SIGNALS AND SYSTEMS LAB (CSE-301L)

Spring 2024, 4th Semester

Lab Report 06

Submitted by**: Hassan Zaib Jadoon**

Registration Number**: 22PWCSE2144**

Section: **A**

“On my honor, as a student at the University of Engineering and Technology Peshawar, I have neither given nor received unauthorized assistance on this academic work.”

Recoverable Signature

X Hassan

Signature: Hassan Zaib Jadoon
Student
Signed by: 506ecb3d-aa56-4353-aa60-5abf648a668d

Submitted To: Dr. Safdar Nawaz Khan Marwat

Department of Computer Systems Engineering
University of Engineering and Technology Peshawar

Lab 06:

Task 1: Generate a 1 × 10 row vector v whose ith component is cos(iπ/4).

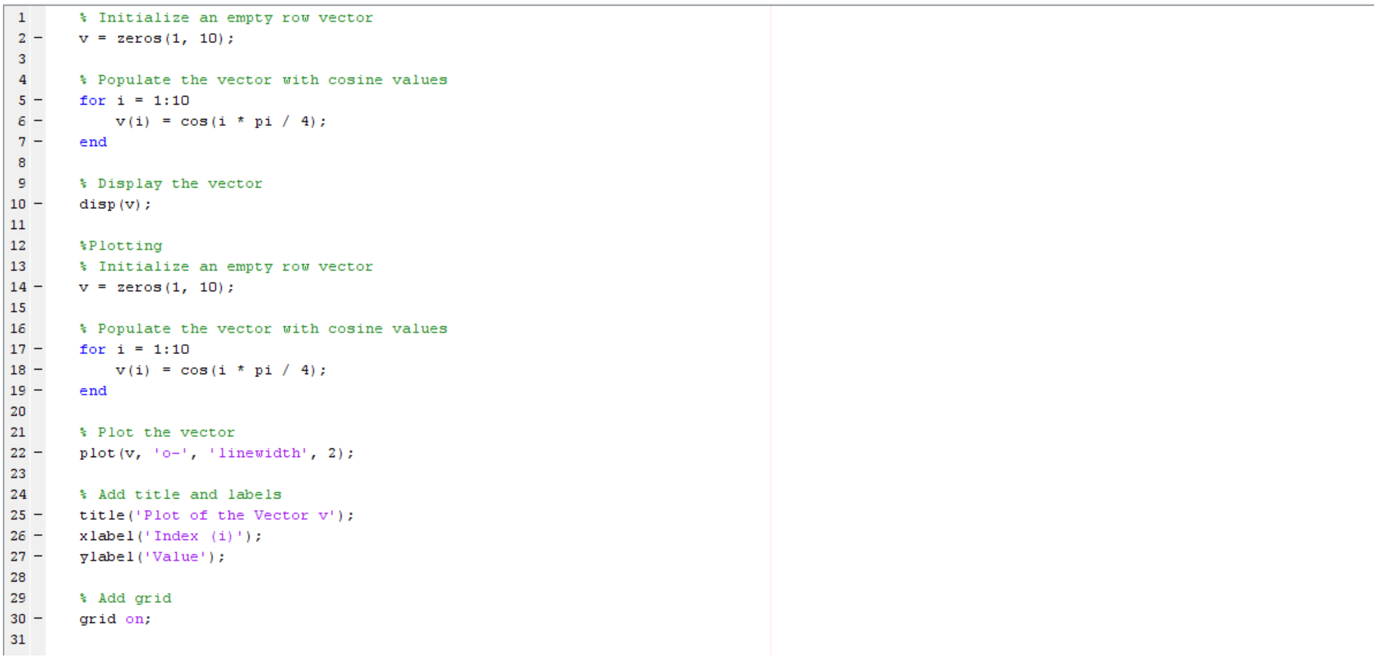
Problem Statement Task 1:

Generate a 1 × 10 row vector **v** where each component **v[i]** is equal to **cos(iπ/4)**.

Algorithm Task 1:

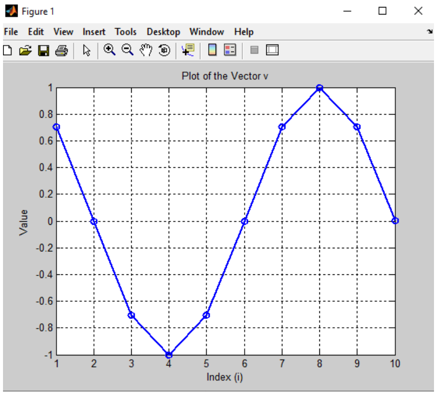
1. Initialize an empty row vector **v**.
2. Iterate **i** from 1 to 10.
3. Calculate the **i**th component of the vector **v** as **cos(iπ/4)**.
4. Store the calculated value in the **i**th position of the vector **v**.
5. Return the vector **v**.

Code:



```
% Initialize an empty row vector
v = zeros(1, 10);

% Populate the vector with cosine values
for i = 1:10
    v(i) = cos(i * pi / 4);
end

% Display the vector
disp(v);

%Plotting
% Initialize an empty row vector
v = zeros(1, 10);

% Populate the vector with cosine values
for i = 1:10
    v(i) = cos(i * pi / 4);
end

% Plot the vector
plot(v, 'o-', 'linewidth', 2);

% Add title and labels
title('Plot of the Vector v');
xlabel('Index (i)');
ylabel('Value');

% Add grid
grid on;

```

Output:



Conclusion:

The algorithm successfully generates a row vector **v** where each component represents the cosine of **iπ/4** for **i** ranging from 1 to 10.

Task 2: Write a MATLAB code that draws a graph of sin(nπx) on the interval - 1 ≤ x ≤ 1 for n = 1, 2, 3, ... , 8. (Hint: Use for loop)

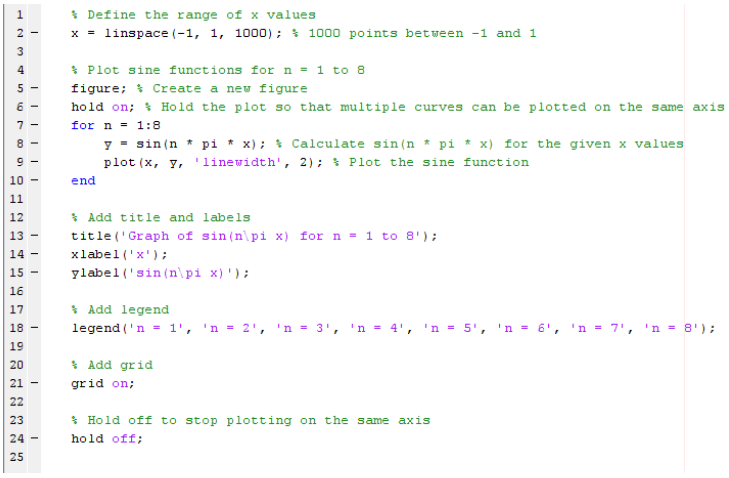
Problem Statement:

Write MATLAB code to draw a graph of sin(nπx) on the interval -1 ≤ x ≤ 1 for n = 1, 2, 3, ... , 8.

Algorithm:

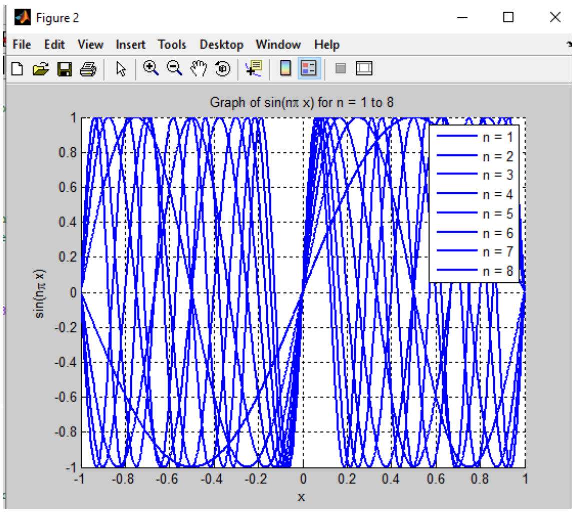
1. Define the interval x from -1 to 1.
2. For each value of n from 1 to 8:
   - Generate the function sin(nπx).
   - Plot the function on the same graph.
3. Label the axes appropriately.
4. Display the graph.

Code:



```
% Define the range of x values
x = linspace(-1, 1, 1000); % 1000 points between -1 and 1

% Plot sine functions for n = 1 to 8
figure; % Create a new figure
hold on; % Hold the plot so that multiple curves can be plotted on the same axis
for n = 1:8
    y = sin(n * pi * x); % Calculate sin(n * pi * x) for the given x values
    plot(x, y, 'linewidth', 2); % Plot the sine function
end

% Add title and labels
title('Graph of sin(n\pi x) for n = 1 to 8');
xlabel('x');
ylabel('sin(n\pi x)');

% Add legend
legend('n = 1', 'n = 2', 'n = 3', 'n = 4', 'n = 5', 'n = 6', 'n = 7', 'n = 8');

% Add grid
grid on;

% Hold off to stop plotting on the same axis
hold off;
```

Output:



Conclusion:

The MATLAB code successfully plots the sine functions sin(nπx) for n ranging from 1 to 8 on the interval -1 ≤ x ≤ 1, providing insight into the behavior of sinusoidal functions with increasing frequency.

Task 3: Given the signal exp(- x)sin(8x) for 0 ≤ x ≤ 2π, plot its continuous-time and discrete-time representations. Use the subplot and label properly.

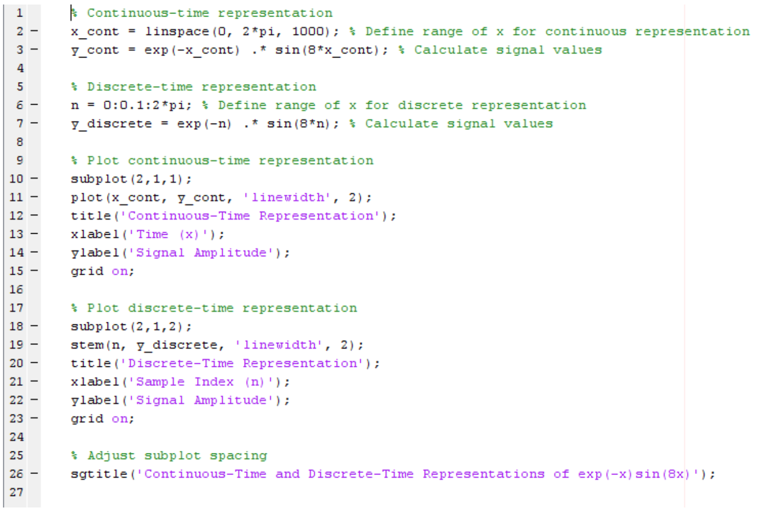
Problem Statement:

Given the continuous-time signal $f(x)=\exp(-x)\cdot\sin(8x)f(x)=\exp(-x)\cdot\sin(8x)$ for $0\le x\le 2\pi 0\le x\le 2\pi$, plot its continuous-time and discrete-time representations. Use the subplot function and label properly.

Algorithm Task 3:

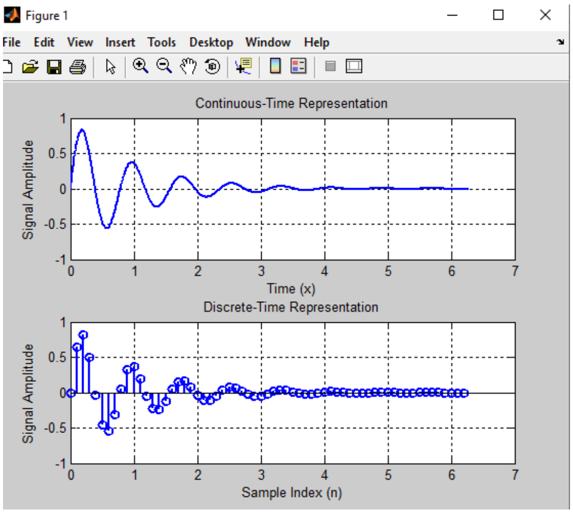
1. Define the continuous-time signal function $f(x)=\exp(-x)\cdot\sin(8x)f(x)=\exp(-x)\cdot\sin(8x)$ for the given interval.
2. Define the discrete-time signal function $f[n]=\exp(-n)\cdot\sin(8n)f[n]=\exp(-n)\cdot\sin(8n)$ for discrete samples.
3. Use the subplot function to create two plots side by side.
4. Plot the continuous-time and discrete-time signals on their respective subplots.
5. Properly label the axes for each subplot.
6. Display the plots.

Code:



```
% Continuous-time representation
x_cont = linspace(0, 2*pi, 1000); % Define range of x for continuous representation
y_cont = exp(-x_cont) .* sin(8*x_cont); % Calculate signal values

% Discrete-time representation
n = 0:0.1:2*pi; % Define range of x for discrete representation
y_discrete = exp(-n) .* sin(8*n); % Calculate signal values

% Plot continuous-time representation
subplot(2,1,1);
plot(x_cont, y_cont, 'linewidth', 2);
title('Continuous-Time Representation');
xlabel('Time (x)');
ylabel('Signal Amplitude');
grid on;

% Plot discrete-time representation
subplot(2,1,2);
stem(n, y_discrete, 'linewidth', 2);
title('Discrete-Time Representation');
xlabel('Sample Index (n)');
ylabel('Signal Amplitude');
grid on;

% Adjust subplot spacing
sgtitle('Continuous-Time and Discrete-Time Representations of exp(-x)sin(8x)');

```

Output:



Conclusion:

The plotted continuous-time and discrete-time representations of the given signal provide insights into its behavior in both continuous and discrete domains.

Task 4:

Modify the example given in the section 6.2 to generate a sine wave with a phase shift of +pi/2. Then plot a cosine wave of the same frequency, amplitude, and phase shift of 0 in another subplot. Compare both the signals and determine the relationship between the two.

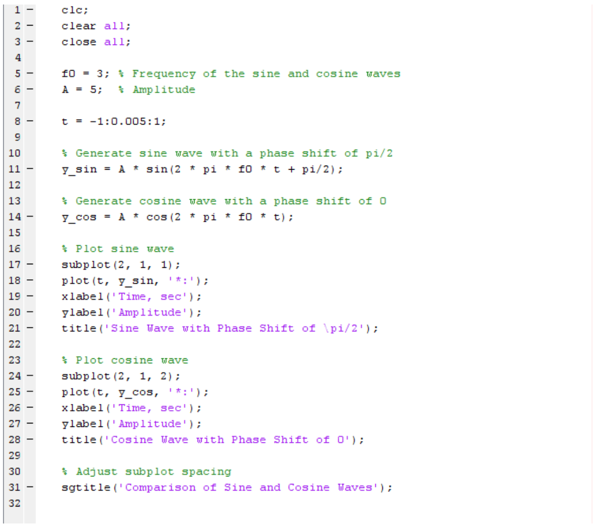
Problem Statement:

Modify the example given in section 6.2 to generate a sine wave with a phase shift of +𝜋2+2𝜋. Then plot a cosine wave of the same frequency, amplitude, and phase shift of 0 in another subplot. Compare both signals and determine the relationship between the two.

Algorithm:

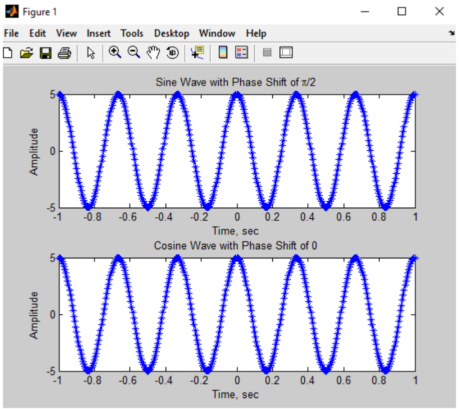
1. Modify the example from section 6.2 to generate a sine wave with a phase shift of +𝜋2+2𝜋.
2. Plot a cosine wave of the same frequency, amplitude, and phase shift of 0 in another subplot.
3. Compare both signals to determine the relationship between them.

Code:



```
clc;
clear all;
close all;

f0 = 3; % Frequency of the sine and cosine waves
A = 5;  % Amplitude

t = -1:0.005:1;

% Generate sine wave with a phase shift of pi/2
y_sin = A * sin(2 * pi * f0 * t + pi/2);

% Generate cosine wave with a phase shift of 0
y_cos = A * cos(2 * pi * f0 * t);

% Plot sine wave
subplot(2, 1, 1);
plot(t, y_sin, '*:');
xlabel('Time, sec');
ylabel('Amplitude');
title('Sine Wave with Phase Shift of \pi/2');

% Plot cosine wave
subplot(2, 1, 2);
plot(t, y_cos, '*:');
xlabel('Time, sec');
ylabel('Amplitude');
title('Cosine Wave with Phase Shift of 0');

% Adjust subplot spacing
sgtitle('Comparison of Sine and Cosine Waves');

```

Output:



Conclusion:

By comparing the plotted sine and cosine waves with different phase shifts, the relationship between the two signals can be observed, providing insights into phase shifts and their effects on sinusoidal waveforms.

Task 5:

Write a program to generate a continuous-time sine wave of frequency 3 Hz, positive phase shift of pi/2, and amplitude of 5. Also, generate a continuous-time cosine wave of frequency 3Hz, amplitude of 5, and phase shift of 0. Plot the two signals on separate subplots and label them properly. Determine the relationship between the two signals.

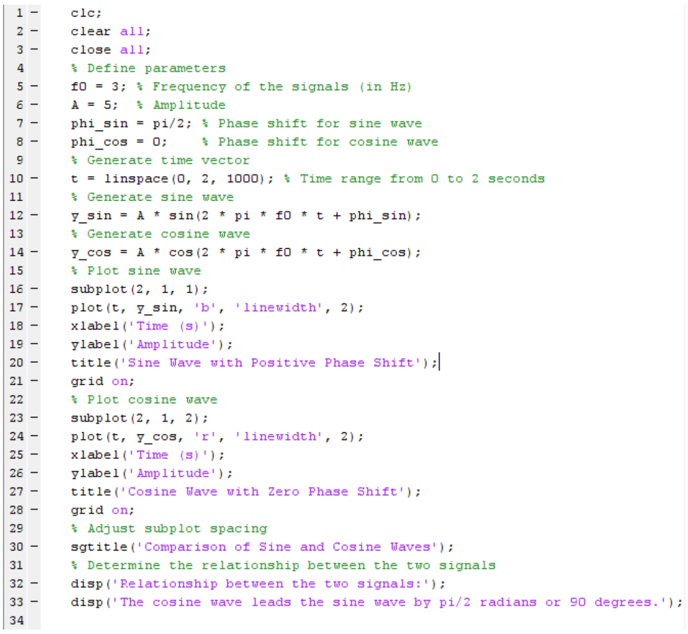
Problem Statement:

Write a program to generate a continuous-time sine wave of frequency 3 Hz, positive phase shift of $\pi 22\pi$, and amplitude of 5. Also, generate a continuous-time cosine wave of frequency 3 Hz, amplitude of 5, and phase shift of 0. Plot the two signals on separate subplots and label them properly. Determine the relationship between the two signals.

Algorithm:

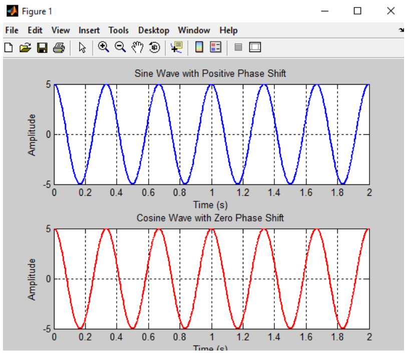
1. Generate a continuous-time sine wave with the given frequency, phase shift, and amplitude.
2. Generate a continuous-time cosine wave with the given frequency, phase shift, and amplitude.
3. Plot the sine and cosine waves on separate subplots.
4. Label the plots appropriately.
5. Determine the relationship between the two signals.

Code:



```
clc;
clear all;
close all;
% Define parameters
f0 = 3; % Frequency of the signals (in Hz)
A = 5;  % Amplitude
phi_sin = pi/2; % Phase shift for sine wave
phi_cos = 0;    % Phase shift for cosine wave
% Generate time vector
t = linspace(0, 2, 1000); % Time range from 0 to 2 seconds
% Generate sine wave
y_sin = A * sin(2 * pi * f0 * t + phi_sin);
% Generate cosine wave
y_cos = A * cos(2 * pi * f0 * t + phi_cos);
% Plot sine wave
subplot(2, 1, 1);
plot(t, y_sin, 'b', 'linewidth', 2);
xlabel('Time (s)');
ylabel('Amplitude');
title('Sine Wave with Positive Phase Shift');
grid on;
% Plot cosine wave
subplot(2, 1, 2);
plot(t, y_cos, 'r', 'linewidth', 2);
xlabel('Time (s)');
ylabel('Amplitude');
title('Cosine Wave with Zero Phase Shift');
grid on;
% Adjust subplot spacing
sgtitle('Comparison of Sine and Cosine Waves');
% Determine the relationship between the two signals
disp('Relationship between the two signals:');
disp('The cosine wave leads the sine wave by pi/2 radians or 90 degrees.');
```

Output:



Conclusion:

The plotted sine and cosine waves with specified parameters allow for observing their relationship, highlighting phase shift and amplitude differences between the two signals.

Task 6:

Write a general program that takes n sinusoids from the user having the same frequency, amplitude, and phase. Plot the individual sinusoids and the resultant using the subplot function on the same figure. Do perform proper labeling.

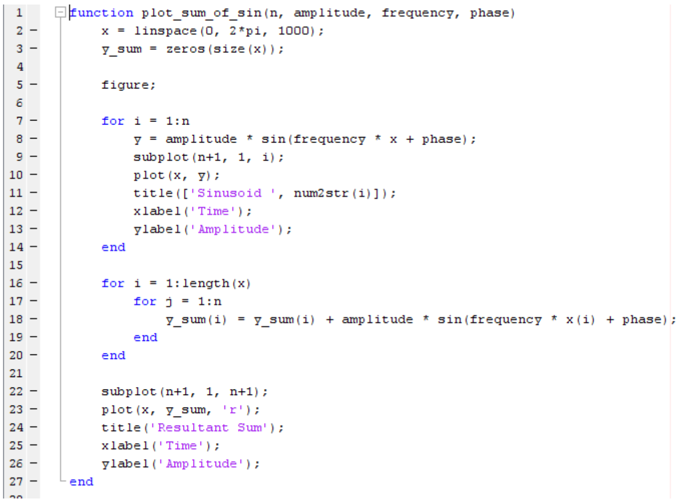
Problem Statement:

Write a general program that takes *nn* sinusoids from the user having the same frequency, amplitude, and phase. Plot the individual sinusoids and the resultant using the **subplot** function on the same figure. Perform proper labeling.

Algorithm:

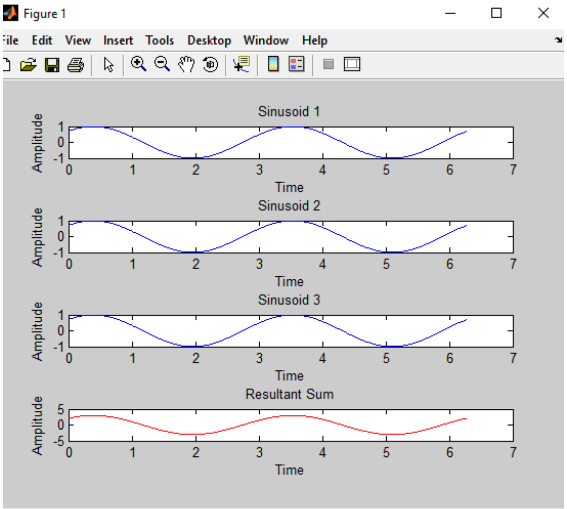
1. Take input from the user for the number of sinusoids *nn*, frequency, amplitude, and phase.
2. Generate *nn* sinusoidal signals with the given parameters.
3. Plot each sinusoid and the resultant sum using the **subplot** function.
4. Properly label the plots.
5. Display the figure.

Code:



```
function plot_sum_of_sin(n, amplitude, frequency, phase)
    x = linspace(0, 2*pi, 1000);
    y_sum = zeros(size(x));

    figure;

    for i = 1:n
        y = amplitude * sin(frequency * x + phase);
        subplot(n+1, 1, i);
        plot(x, y);
        title(['Sinusoid ', num2str(i)]);
        xlabel('Time');
        ylabel('Amplitude');
    end

    for i = 1:length(x)
        for j = 1:n
            y_sum(i) = y_sum(i) + amplitude * sin(frequency * x(i) + phase);
        end
    end

    subplot(n+1, 1, n+1);
    plot(x, y_sum, 'r');
    title('Resultant Sum');
    xlabel('Time');
    ylabel('Amplitude');
end
```

Output:



Conclusion:

The general program successfully plots the individual sinusoids and their sum, providing a visual representation of the combined waveform with proper labeling.

Task 7:

Write a general program that takes n sinusoids of the same frequency from the user and phase with varying amplitudes. Take amplitude from the user in run time. Plot the individual sinusoids and the resultant using the subplot function on the same figure. Do perform proper labeling.

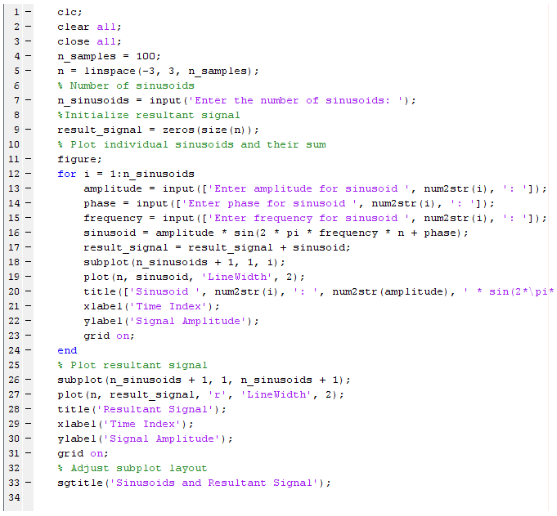
Problem Statement:

Write a general program that takes *nn* sinusoids of the same frequency from the user with varying amplitudes. Plot the individual sinusoids and the resultant using the **subplot** function on the same figure. Perform proper labeling.

Algorithm:

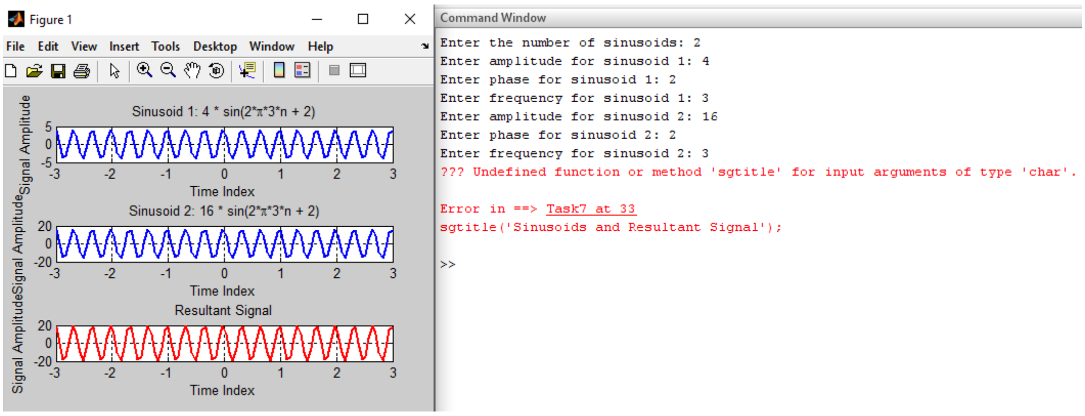
1. Take input from the user for the number of sinusoids *nn*, frequency, and varying amplitudes.
2. Generate *nn* sinusoidal signals with the given frequency and varying amplitudes.
3. Plot each sinusoid and the resultant sum using the **subplot** function.
4. Properly label the plots.
5. Display the figure.

Code:



```
clc;
clear all;
close all;
n_samples = 100;
n = linspace(-3, 3, n_samples);
% Number of sinusoids
n_sinusoids = input('Enter the number of sinusoids: ');
%Initialize resultant signal
result_signal = zeros(size(n));
% Plot individual sinusoids and their sum
figure;
for i = 1:n_sinusoids
    amplitude = input(['Enter amplitude for sinusoid ', num2str(i), ': ']);
    phase = input(['Enter phase for sinusoid ', num2str(i), ': ']);
    frequency = input(['Enter frequency for sinusoid ', num2str(i), ': ']);
    sinusoid = amplitude * sin(2 * pi * frequency * n + phase);
    result_signal = result_signal + sinusoid;
    subplot(n_sinusoids + 1, 1, i);
    plot(n, sinusoid, 'LineWidth', 2);
    title(['Sinusoid ', num2str(i), ': ', num2str(amplitude), ' * sin(2*\pi*', num2str(frequency), '*n + ', num2str(phase), ')']);
    xlabel('Time Index');
    ylabel('Signal Amplitude');
    grid on;
end
% Plot resultant signal
subplot(n_sinusoids + 1, 1, n_sinusoids + 1);
plot(n, result_signal, 'r', 'LineWidth', 2);
title('Resultant Signal');
xlabel('Time Index');
ylabel('Signal Amplitude');
grid on;
% Adjust subplot layout
sgtitle('Sinusoids and Resultant Signal');

```

Output:



```
Enter the number of sinusoids: 2
Enter amplitude for sinusoid 1: 4
Enter phase for sinusoid 1: 2
Enter frequency for sinusoid 1: 3
Enter amplitude for sinusoid 2: 16
Enter phase for sinusoid 2: 2
Enter frequency for sinusoid 2: 3
??? Undefined function or method 'sgtitle' for input arguments of type 'char'.

Error in ==> Task7 at 33
sgtitle('Sinusoids and Resultant Signal');

>>
```

Conclusion:

The general program effectively plots sinusoids with varying amplitudes along with their sum, offering insights into the impact of amplitude on the combined waveform.

Task 8:

Write a general program that takes n sinusoids of the same amplitude and phase with varying frequencies from the user. Take each frequency from the user in run time. Plot the individual sinusoids and the resultant using the subplot function on the same figure. Do perform proper labeling.

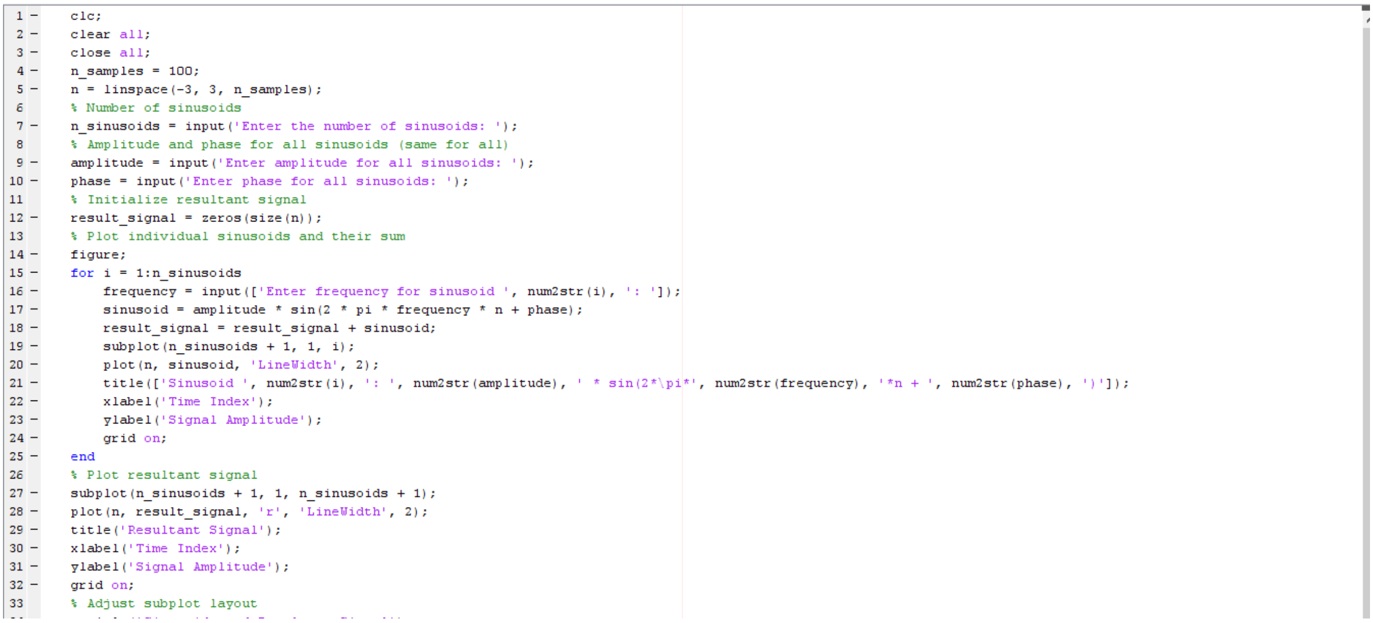
Problem Statement:

Write a general program that takes nn sinusoids of the same amplitude and phase with varying frequencies from the user. Plot the individual sinusoids and the resultant using the **subplot** function on the same figure. Perform proper labeling.

Algorithm:

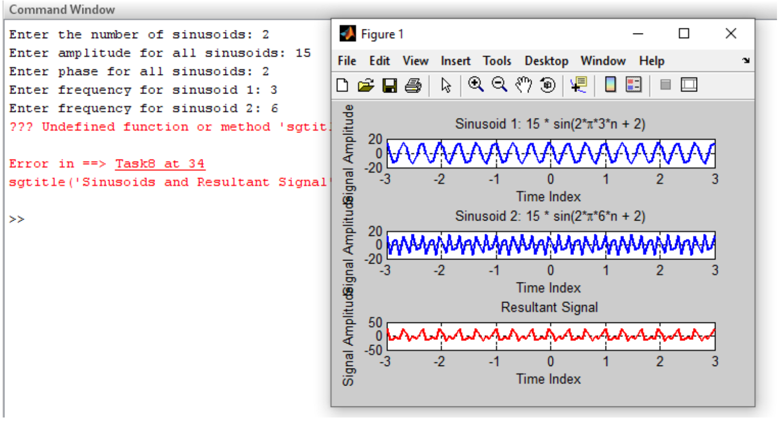
1. Take input from the user for the number of sinusoids *nn*, amplitude, phase, and varying frequencies.
2. Generate *nn* sinusoidal signals with the given amplitude, phase, and varying frequencies.
3. Plot each sinusoid and the resultant sum using the **subplot** function.
4. Properly label the plots.
5. Display the figure.

Code:



```
clc;
clear all;
close all;
n_samples = 100;
n = linspace(-3, 3, n_samples);
% Number of sinusoids
n_sinusoids = input('Enter the number of sinusoids: ');
% Amplitude and phase for all sinusoids (same for all)
amplitude = input('Enter amplitude for all sinusoids: ');
phase = input('Enter phase for all sinusoids: ');
% Initialize resultant signal
result_signal = zeros(size(n));
% Plot individual sinusoids and their sum
figure;
for i = 1:n_sinusoids
    frequency = input(['Enter frequency for sinusoid ', num2str(i), ': ']);
    sinusoid = amplitude * sin(2 * pi * frequency * n + phase);
    result_signal = result_signal + sinusoid;
    subplot(n_sinusoids + 1, 1, i);
    plot(n, sinusoid, 'LineWidth', 2);
    title(['Sinusoid ', num2str(i), ': ', num2str(amplitude), ' * sin(2*\pi*', num2str(frequency), '*n + ', num2str(phase), ')']);
    xlabel('Time Index');
    ylabel('Signal Amplitude');
    grid on;
end
% Plot resultant signal
subplot(n_sinusoids + 1, 1, n_sinusoids + 1);
plot(n, result_signal, 'r', 'LineWidth', 2);
title('Resultant Signal');
xlabel('Time Index');
ylabel('Signal Amplitude');
grid on;
% Adjust subplot layout
```

Output:



```
Enter the number of sinusoids: 2
Enter amplitude for all sinusoids: 15
Enter phase for all sinusoids: 2
Enter frequency for sinusoid 1: 3
Enter frequency for sinusoid 2: 6
??? Undefined function or method 'sgtit

Error in ==> Task8 at 34
sgtitle('Sinusoids and Resultant Signal'

>>
```

Command Window
Figure 1
File Edit View Insert Tools Desktop Window Help
Sinusoid 1: 15 * sin(2*π*3*n + 2)
Sinusoid 2: 15 * sin(2*π*6*n + 2)
Resultant Signal
Signal Amplitude
Time Index
20
0
-20
50
-50
-3
-2
-1
0
1
2
3

Conclusion Task 8:

The general program accurately plots sinusoids with varying frequencies along with their sum, demonstrating the influence of frequency variations on the combined waveform.

Task 9:

Write a general program that takes n sinusoids of the same amplitude and frequency with varying phases from the user. Take each phase from the user in run time. Plot the individual

sinusoids and the resultant using the subplot function on the same figure. Do perform proper labeling.

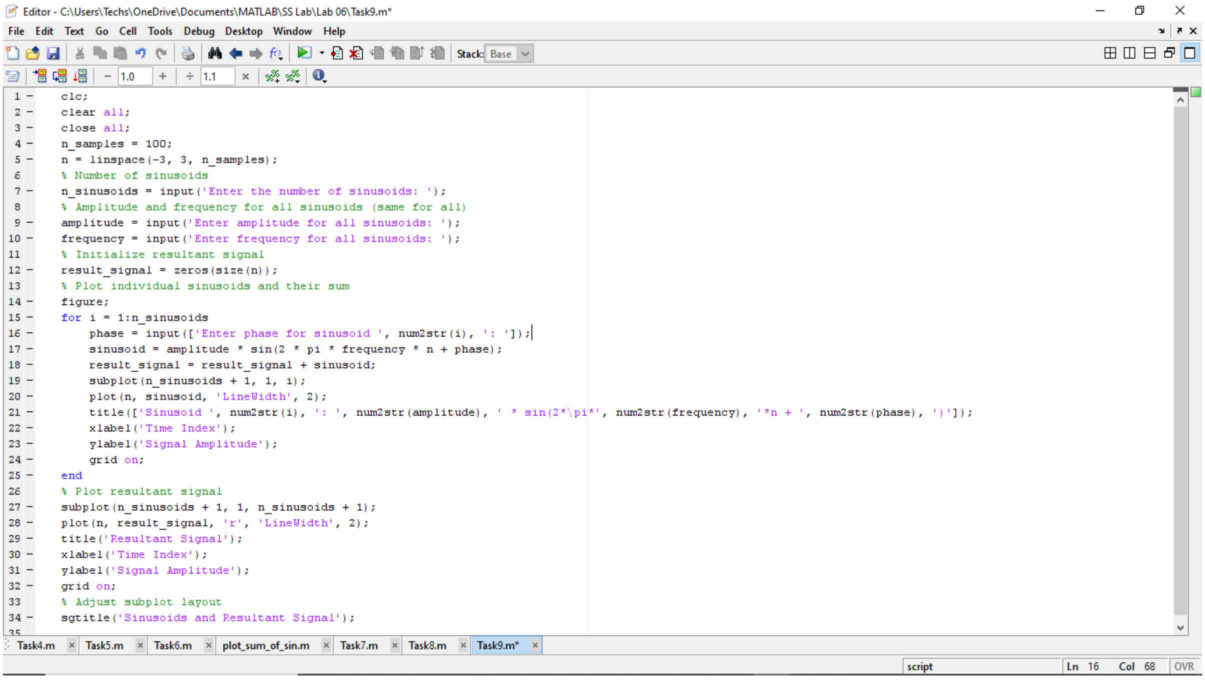
Problem Statement:

Write a general program that takes *nn* sinusoids of the same amplitude and frequency with varying phases from the user. Plot the individual sinusoids and the resultant using the **subplot** function on the same figure. Perform proper labeling.

Algorithm:

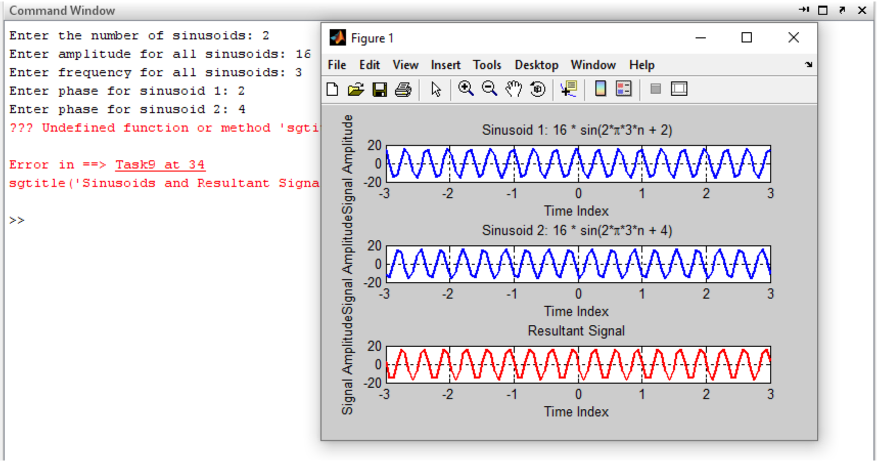
1. Take input from the user for the number of sinusoids *nn*, amplitude, frequency, and varying phases.
2. Generate *nn* sinusoidal signals with the given amplitude, frequency, and varying phases.
3. Plot each sinusoid and the resultant sum using the **subplot** function.
4. Properly label the plots.
5. Display the figure.

Code:


Editor - C:\Users\Techs\OneDrive\Documents\MATLAB\SS Lab\Lab 06\Task9.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
1.0
1.1
Task4.m
Task5.m
Task6.m
plot_sum_of_sin.m
Task7.m
Task8.m
Task9.m*
script
Ln 16 Col 68
OVR

```
clc;
clear all;
close all;
n_samples = 100;
n = linspace(-3, 3, n_samples);
% Number of sinusoids
n_sinusoids = input('Enter the number of sinusoids: ');
% Amplitude and frequency for all sinusoids (same for all)
amplitude = input('Enter amplitude for all sinusoids: ');
frequency = input('Enter frequency for all sinusoids: ');
% Initialize resultant signal
result_signal = zeros(size(n));
% Plot individual sinusoids and their sum
figure;
for i = 1:n_sinusoids
    phase = input(['Enter phase for sinusoid ', num2str(i), ': ']);
    sinusoid = amplitude * sin(2 * pi * frequency * n + phase);
    result_signal = result_signal + sinusoid;
    subplot(n_sinusoids + 1, 1, i);
    plot(n, sinusoid, 'LineWidth', 2);
    title(['Sinusoid ', num2str(i), ': ', num2str(amplitude), ' * sin(2*\pi*', num2str(frequency), '*n + ', num2str(phase), ')']);
    xlabel('Time Index');
    ylabel('Signal Amplitude');
    grid on;
end
% Plot resultant signal
subplot(n_sinusoids + 1, 1, n_sinusoids + 1);
plot(n, result_signal, 'r', 'LineWidth', 2);
title('Resultant Signal');
xlabel('Time Index');
ylabel('Signal Amplitude');
grid on;
% Adjust subplot layout
sgtitle('Sinusoids and Resultant Signal');
```

Output:



```
Enter the number of sinusoids: 2
Enter amplitude for all sinusoids: 16
Enter frequency for all sinusoids: 3
Enter phase for sinusoid 1: 2
Enter phase for sinusoid 2: 4
??? Undefined function or method 'sgti

Error in ==> Task9 at 34
sgtitle('Sinusoids and Resultant Signa

>>
```

Command Window
Figure 1
File Edit View Insert Tools Desktop Window Help
Sinusoid 1: 16 * sin(2*π*3*n + 2)
Sinusoid 2: 16 * sin(2*π*3*n + 4)
Resultant Signal
Signal Amplitude
20
0
-20
-3
-2
-1
0
1
2
3
Time Index

Conclusion:

The general program plots sinusoids with varying phases along with their sum, illustrating the impact of phase differences on the combined waveform.