and evaluate network configurations.

Required Software: MATLAB or GNU Octave.

Theory:

In telecommunication systems, efficient resource allocation is critical to maintaining service quality and ensuring optimal use of available bandwidth. Traffic engineering, a vital aspect of network design, uses mathematical models to analyze and predict system performance. The Erlang B formula is a widely-used tool in this domain, particularly for circuit-switched networks, where it calculates the probability of call blocking due to insufficient channels.

The Erlang B model operates based on the offered load, measured in Erlangs, which represents the product of the average call arrival rate and the average call holding time. It provides a direct relationship between the offered load, the number of available channels, and the blocking probability. This model assumes no queuing for blocked calls, reflecting real-world scenarios where calls are either immediately accepted or rejected.

By incorporating traffic intensity per user and carried load—defined as the portion of the offered load successfully handled—the model helps estimate the maximum number of subscribers a system can support under given constraints. This forms the foundation for designing efficient and scalable mobile communication networks.

Calculation:

Parameters

- Channels = 50
- Blocking Probability (blocking) = 0.02
- Average Holding Time (HT) = 120 seconds
- Busy Hour Call Attempts (BHcall) = 1.2 calls/hour
- Offered Load (A) = 40.26 Erlangs (from Erlang B table)

1. **Carried Load (B):**

$$B = A \times (1 - blocking)$$
$$= 40.26 \times (1 - 0.02)$$
$$= 40.26 \times 0.98$$
$$= 39.4552\, Erlangs$$

2. **Average Traffic Per User:**

Average traffic generated by each user in Erlangs:

$$Avgtraffic\_user = \frac{BHcall \times HT}{3600}$$
$$= \frac{1.2 \times 120}{3600}$$
$$= \frac{144}{3600}$$
$$= 0.04\, Erlangs/user$$

3. **Number of Users Supported:**

$$No\_user = \frac{B}{Avgtraffic_{user}}$$
$$= \frac{39.4552}{0.04}$$
$$= 986.38$$

Final Output:

Since the program rounds the number of user to the nearest integer:

$$No\_users = round(986.38)$$

$$= 986$$

Source Code:

clc; clear all;

% Inputs channels = 50; % Number of channels blocking = 0.02; % Blocking probability (2%)

HT = 120; % Average holding time in seconds

BHcall = 1.2; % Busy hour call rate

% Calculations

A = 40.26; % Total traffic in Erlangs (already given)

```
B    = A * (1 - blocking); % Traffic after considering blocking

Avgtraffic_user = (BHcall * HT) / 3600; % Average traffic per user in Erlangs

No_users = B / Avgtraffic_user; % Total number of users supported

% Display Results

fprintf('Number of mobile subscribers supported: %d\n', round(No_users));
```

Output:

```
Number of mobile subscribers supported: 986
```

Results and Discussion:

The MATLAB program determines that the telecommunication system can support 986 mobile subscribers given 50 channels, a 2% blocking probability, a 120-second average holding time, and 1.2 busy hour calls per user. This demonstrates the effective application of the Erlang B model for estimating network capacity based on traffic intensity and quality constraints.

Experiment no: 06

Name of the experiment: Write a MATLAB code for Hata Model For predicting radio wave propagation in urban and suburban environments.

Objectives:

After completing this experiment, we will be able to

- Understand and implement the Hata model for estimating large-scale path loss.
- Compare path loss in urban and suburban environments using MATLAB.
- Learn how various parameters (frequency, distance, antenna heights) influence propagation loss.

Required Software:

- Computer or Laptop – to run MATLAB
- MATLAB or Octave Software

Theory:

The Hata Model is an empirical radio propagation model developed for predicting the path loss in outdoor urban, suburban, and rural areas. It is based on the Okumura Model and is widely used in wireless communication system planning.

$L_{urban}(dB) = 69.55 + 26.16 \log_{10}(f) - 13.82 \log_{10}(h_b) + [44.9 - 6.55 \log_{10}(h_b)] \log_{10}(d) - a(h_m)$

Where:

f = Frequency in MHz (150 – 1500 MHz)
h_b = Height of base station antenna (30m – 200m)
h_m = Height of mobile antenna (1m – 10m)
d = Distance between transmitter and receiver (1 km – 20 km)
$a(h_m)$ = Mobile antenna correction factor:

For large cities and f ≥ 300 MHz:

$a(h_m) = 3.2 (\log_{10}(11.75 \cdot h_m))^2 - 4.97$

Otherwise:

$a(h_m) = 8.29 (\log_{10}(1.54 \cdot h_m))^2 - 1.1$

Suburban Area Adjustment:

$L_{suburban} = L_{urban} – 2 [\log_{10}(f / 28)]^2 - 5.4$

Source Code:

```
clc;
clear all;
close all;

% Predefined parameters
```

```
f = 900;     % frequency in MHz
d = 5;       % distance in km
hb = 50;     % base station antenna height in meters
hm = 1.5;    % mobile station antenna height in meters

% Correction factor for mobile antenna height (for urban large city, f ? 300 MHz)
if f >= 300
    a_hm = 3.2*(log10(11.75*hm))^2 - 4.97;
else
    a_hm = 8.29*(log10(1.54*hm))^2 - 1.1;
end

% Urban path loss calculation
L_urban = 69.55 + 26.16*log10(f) - 13.82*log10(hb) + (44.9 - 6.55*log10(hb)) * log10(d) -
a_hm;

% Suburban path loss calculation
L_suburban = L_urban - 2*(log10(f/28))^2 - 5.4;

% Display results
fprintf('Median Path Loss in Urban Environment: %.2f dB\n', L_urban);
fprintf('Median Path Loss in Suburban Environment: %.2f dB\n', L_suburban);
```

Calculation:

Given values:

Frequency, f = 900 MHz

Distance, d = 5 km

Base Station Height, h_b = 50 m

Mobile Station Height, h_m = 1.5 m

MATLAB Output Calculation:

```
Median Path Loss in Urban Environment: 146.96 dB

Median Path Loss in Suburban Environment: 137.02 dB
```

Result and discussion:

- The calculated path loss in urban areas is higher due to building density and greater signal obstruction.
- The suburban environment shows lower path loss as expected due to fewer obstructions and lower environmental complexity.
- This experiment confirms how different geographical environments impact radio wave propagation.
- Increasing frequency or distance increases path loss, while increasing antenna heights reduces it.

Experiment Number :07

Name of the Experiment: Write MATLAB code to calculate the average busy hour call attempts

Objectives:

- To understand the concept of Busy Hour Call Attempts (BHCA) in telecommunication networks.
- To calculate the average number of call attempts made during busy hours using MATLAB.
- To analyze network traffic intensity based on user behavior.

Equipment/Software Required:

A computer with MATLAB software installed.

Theory:

In telecommunications, Busy Hour Call Attempts (BHCA) refers to the average number of call attempts made during the busiest hour of the day. This metric is crucial for network capacity planning, switch sizing, and traffic engineering.

Key Definitions:

- **Call Attempt:** An attempt by a user to make a call, regardless of whether it is successful or not.
- **Busy Hour:** The one-hour period during which the highest number of call attempts are made.

Formula:

Average BHCA = $\frac{Total\ Call\ Attempts\ in\ a\ Day}{Numbers\ of\ busy\ hours}$

This assumes the busy hour traffic is evenly distributed among the defined busy hours.

Calculation:

Let:

- Total call attempts in a day = T
- Number of busy hours = H

Average BHCA = $\frac{T}{H}$;

Source code:

```
clc;
clear all;
close all;


% Input total number of call attempts in a day
```

```
total_call_attempts = input('Enter total number of call attempts in a day: ');

% Input number of busy hours in a day (typically 1 or 2 hours)
busy_hours = input('Enter number of busy hours in a day: ');

% Calculate average BHCA
average_BHCA = total_call_attempts / busy_hours;

% Display the result
fprintf('Average Busy Hour Call Attempts (BHCA) = %.2f calls\n', average_BHCA);
```

Input:

```
Enter total number of call attempts in a day: 150000
Enter number of busy hours in a day: 2
```

Output:

```
Average Busy Hour Call Attempts (BHCA) = 75000.00 calls
```

Result and Discussion:

- The code calculates the average BHCA based on total call attempts and busy hours.
- In this example, with 150,000 calls and 2 busy hours, the system observes 75,000 call attempts per busy hour.
- This metric helps telecom engineers decide:
    - ➢ How many lines or channels are needed.
    - ➢ What capacity a switch must handle.
    - ➢ How to manage peak-time network load effectively.

Experiment Number: 08

Name of the Experiment: Write a MATLAB code for calculating signal attenuation (path loss) over distance using Log Distance path loss Model.

Objective:

1. To understand how signal strength attenuates over distance.
2. To model and calculate path loss using the Log Distance Path Loss Model
3. To visualize the relationship between path loss and distance using MATLAB

Required Software: A Computer with Matlab Software.

Theory:

Log Distance Path Loss Model is theoretical and measurement-based propagation models indicate that average received signal power decrease logarithmically with distance, whether in outdoor or indoor radio channels. Wireless communication signals attenuate as they travel through space. This path loss depends on several factors such as distance, environment (urban, rural), and frequency. The Log Distance Path Loss Model is a simple yet powerful empirical model to estimate this attenuation in dB.

Log Distance Path Loss Equation:

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

Where:

$\boldsymbol{PL(d)}$ = Path loss at distance d in dB

$\boldsymbol{PL(d_0)}$ = Path loss at reference distance d_0

n = Path loss exponent (depends on environment)

d= Distance from transmitter

d_0 = Reference distance (usually 1 meter)

Environment	**Path Loss Exponent, n**
Free space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of-sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

Source code:

```
clc; clear;
```

```
% Parameters
fc = 2.4e9;            % Carrier frequency in Hz (2.4 GHz)
d0 = 1;                % Reference distance in meters
n = 3;                 % Path loss exponent
Pt = 20;               % Transmit power in dBm
% Distance range (in meters)
d = 1:0.5:100;         % From 1m to 100m
% Wavelength (in meters)
lambda = 3e8 / fc;
% Free-space Path Loss at reference distance d0
PL_d0 = 20*log10(4*pi*d0/lambda);
% Path Loss at distance d using log-distance model
PL_d = PL_d0 + 10 * n * log10(d/d0);
% Received Power at distance d
Pr_d = Pt - PL_d;
% Print only PL_d and Pr_d vectors
disp('Path Loss (dB):');
disp(PL_d);
disp('Received Power (dBm):');
disp(Pr_d);
```

Output:

```
Path Loss (dB):  40.0515   41.5731
Received Power (dBm): -20.0515 -21.5731
```

Result and Discussion:

In this experiment, we implemented the Log Distance Path Loss Model using MATLAB to understand how signal strength attenuates over distance. The simulation showed that path loss increases logarithmically as the distance between the transmitter and receiver grows. At a reference distance of 1 meter, the path loss was approximately 40 dB, and as the distance extended up to 100 meters, the path loss increased to over 80 dB. Consequently, the received signal power decreased from around -20 dBm to -64 dBm, clearly indicating the inverse relationship between distance and received power. This behavior aligns with theoretical expectations, where the path loss exponent (n = 3) modeled an urban environment.

Experiment no: 09

Name of the experiment: Write a MATLAB code for Amplitude Modulation

Objectives:

After completing this experiment, we will be able to

- Understand the principle of Amplitude Modulation (AM) in analog communication.
- Generate an AM wave using MATLAB by modulating a sinusoidal message signal with a high-frequency carrier.
- Visualize and analyze the message signal, carrier signal, and modulated AM signal.

Required Software:

- Computer or Laptop – to run MATLAB
- MATLAB or Octave Software

Theory:
Amplitude Modulation (AM) is a technique used in analog communication in which the amplitude of a high-frequency carrier wave is varied in proportion to the instantaneous amplitude of the message (information) signal.

Mathematical Expression of AM:

Let:

• Message signal, $m(t) = A_m \sin(2\pi f_m t)$

• Carrier signal, $c(t) = A_c \sin(2\pi f_c t)$

Then, the AM signal is given by:

- $s(t) = A_c \cdot [1 + \mu . m(t)] \cdot \sin(2\pi f_c t)$

Where:

• μ = Modulation index ($0 \le \mu \le 1$ for no distortion)

• A_c = Carrier amplitude

• f_c = Carrier frequency

• f_m = Message signal frequency

Source Code:

```
close all;
clc;
% AM Wave
```

```
t=0:.0001:.1;
a=2;
fm=50;
fc=1000;
m=0.9;
m1=sin(2*pi*fm*t);
c1=sin(2*pi*fc*t);
am=a*(1+m*m1).*c1;
figure

% Message Signal
subplot(3,1,1)
plot(t,m1);
xlabel('time')
ylabel('m1');
title('Information signal');

% Carrier Signal
subplot(3,1,2)
plot(t,c1);
xlabel('time')
ylabel('c1');
title('Carrier Signal');

% AM Wave
subplot(3,1,3)
plot(t,am);
xlabel('time')
ylabel('AM');
```

Calculation:

Given values:

Carrier amplitude (A) = 2

Message frequency (f_m) = 50 Hz

Carrier frequency (f_c) = 1000 Hz

Modulation index (μ) = 0.9

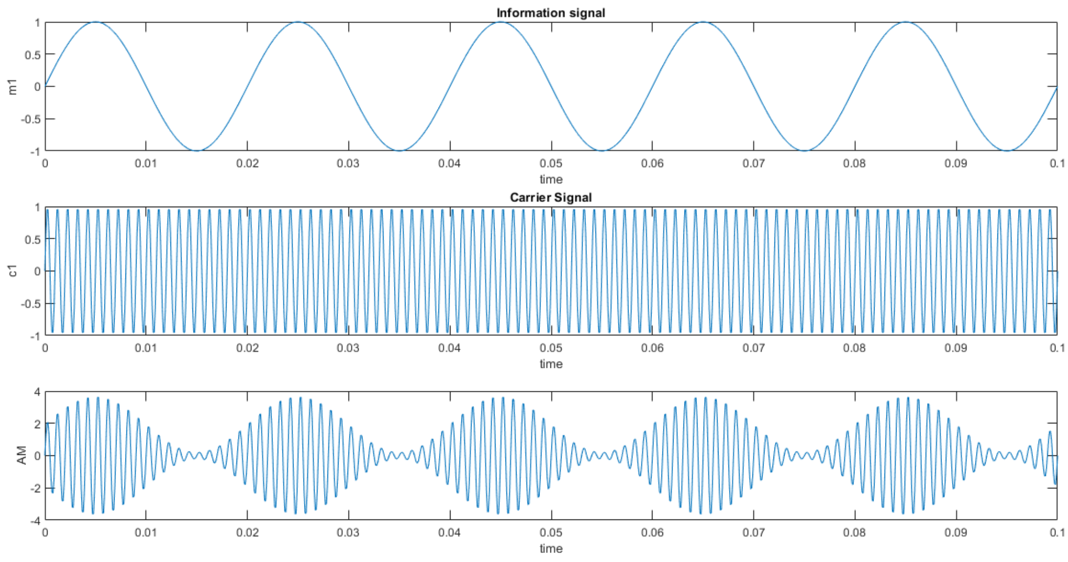
Time range: t = 0 to 0.1 sec with step size 0.0001

Message signal: $m(t) = \sin(2\pi \cdot 50 \cdot t)$

Carrier signal: $c(t) = \sin(2\pi \cdot 1000 \cdot t)$

AM signal: $am(t) = 2 \cdot (1 + 0.9 \cdot m(t)) \cdot c(t)$

Output:



Result & Discussion:

- The message signal is a low-frequency sine wave.
- The carrier signal is a high-frequency sine wave used to carry the message.
- The resulting AM signal varies in amplitude according to the message signal while maintaining the carrier frequency.

Observations from MATLAB plots:

1. The first subplot shows the message (information) signal.
2. The second subplot displays the carrier wave.
3. The third subplot clearly shows the amplitude-modulated waveform, where the envelope of the carrier wave follows the shape of the message signal.

Experiment No: 10

Name of the Experiment: Implementation of Frequency Division Multiple Access (FDMA) in MATLAB.

Objective:

1. To understand the concept of Frequency Division Multiple Access (FDMA).
2. To implement a basic FDMA system using MATLAB.
3. To observe how multiple users can share a communication channel using different frequency bands.

Theory:

Frequency Division Multiple Access (FDMA) is a channel access method used in multiple-access protocols. In FDMA, the available bandwidth is divided into multiple frequency bands. Each user is assigned a specific frequency band, and all users transmit simultaneously without interfering with one another because they are on different frequencies.

Key Points:

- Users transmit at the same time but on different frequencies.
- Guard bands are used between channels to prevent overlap.
- Used in analog systems like radio broadcasting and first-generation mobile systems.

Example:
If we have 3 users and a total bandwidth of 12 kHz, we can allocate 4 kHz to each user (with or without guard bands).

Calculation:

Assume:

- Total Bandwidth = 12 kHz
- Number of Users = 3
- Bandwidth per user = 4 kHz

Let's simulate:

- Each user transmits a sine wave at a unique carrier frequency:
    - User 1 → 2 kHz
    - User 2 → 6 kHz
    - User 3 → 10 kHz

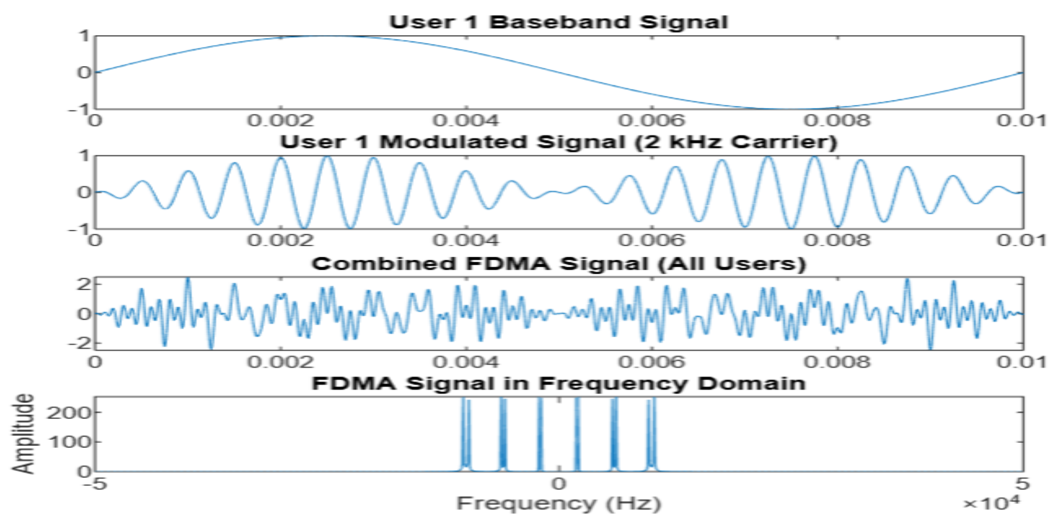
Each user sends a baseband signal (e.g., `sin(2πf_base t)`), modulated onto a carrier frequency.

Source Code:

```
clc;
clear;
% Time axis
fs = 100000;          % Sampling frequency
t = 0:1/fs:0.01;      % Time vector (0 to 10 ms)
% Baseband signals (low frequency sine waves)
s1 = sin(2*pi*100*t);     % User 1 data
s2 = sin(2*pi*200*t);     % User 2 data
s3 = sin(2*pi*300*t);     % User 3 data
% Carrier frequencies
f1 = 2000;   % User 1
f2 = 6000;   % User 2
f3 = 10000;  % User 3
% Modulated signals
mod1 = s1 .* cos(2*pi*f1*t);
mod2 = s2 .* cos(2*pi*f2*t);
mod3 = s3 .* cos(2*pi*f3*t);
% Combined FDMA signal
fdma_signal = mod1 + mod2 + mod3;
% Plotting

figure;
subplot(4,1,1);
plot(t, s1);
title('User 1 Baseband Signal');
subplot(4,1,2);
plot(t, mod1);
title('User 1 Modulated Signal (2 kHz Carrier)');

subplot(4,1,3);
plot(t, fdma_signal);
```

```
title('Combined FDMA Signal (All Users)');

subplot(4,1,4);
% Frequency domain view
n = length(fdma_signal);
f = (-n/2:n/2-1)*(fs/n);
fdma_fft = abs(fftshift(fft(fdma_signal)));

plot(f, fdma_fft);
title('FDMA Signal in Frequency Domain');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
```

Output:



Result and Discussion:

- The simulation clearly shows how FDMA allows multiple users to transmit simultaneously over separate frequency bands.
- The time-domain plots illustrate how each user's data is modulated to a unique carrier.
- The frequency-domain plot verifies that all three signals occupy separate bands, minimizing interference.
- This lab helps visualize the concept of frequency multiplexing in practical communication systems.

Experiment No: 11

Experiment Name: Implementation of Time Division Multiple Access (TDMA) in MATLAB.

Objectives:

1. To understand the working principle of TDMA.
2. To observe how multiple users share the same frequency channel through time slot division.
3. To simulate a basic TDMA-based communication system.
4. To learn how time slots are allocated to each user in TDMA.

Required Software:

- MATLAB (R2017b or later)

Theory:

Time Division Multiple Access (TDMA) is a technique where multiple users share the same frequency channel, but each is allocated a different time slot for communication. Instead of transmitting simultaneously, users transmit one after another in their assigned time slots. This time-based separation allows multiple users to use the same bandwidth without interference.

Applications of TDMA:

- GSM (2G Mobile Systems)
- Satellite Communication
- Digital Trunked Radio Systems

Mathematical Equations:

1. **Total Time Frame:**

   $$T_{frame} = N \times T_{slot}$$

   Where-

   N = Number of users

   Tslot = Time duration of each slot

2. **Time Slot Allocation for User *i*:**

   $$T_i = i \times T_{slot} \text{ , for } i = 0, 1, 2, ..., N-1$$

Calculation:

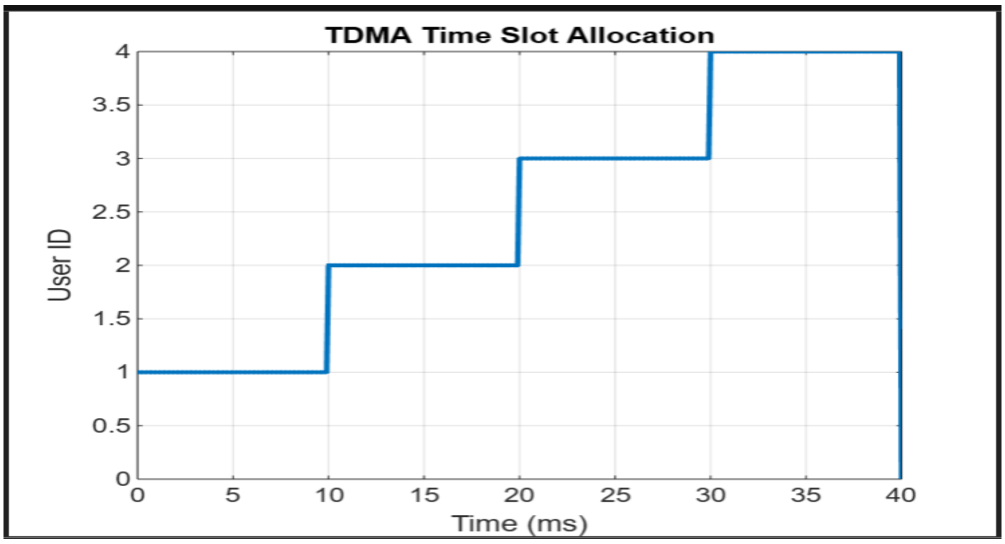
1. Assume there are N=4 users.
2. Assign a time slot of 10ms for each user.
3. Let each user transmit a unique signal only during its assigned time slot.

4. Use MATLAB to simulate the transmission and visualize the time slot allocations.

Source Code:

```
clc;
clear;
% Number of users
N = 4;
% Time per slot (in ms)
T_slot = 10;
% Total time
T_total = N * T_slot;
t = 0:0.1:T_total;  % time axis
signal = zeros(size(t));
% Assign signal values for each user's time slot
for i = 1:N
    start_time = (i - 1) * T_slot;
    end_time = i * T_slot;
    idx = find(t >= start_time & t < end_time);
    signal(idx) = i;  % user i transmits
end
% Plotting
figure;
plot(t, signal, 'LineWidth', 2);
xlabel('Time (ms)');
ylabel('User ID');
title('TDMA Time Slot Allocation');
grid on;
```

Output:



Result and Discussion: The TDMA simulation successfully demonstrates how multiple users can share the same communication channel by transmitting in distinct time slots without overlapping.

- TDMA is simple and bandwidth-efficient.
- If a user doesn't transmit during its slot, the time is wasted — no reuse by others.
- Precise synchronization is required for proper functioning.
- TDMA is not ideal for bursty or asynchronous data transmission.

Experiment Name: 12

Experiment Name: Implementation of Code Division Multiple Access (CDMA) in MATLAB

Objectives:

1. To understand the concept and working of Code Division Multiple Access (CDMA).
2. To simulate CDMA transmission and decoding using MATLAB.
3. To observe how multiple users can transmit simultaneously over the same channel using unique codes.
4. To demonstrate orthogonality of spreading codes for noise-free transmission.

Required Software:

- MATLAB (R2015 or later)
- Optional: Signal Processing Toolbox (for advanced analysis)

Theory:

CDMA (Code Division Multiple Access) is a channel access method where multiple users transmit simultaneously over the same frequency and time using unique orthogonal codes to distinguish their signals. Each user's data is spread using a unique code sequence (e.g., Walsh codes, PN sequences), and all users transmit their spread signals together. At the receiver, the signal is despread using the corresponding code to retrieve the original data.

Applications:

- 3G Cellular Networks (W-CDMA)
- GPS (Global Positioning System)
- Satellite Communications

Mathematical Equations:

1. Spread Signal:

$$Si(t)=Di(t)\cdot Ci(t)$$

Where:

-Di(t): Data signal of user i

-Ci(t): Code assigned to user i

2. Combined CDMA Signal:

$$S(t) = \sum Si(t) = \sum Di(t)\cdot Ci(t)$$

3. Despreading at Receiver (User j):

$$Dj(t) = S(t)\cdot Cj(t)$$

Then average the signal over the chip length.

Procedure:

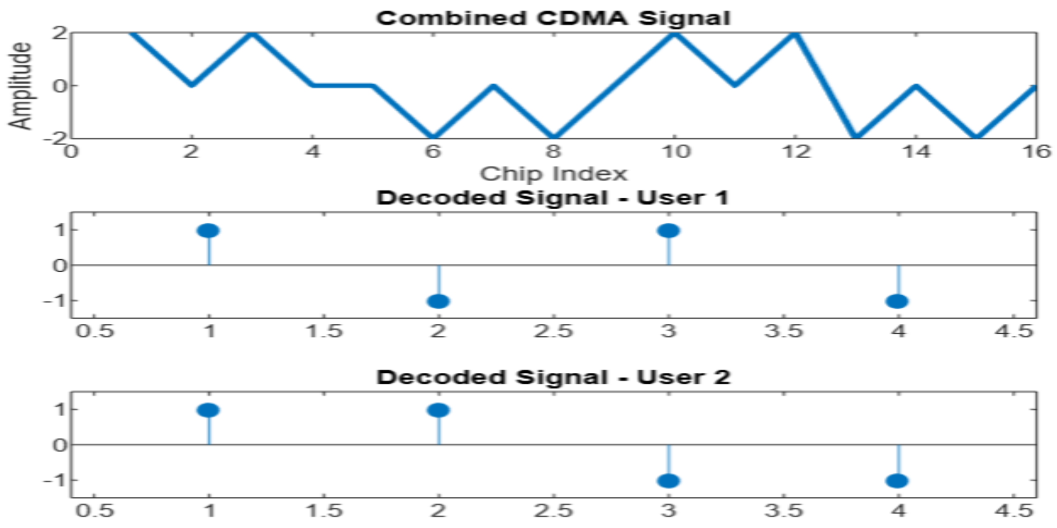
1. Assign binary data to 2 or more users (e.g., 1 and -1).
2. Use orthogonal codes (e.g., Walsh codes) for each user.
3. Multiply each user's data with their unique code → spread signal.
4. Add all spread signals → combined channel signal.
5. Multiply the combined signal with the same user's code to retrieve their data (despread).

Source Code:

```
clc;
clear;
% Binary data for 2 users (1 = +1, 0 = -1)
data_user1 = [1 -1 1 -1];  % bits
data_user2 = [1 1 -1 -1];
% Walsh codes (orthogonal)
code_user1 = [1 1 1 1];
code_user2 = [1 -1 1 -1];
% Length of one bit in time units
bit_len = length(code_user1);
% Spread signals
spread_user1 = kron(data_user1, code_user1);  % user1 signal spreading
spread_user2 = kron(data_user2, code_user2);  % user2 signal spreading
% CDMA combined signal (simultaneous transmission)
cdma_signal = spread_user1 + spread_user2;

% Receiver tries to decode user1
recv_user1 = cdma_signal .* repmat(code_user1, 1, length(data_user1));
decoded_user1 = [];
for i = 1:bit_len:length(recv_user1)
   avg = sum(recv_user1(i:i+bit_len-1))/bit_len;
   decoded_user1 = [decoded_user1, sign(avg)];
end
```

```
% Receiver tries to decode user2
recv_user2 = cdma_signal .* repmat(code_user2, 1, length(data_user2));
decoded_user2 = [];
for i = 1:bit_len:length(recv_user2)
   avg = sum(recv_user2(i:i+bit_len-1))/bit_len;
   decoded_user2 = [decoded_user2, sign(avg)];
end
% Display results
disp('Original Data User 1:'); disp(data_user1);
disp('Decoded Data User 1:'); disp(decoded_user1);

disp('Original Data User 2:'); disp(data_user2);
disp('Decoded Data User 2:'); disp(decoded_user2);

% Plotting
figure;
subplot(3,1,1);
plot(cdma_signal, 'LineWidth', 2);
title('Combined CDMA Signal');
xlabel('Chip Index');
ylabel('Amplitude');

subplot(3,1,2);
stem(decoded_user1, 'filled'); title('Decoded Signal - User 1');
ylim([-1.5 1.5]);

subplot(3,1,3);
stem(decoded_user2, 'filled'); title('Decoded Signal - User 2');
ylim([-1.5 1.5]);
```

Output:



Result and Discussion: The MATLAB simulation successfully demonstrates the working of CDMA where:

- Two users transmit simultaneously over the same channel.
- Unique spreading codes allow each user to be decoded independently and accurately.

  CDMA allows simultaneous and asynchronous multi-user communication.
- Orthogonal spreading codes are critical to prevent interference.
- The simulation proves that CDMA provides better resource utilization than FDMA or TDMA.
- CDMA is highly resistant to narrowband interference and supports secure communication