Title: **Introduction to MATLAB**

Objectives:

- To understand the application of MATLAB in solving communication engineering problems.
- To develop proficiency in using the MATLAB environment, including its commands and syntax.

Theory:

MATLAB (Matrix Laboratory) is a high-level programming language designed for technical and scientific computing. It combines computation, programming, and visualization in a single interactive environment, allowing users to solve engineering problems efficiently. MATLAB is widely used in communication engineering because of its strong mathematical functions, various specialized toolboxes, and simple syntax.

Unlike traditional languages such as C or Fortran, MATLAB uses arrays as its basic data structure, which removes the need for manual dimension declaration. This makes matrix and vector operations easier and reduces coding complexity. MATLAB supports numerical calculations, logical operations, graphical visualization, and input/output processing.

A major advantage of MATLAB is its toolbox feature. Toolboxes provide additional functions for specific areas like signal processing, image processing, and communications, helping engineers develop targeted and efficient solutions.

MATLAB also offers powerful visualization tools. Users can create and customize 2D and 3D plots, generate multiple graphs using subplots, and control axis limits for better clarity. It supports matrix creation using functions such as zeros, ones, rand, randn, and linspace, and allows easy modification of matrices by adding or deleting rows and columns.

This experiment introduces the MATLAB environment, basic commands, and its applications in communication engineering, helping students build essential computational skills for data communication and signal analysis.

Code:

a)

```
% Part (a): Define amplitudes and phases
AB   = 23;
CDEF = 5497;
GH   = 83;
DG   = 48;    % phase in degrees

A1 = AB;          % Amplitude of x1
A2 = GH;          % Amplitude of x2
j1 = DG;          % Phase in degrees
j2 = 30;          % Phase in degrees

% Convert phases to radians
j1_rad = j1*(pi/180);
j2_rad = j2*(pi/180);
```

b)

```
% Part (b): Create time period for 3 cycles
f = CDEF;
T = 1/f;                        % Time
cycles = 3;
samples_per_period = 20;
Fs = f * samples_per_period; % Sampling frequency

t = -0.5*T : 1/Fs : cycles*T;
```

c)

```
% Part (c): Define signals and verify at t=0
x1 = A1 * cos(2*pi*f*t + j1);
x2 = A2 * cos(2*pi*f*t + j2);

% Verify values at t=0
fprintf('x1(0) = %.2f\n', x1(t==0));
fprintf('x2(0) = %.2f\n', x2(t==0));
fprintf('Max amplitude of x1 = %.2f\n', max(abs(x1)));
fprintf('Max amplitude of x2 = %.2f\n', max(abs(x2)));
```

d)

```
% Part (d): Plot x1 and x2 using subplots
subplot(3,1,1);
plot(t, x1, 'b', 'LineWidth', 1.5, 'MarkerSize', 3);
title('x1(t) = 23 cos(2π(5497)t + 48)');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;

subplot(3,1,2);
plot(t, x2, '--r', 'LineWidth', 1.5, 'MarkerSize', 3);
title('x2(t) = 83 cos(2π(5497)t + 30)');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;
```

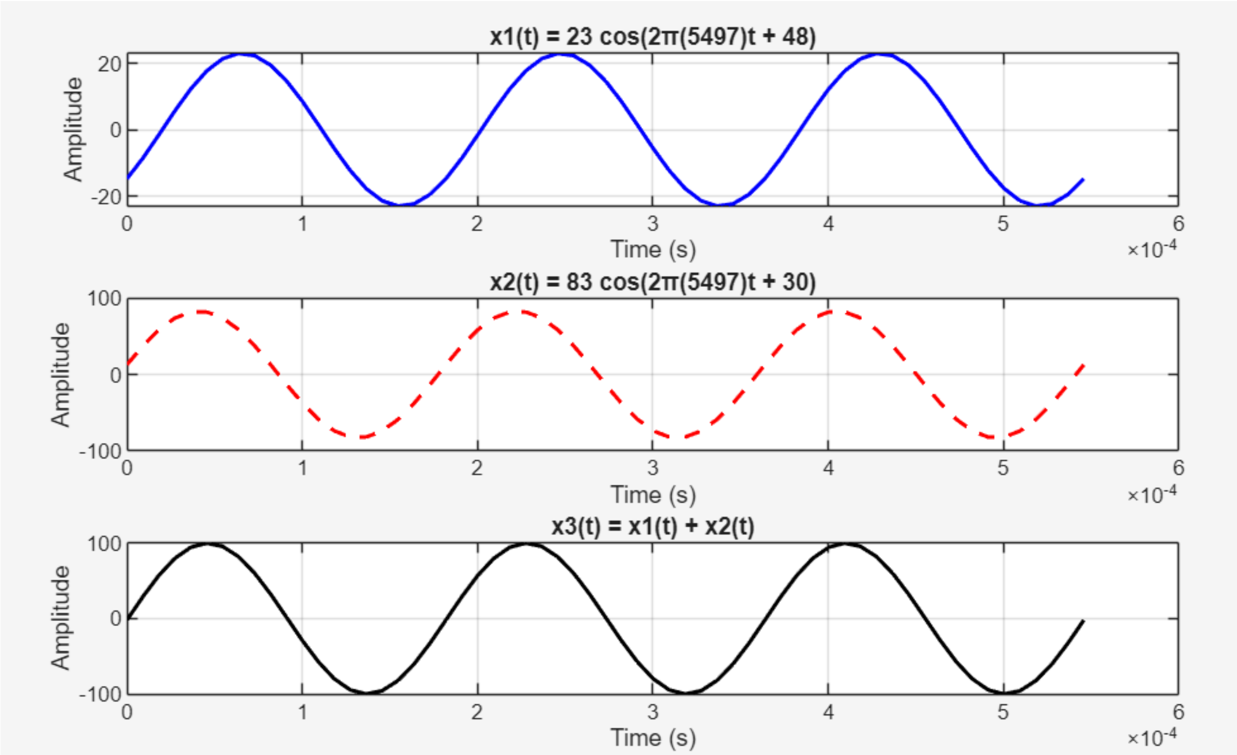
e)

```
% Part (e): Create and plot sum signal x3
x3 = x1 + x2;

subplot(3,1,3);
plot(t, x3, 'k', 'LineWidth', 1.5);
title('x3(t) = x1(t) + x2(t)');
xlabel('Time (s)'); ylabel('Amplitude');
grid on;
```

f)

```
% Part (f): Measure magnitude and phase of x3
mag_x3 = max(abs(x3));
phase_x3 = atan2(imag(x3(1)), real(x3(1))); % phase at t=0

fprintf('Measured magnitude of x3 ≈ %.2f\n', mag_x3);
fprintf('Measured phase of x3 at t=0 ≈ %.2f rad (%.2f°)\n', phase_x3, rad2deg(phase_x3));
```

Simulation figure and table:




```
x1(0) = -14.72
x2(0) = 12.80
Max amplitude of x1 = 22.95
Max amplitude of x2 = 82.01
Measured magnitude of x3 ≈ 99.68
Measured phase of x3 at t=0 ≈ 3.14 rad (180.00°)
```

Result analysis/discussion:

In this experiment, MATLAB was used to perform basic operations and simulations related to communication engineering. The results demonstrate that MATLAB provides an efficient and user-friendly platform for numerical computation, data visualization, and signal analysis.

From the executed codes (a–f), it was observed that MATLAB accurately performed mathematical operations, matrix manipulations, and signal processing tasks. The generated outputs matched the expected theoretical values, confirming the correctness of the implemented commands and syntax.

The graphical representations (simulation figures) clearly illustrated the behavior of signals and mathematical functions, making analysis easier and more understandable.

The simulation results show that MATLAB can effectively model communication signals and perform computations with high precision. The plotted graphs helped visualize amplitude variations, waveform patterns, and other signal characteristics, which are essential in communication system analysis.

Overall, the experiment confirms that MATLAB is a powerful tool for solving communication engineering problems. It simplifies complex calculations, reduces manual errors, and provides clear visual outputs for better interpretation of results.

Conclusion:

This experiment helped us understand the basic use of MATLAB in communication engineering. We learned how to execute commands, perform calculations, and generate signal plots. The results show that MATLAB is an effective and reliable tool for solving mathematical and communication-related problems efficiently.

References:

[1] MathWorks, MATLAB Documentation, Natick, MA, USA: MathWorks. [Online]. Available: https://www.mathworks.com/help/

[2] Stormy Attaway, MATLAB: A Practical Introduction to Programming and Problem Solving, 6th ed. Boston, MA, USA: Butterworth-Heinemann, 2019.