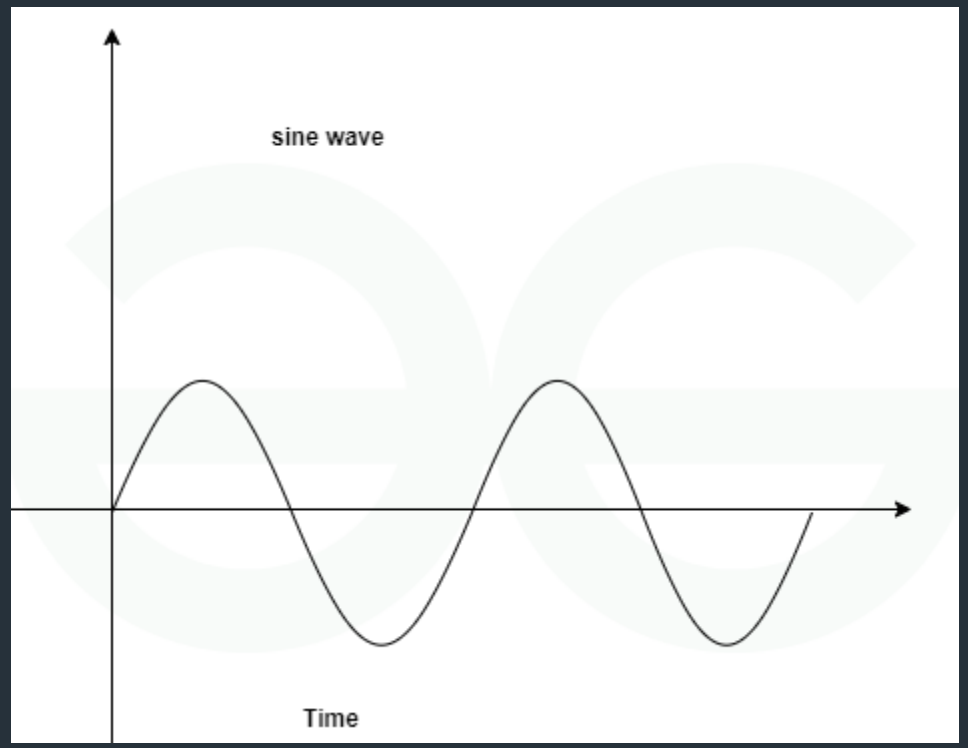
Introduction to Matlab

Course Title: Data Communication Sessional

Course Code: cse-322

1.Sinusoidal Signal[analog]

$y(t) = A \sin(2\pi ft + \theta)$




```
clc; clear all; close all;

Fs = 1000;
t = 0:1/Fs:1;
f = 5;
A = 2;

x = A*sin(2*pi*f*t);

plot(t,x);
xlabel('Time (s)');
ylabel('Amplitude');
title('Sinusoidal Signal');
grid on;
```

In MATLAB, these three commands are used to clear the workspace and command window:

1.clc:

- Clears the Command Window, removing all text and output displayed there.
- This is useful for cleaning up the screen and improving readability.

2.clear all:

- Removes all variables from the workspace.
- This is helpful when you want to start a new session without any existing variables interfering with your calculations.

3.close all:

- Closes all open figures.
- This is useful when you have multiple figures open and want to clear them all at once.

Parameter Definition:

- Fs: Sampling frequency determines how many samples are taken per second.
- t: Time vector, created using 0:1/Fs:1, spans from 0 to 1 second with steps of 1/Fs.
- f: Frequency of the sinusoidal signal.
- A: Amplitude of the sinusoidal signal.

Plotting the Signal:

- plot(t,x): Plots the signal x against time t.
- xlabel, ylabel, and title: Add labels to the x-axis, y-axis, and the plot title, respectively.
- grid on: Turns on the grid lines for better readability.

2. Digital Signal in MATLAB

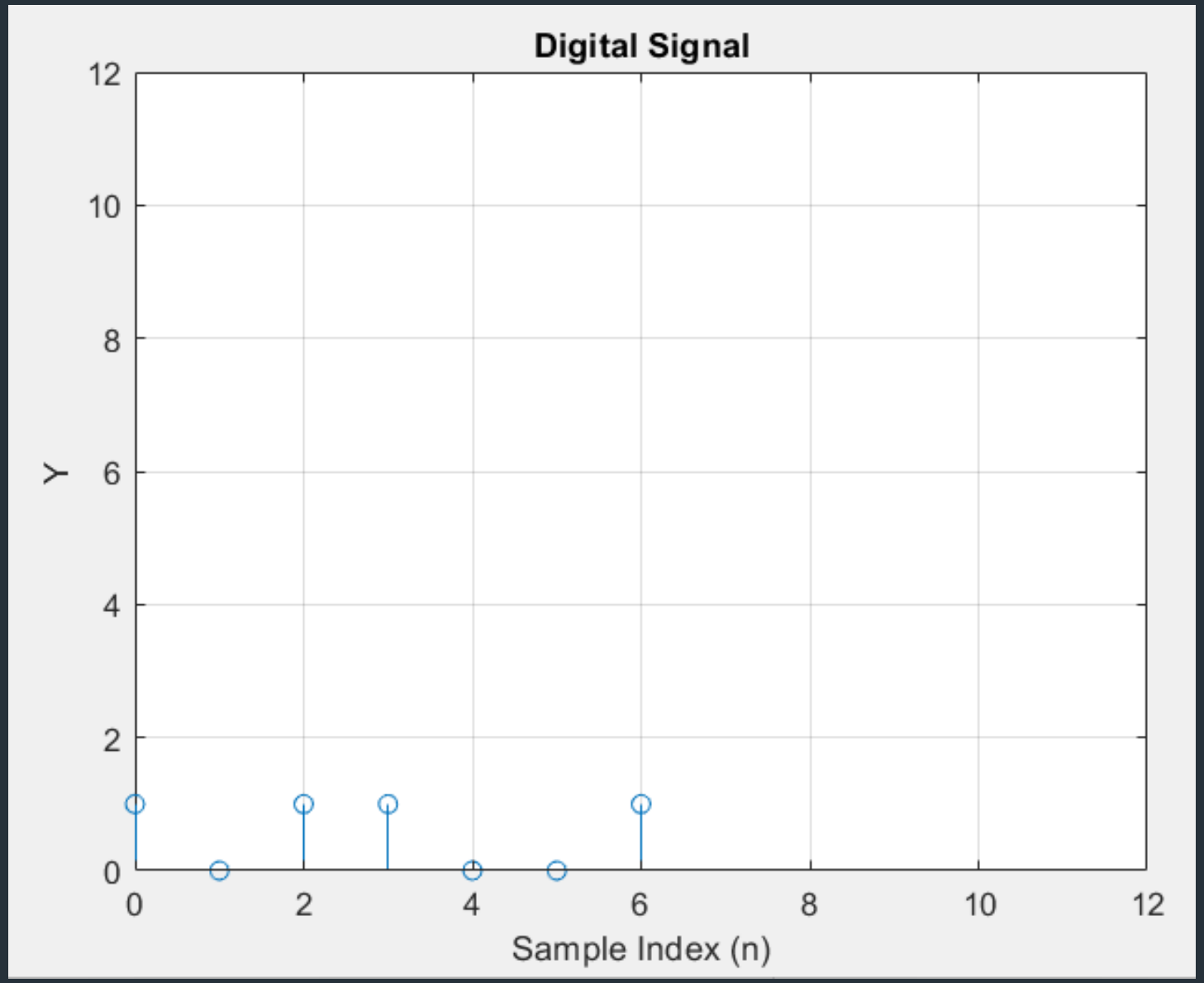
```
clc; clear all; close all;
x = [1 0 1 1 0 0 1];

n = 0:length(x)-1;

stem(n, x);
axis([0 12 0 12]);

xlabel('Sample Index (n)');
ylabel('Amplitude');
title('Digital Signal');

grid on;
```



- **Defining the Signal:** We create a vector x to represent the digital signal. Each element in x corresponds to a sample of the signal. Creating the Time Vector: The n vector represents the sample indices, starting from 0.

- **Plotting the Signal:** stem(n, x): This command creates a stem plot, where the vertical lines represent the amplitude of each sample at the corresponding time index.

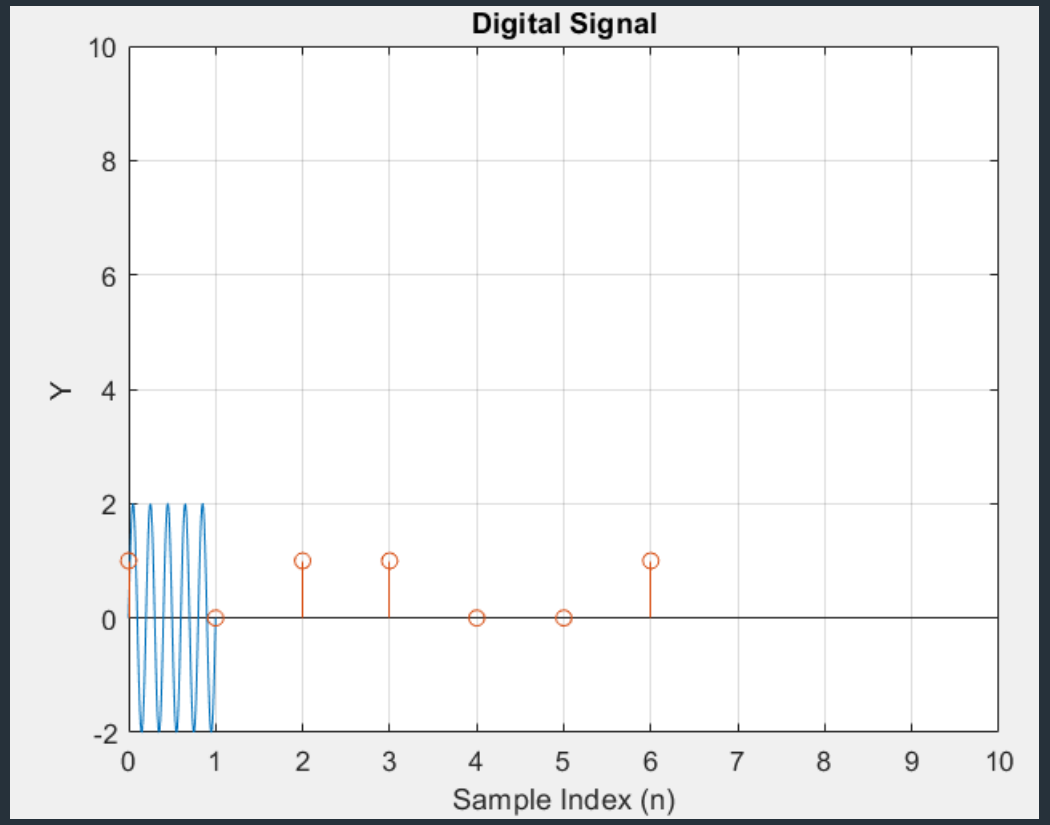
- **Adding Labels:** xlabel, ylabel, and title are used to label the x-axis, y-axis, and the entire plot, respectively.

- **Customizing the Plot:**

grid on: This command adds a grid to the plot for better readability.

3.Use of hold

```
clc; clear all; close all;
%For analog signal
Fs = 1000;
t = 0:1/Fs:1;
f = 5;
A = 2;
x = A*sin(2*pi*f*t);
plot(t,x);
xlabel('Time (s)');
ylabel('Amplitude');
title('Sinusoidal Signal');
grid off;
hold on;
%For digital Signal
x = [1 0 1 1 0 0 1];
n = 0:length(x)-1;
stem(n, x);
axis([0 10 0 10]);
xlabel('Sample Index (n)');
ylabel('Y');
title('Digital Signal');
grid on;
```

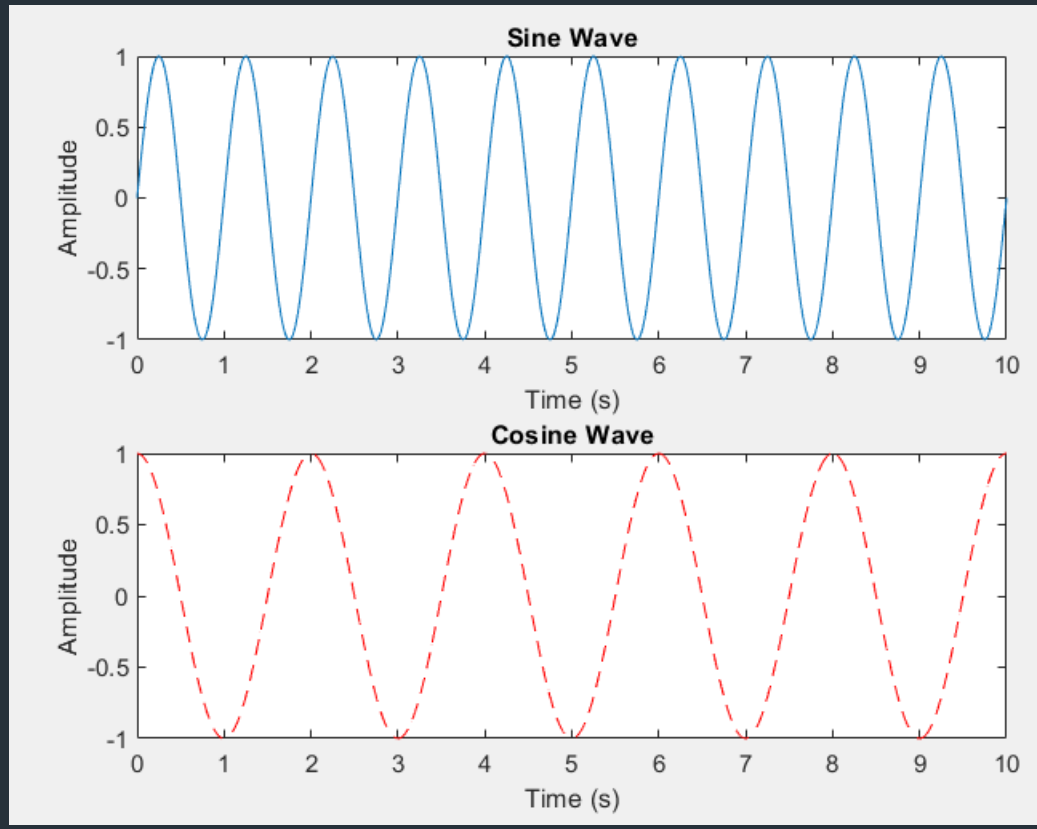


4.Use of subplot[two]

```
clc; clear all; close all;
t = 0:0.01:10;
x1 = sin(2*pi*t);
x2 = cos(2*pi*t/2);

subplot(2,1,1);
plot(t, x1);
xlabel('Time (s)');
ylabel('Amplitude');
title('Sine Wave');

subplot(2,1,2);
plot(t, x2, 'r--');
xlabel('Time (s)');
ylabel('Amplitude');
title('Cosine Wave');
```



Explanation:

- subplot(2,1,1); divides the figure into a 2x1 grid and selects the first subplot.
- subplot(2,1,2); selects the second subplot.
- The plot commands in each subplot plot the respective signals.

plot(t, x2, 'r--');

In this MATLAB code, 'r--' is a format string that specifies the color and line style of the plot.

- **r**: This specifies the color of the line, which is red.
- **--**: This specifies the line style, which is a dashed line.

So, 'r--' tells MATLAB to plot the data with a red dashed line.

```
plot(x, y1, 'b--'); % Blue dashed line
plot(x, y2, 'r:');  % Red dotted line
plot(x, y3, 'g-.'); % Green dash-dot line
```

Here are some common color and line style formats you can use in MATLAB's plot function:

Colors:

- **b**: Blue
- **g**: Green
- **r**: Red
- **c**: Cyan
- **m**: Magenta
- **y**: Yellow
- **k**: Black
- **w**: White

Line Styles:

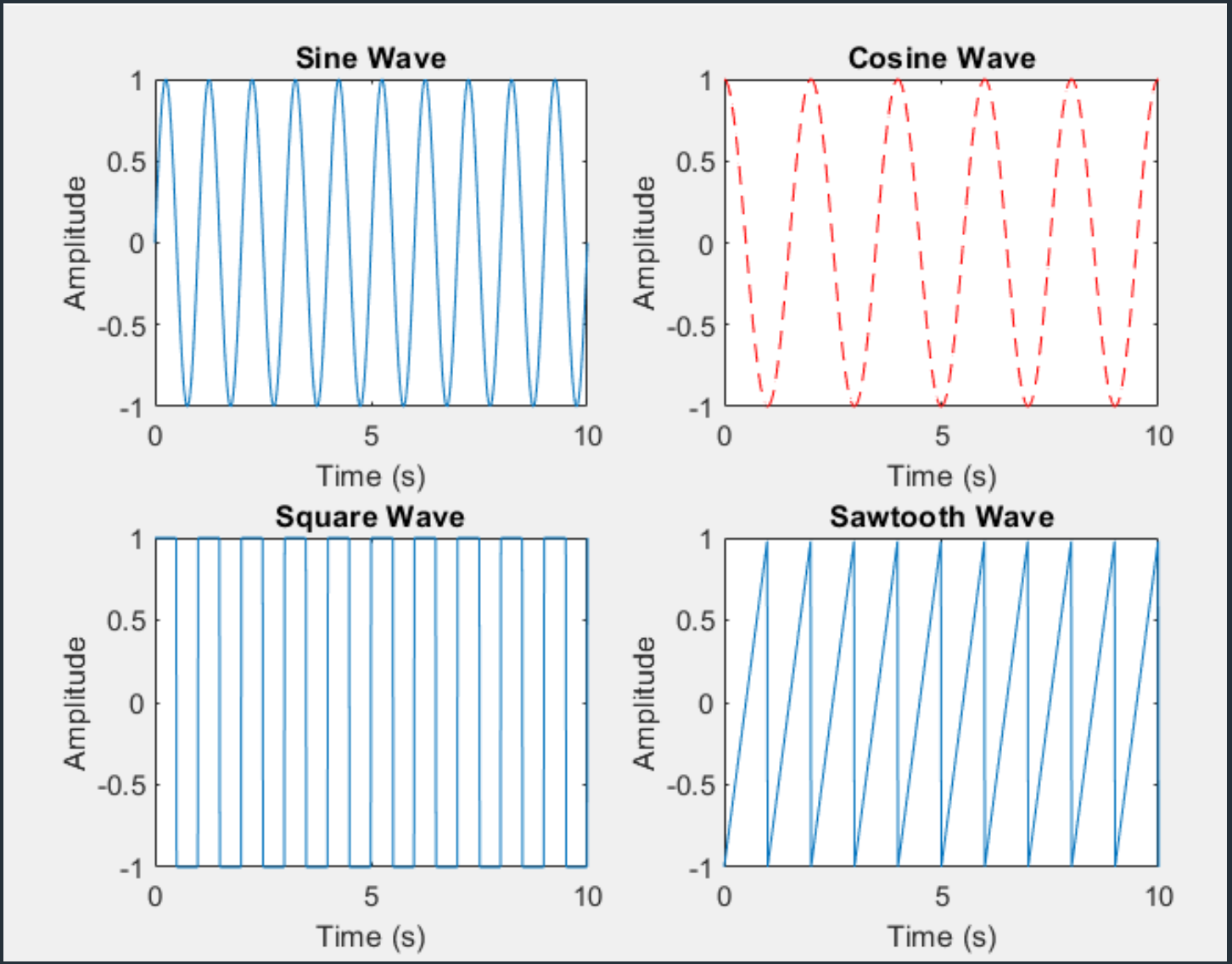
- **-**: Solid line (default)
- **--**: Dashed line
- **:**: Dotted line
- **-.**: Dash-dot line

Combining Color and Line Style:

You can combine color and line style in a single format string:

- **b--**: Blue dashed line
- **r:**: Red dotted line
- **g-.**: Green dash-dot line

5.Use of subplot[four]



Task-
Draw 6 analog signal using subplot

Use of subplot[four]

```
% Time vector
t = 0:0.01:10;

% Four different signals
x1 = sin(2*pi*t);
x2 = cos(2*pi*t/2);
x3 = square(2*pi*t);
x4 = sawtooth(2*pi*t);

% Create a new figure
figure;

% Plot the first signal in the top-left subplot
subplot(2,2,1);
plot(t, x1);
xlabel('Time (s)');
ylabel('Amplitude');
title('Sine Wave');

% Plot the second signal in the top-right subplot
subplot(2,2,2);
plot(t, x2, 'r--');
xlabel('Time (s)');
ylabel('Amplitude');
title('Cosine Wave');

% Plot the third signal in the bottom-left subplot
subplot(2,2,3);
plot(t, x3);
xlabel('Time (s)');
ylabel('Amplitude');
title('Square Wave');

% Plot the fourth signal in the bottom-right subplot
subplot(2,2,4);
plot(t, x4);
xlabel('Time (s)');
ylabel('Amplitude');
title('Sawtooth Wave');
```

Task: Plot Signals in Different Figure

6.Signal Operations: Addition and Subtraction:

```
t = 0:0.01:10;
x1 = sin(2*pi*5*t);
x2 = cos(2*pi*3*t);
x3 = x1 + x2;
x4 = x1 - x2;
plot(t,x1,t,x2,t,x3,t,x4);
legend('x1','x2','x1+x2','x1-x2');
```

Task:

1. Draw 1 sine wave, 1 cos wave and 1 addition of these two-wave using subplot
2. Draw 1 sine wave, 1 cos wave and 1 subtract of these two-wave using subplot

7.Multiplication and Division:

```
t = 0:0.01:10;
x1 = sin(2*pi*5*t);
x2 = cos(2*pi*3*t);
x3 = x1 .* x2;
x4 = x1 ./ x2;
plot(t,x1,t,x2,t,x3,t,x4);
legend('x1','x2','x1*x2','x1/x2');
```

Task:

1. Draw 1 sine wave, 1 cos wave and 1 multiplication of these two-wave using subplot

2. Draw 1 sine wave, 1 cos wave and 1 division of these two-wave using subplot