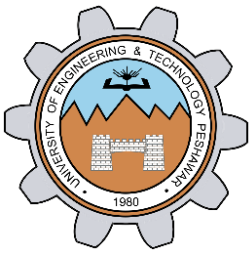
INTRODUCTION TO SINUSOIDAL SIGNALS IN MATLAB

LAB # 06



Spring 2023

CSE301L Signals & Systems Lab

Submitted by: **Ali Asghar**

Registration No. : **21PWCSE2059**

Class Section: **C**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Submitted to:

Engr. Sumayyea Salahuddin

Date:

May 4, 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Lab Objective(s):

Objectives of this Lab are;

- Generating Sinusoids
- Addition of Sinusoids with Variation in Parameters and their Plots
- Linear Phase Shift Concept When Dealing With Sum of Sinusoids

Task # 01:

Generate the 1x10 row vector v whose i-th component is cos (iπ/4).

Problem Analysis:

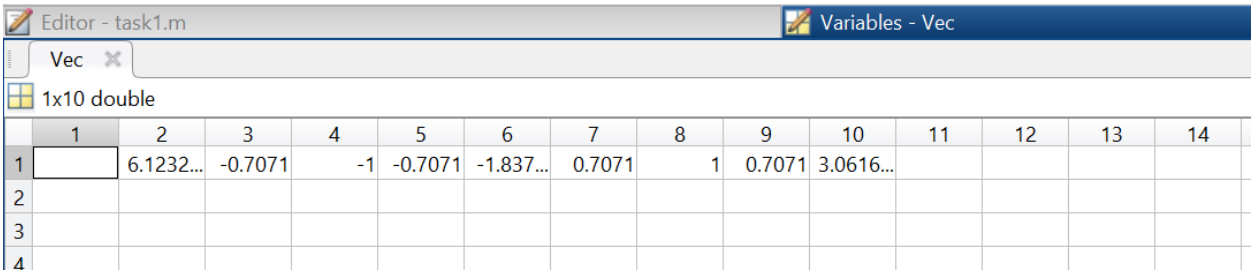
In the production and analysis of waves, vectors are important.

Algorithm:

- Take in vector parameters as given in problem statement.
- Run a for loop from 1 to 10 and save the ith component in ith column of Vec

Code:

Output / Graphs / Plots / Results:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1		6.1232...	-0.7071	-1	-0.7071	-1.837...	0.7071	1	0.7071	3.0616...				
2														
3														
4														

Discussion and Conclusion:

In MATLAB, we can generate vectors whose ith component is sinusoid.

Task # 02:

Write matlab code that draw graphs of sin (nπx) on the interval $-1 \leq x \leq 1$ for n = 1, 2, 3, …, 8.
(Hint: Use for loop)

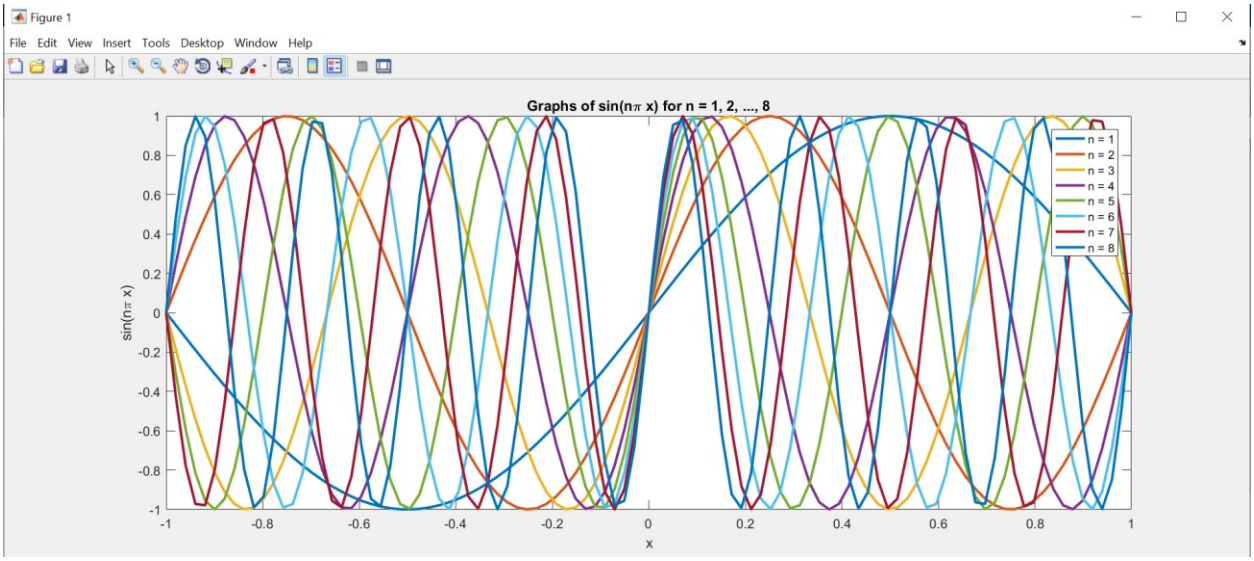
Problem Analysis:

For its analysis, wave simulation is needed. MATLAB allows us to simulate and create waves.

Algorithm:

- Create a vector of x-values from -1 to 1 using the `linspace()` function.
- Create a vector of n-values from 1 to 8.
- Create a new figure using the `figure()` function.
- Loop over the n-values using a `for` loop.
- Within the `for` loop, calculate the values of `y` for the current value of `n` using the `sin()` function and the x-values.
- Plot the `y` values against the `x` values using the `plot()` function with a line width of 2, and use `hold on` to add each line to the same plot.
- Label the x-axis and y-axis using the `xlabel()` and `ylabel()` functions.
- Set the title of the figure using the `title()` function.
- Add a legend to the figure using the `legend()` function with labels for each line.

Output / Graphs / Plots / Results:



Discussion and Conclusion:

For its analysis, wave simulation is needed. MATLAB allows us to simulate and create waves.

Task # 03:

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

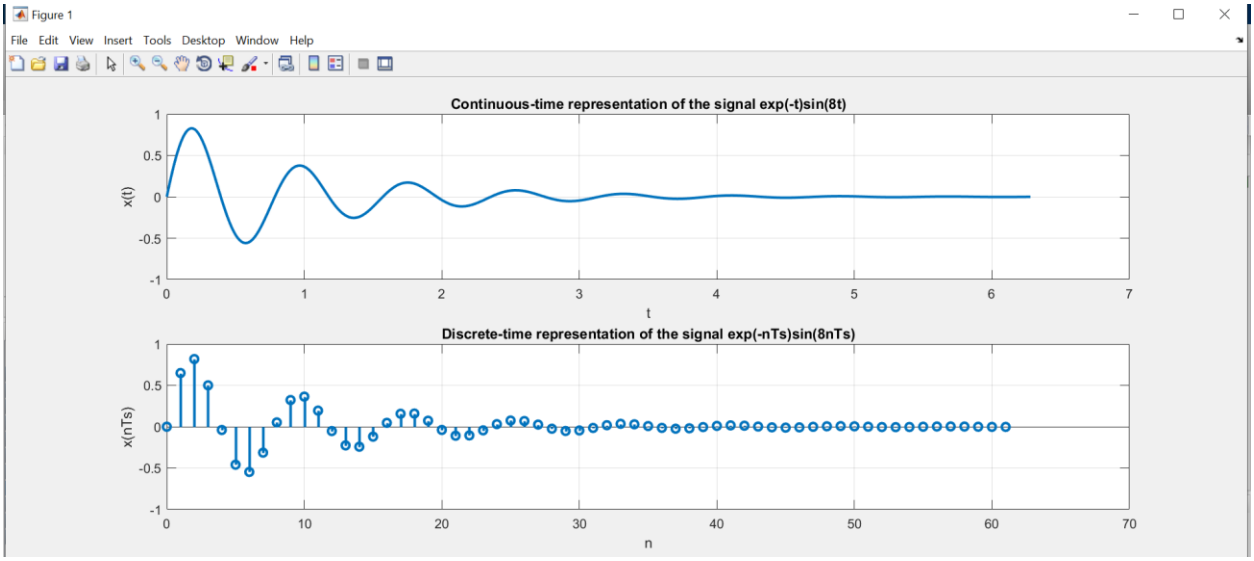
Problem Analysis:

The key concepts of signals and systems are continuous time and discrete time signals.

Algorithm:

- Define a vector of time values t using the 0:0.01:2*pi syntax.
- Define a continuous-time signal x as the product of exp(-t) and sin(8*t).
- Define the sampling frequency fs and sampling interval Ts.
- Create a vector of discrete-time values n using the 0:fs*2*pi-1 syntax.
- Sample the continuous-time signal x at the discrete-time values n using exp(-n*Ts).*sin(8*n*Ts).
- Plot the continuous-time signal x against t on the first subplot using the plot() function with a line width of 2 and labeled axes.
- Plot the discrete-time signal x_sampled against n on the second subplot using the stem() function with a line width of 2 and labeled axes.

Output / Graphs / Plots / Results:



Discussion and Conclusion:

We can plot and visualize continuous and discrete time signals in MATLAB.

Task # 04:

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

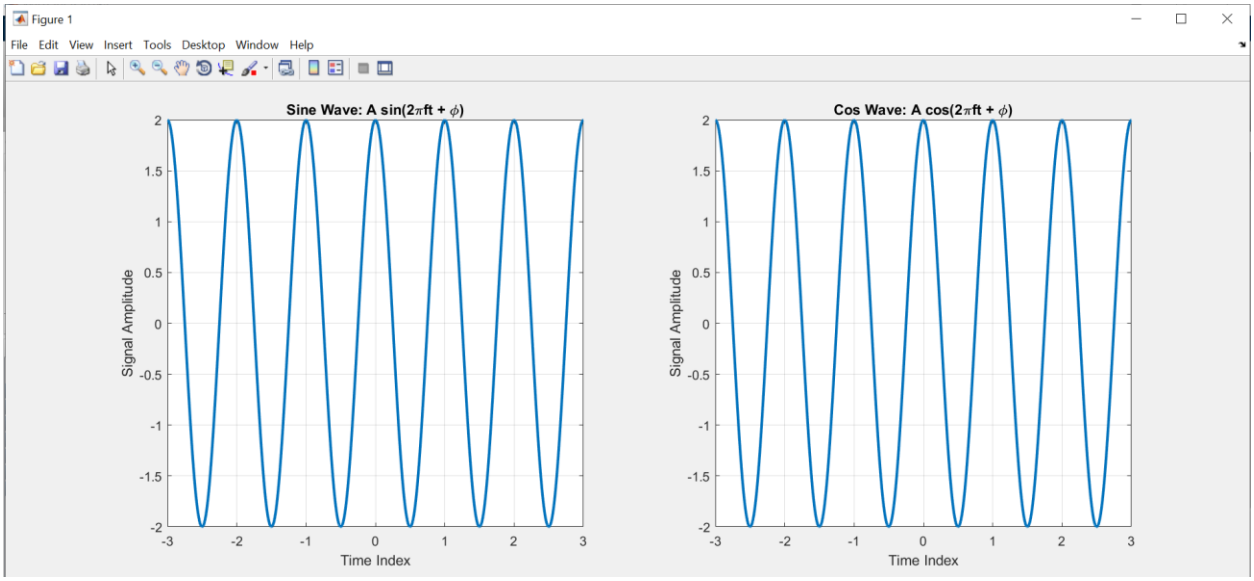
Problem Analysis:

For different tasks, we may need sine and cosine waves with ranging/identical characteristics. That can be achieved in MATLAB.

Algorithm:

- Set the sampling frequency fs to 1000 Hz.
- Define the time vector t using -3:1/fs:3 syntax, which creates a time vector from -3 to 3 seconds with a step size of 1/fs.
- Set the amplitude A to 2, phase phase to pi/2, and frequency f to 1.
- Define x1 as a sine wave with amplitude A, frequency f, phase phase, and sampled at times t.
- Define x2 as a cosine wave with amplitude A, frequency f, phase 0, and sampled at times t.
- Create a new figure with two subplots arranged horizontally.

Output / Graphs / Plots / Results:



Discussion and Conclusion:

In MATLAB, sinousoidal waves can be generated continuously.

Task # 05:

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 3 Hz, amplitude of 5, and phase shift of 0. Plot the two signals on separate subplots and properly label them. Determine the relationship between the two signals

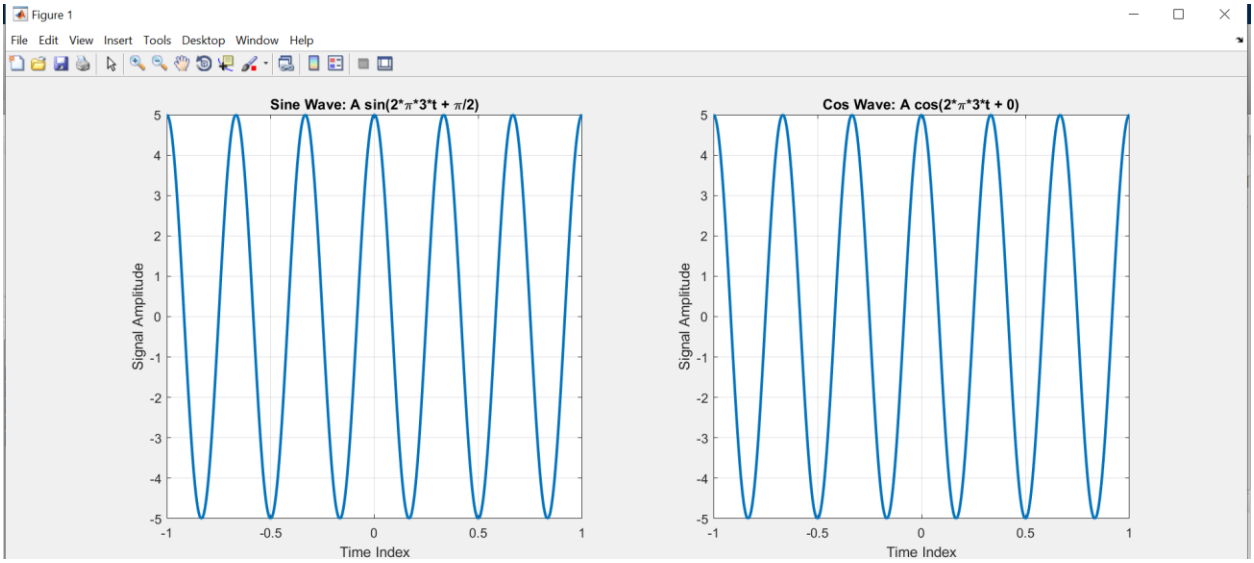
Problem Analysis:

We may need different characteristic sine waves for our different operations. We can do that in MATLAB.

Algorithm:

- Define the sampling frequency fs as 1000 Hz. and time vector t using the linspace function with a range of -1 to 1, and a step size of 1/fs and set the amplitude of the signals to 5.
- Define the first signal x1 as a sinusoidal wave with frequency 3 Hz and a phase shift of pi/2 radians.
- Define the second signal x2 as a cosine wave with frequency 3 Hz and no phase shift.
- The signals are now stored in variables x1 and x2.
- Create a new figure with two subplots arranged horizontally.

Output / Graphs / Plots / Results:



Discussion and Conclusion:

Generating continuous sinusoidal waves in MATLAB is possible.

Task # 06:

Write a general program that takes 'n' sinusoids from user of same frequency, amplitude, and phase. Plot the individual sinusoids & the resultant using subplot function on same figure. Do perform proper labeling. Note: Take the amplitude, frequency, and phase given in example of case 1. Run the code for different values of n and state the result on paper.

Problem Analysis:

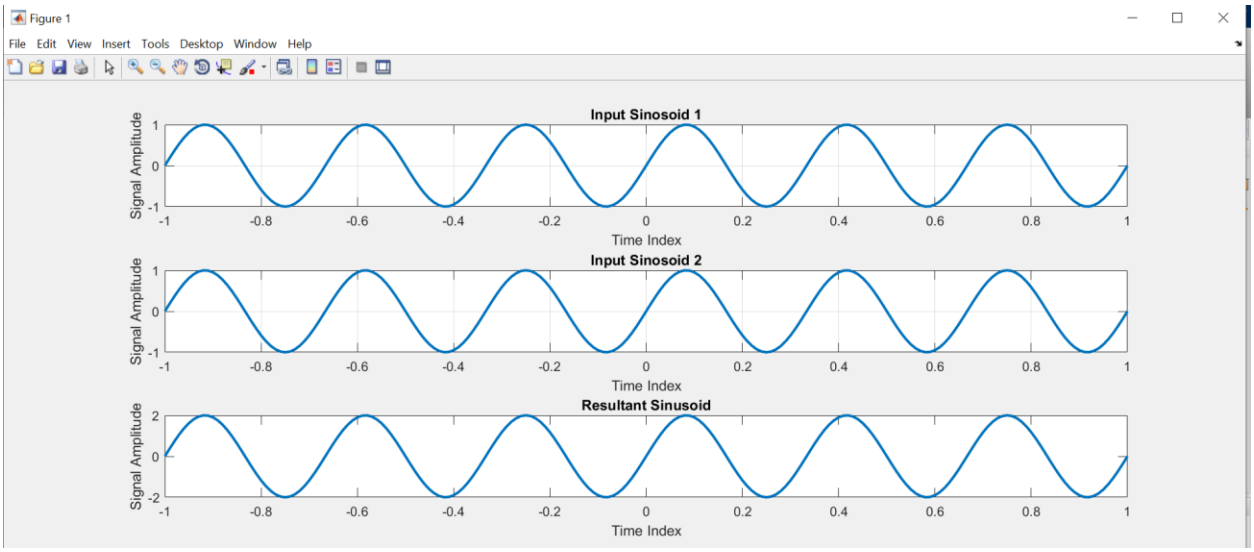
We frequently have to compare different sinusoids with different frequencies. That is possible in MATLAB.

Algorithm:

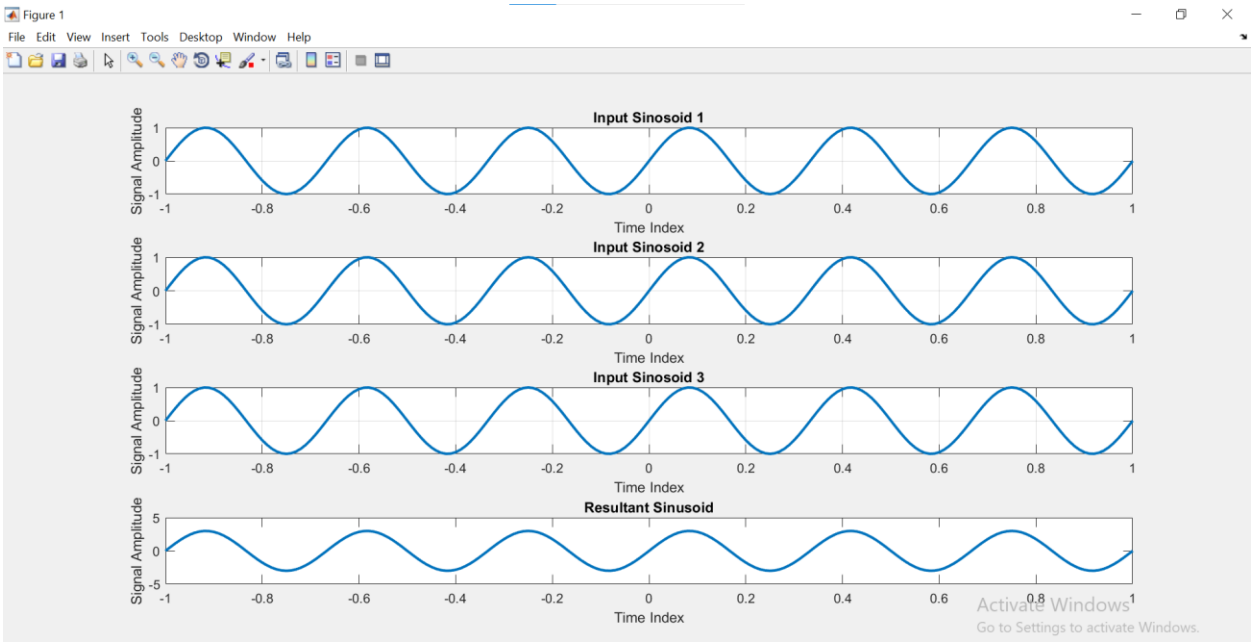
- Take in no, of sinusoidals.
- Input their varying parameters.
- Create these sinousoidals and subplot them.

Output / Graphs / Plots / Results:

For n = 2



For n = 3



Discussion and Conclusion:

Task # 07:

Write a general program that takes ‘n’ sinusoids from user of same frequency and phase with varying amplitudes. Take amplitude from user on run time. Plot the individual sinusoids & the resultant using subplot function on same figure. Do perform proper labeling. Note: Take the amplitude and frequency given in example of case 2. Run the code for different values of n and state the result on paper.

Problem Analysis:

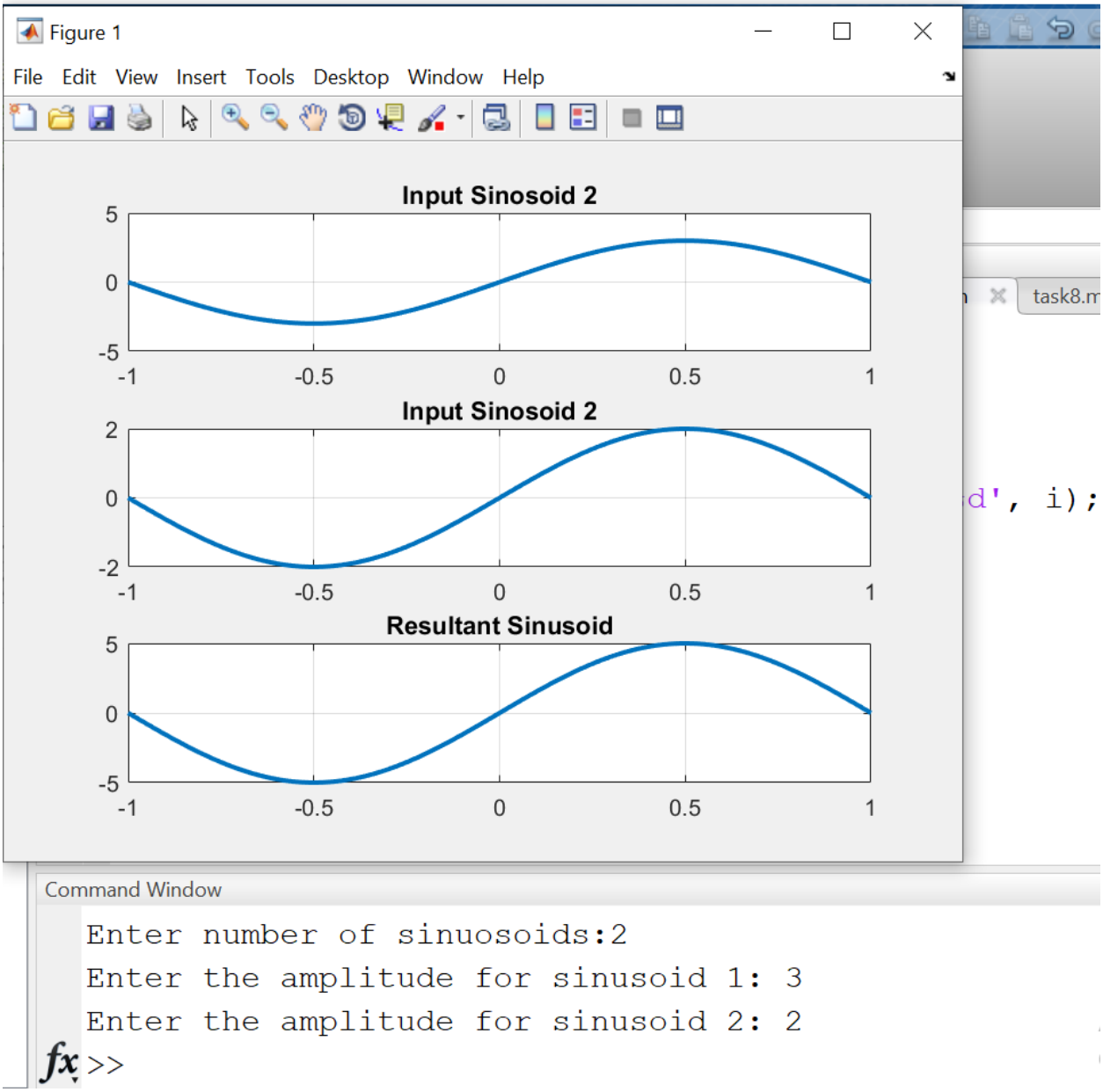
We frequently have to compare different sinusoids with different frequencies. That can be accomplished in MATLAB.

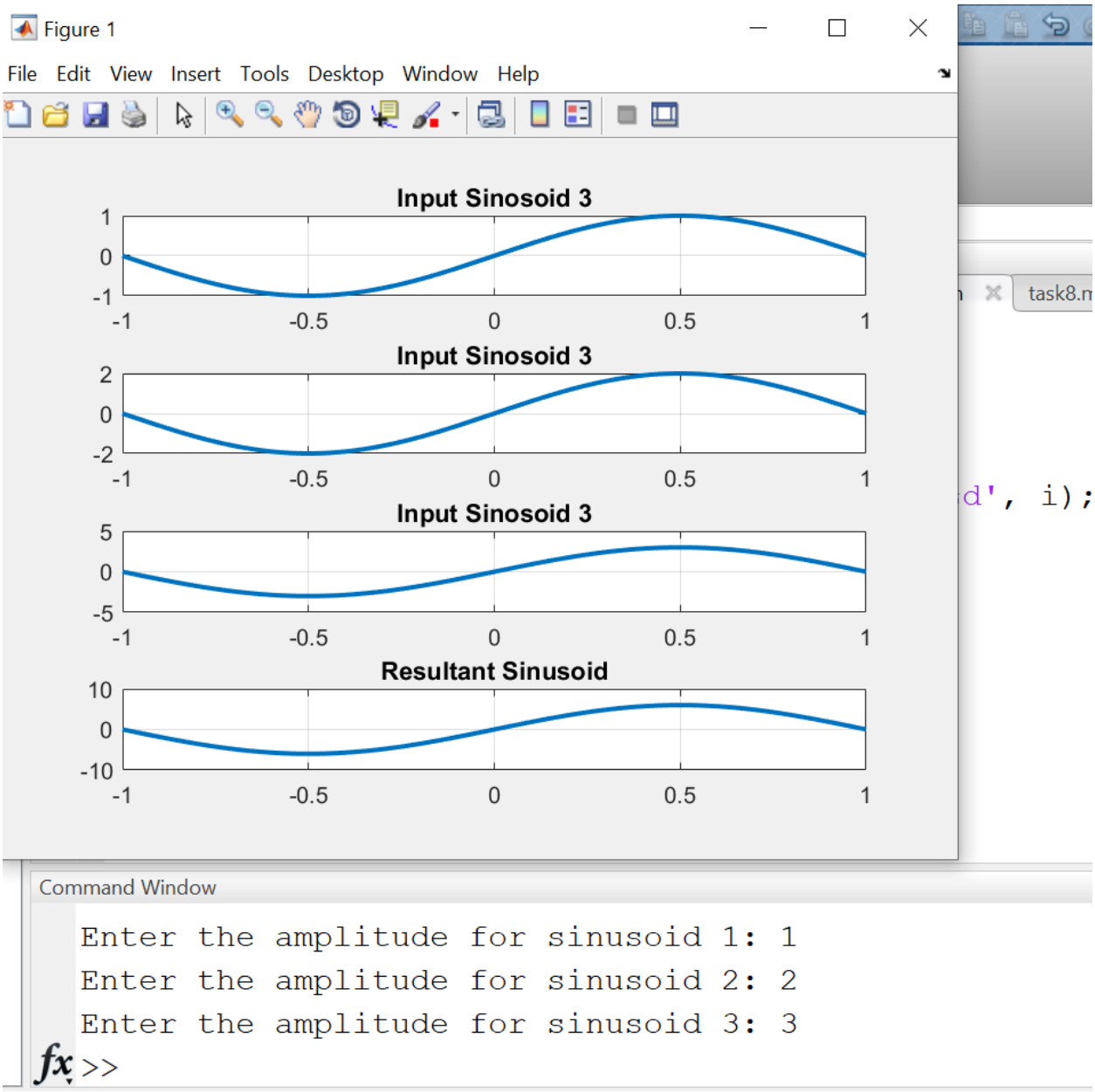
Algorithm:

- Take in no, of sinousoidals.
- Input their varying parameters.
- Create these sinousoidals and subplot them.

Code:

Output / Graphs / Plots / Results:





Discussion and Conclusion:

As a result, we can compare sine waves in MATLAB.

Task # 08:

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

amplitude and phase given in example of case 3. Run the code for different values of n and state the result on paper.

Problem Analysis:

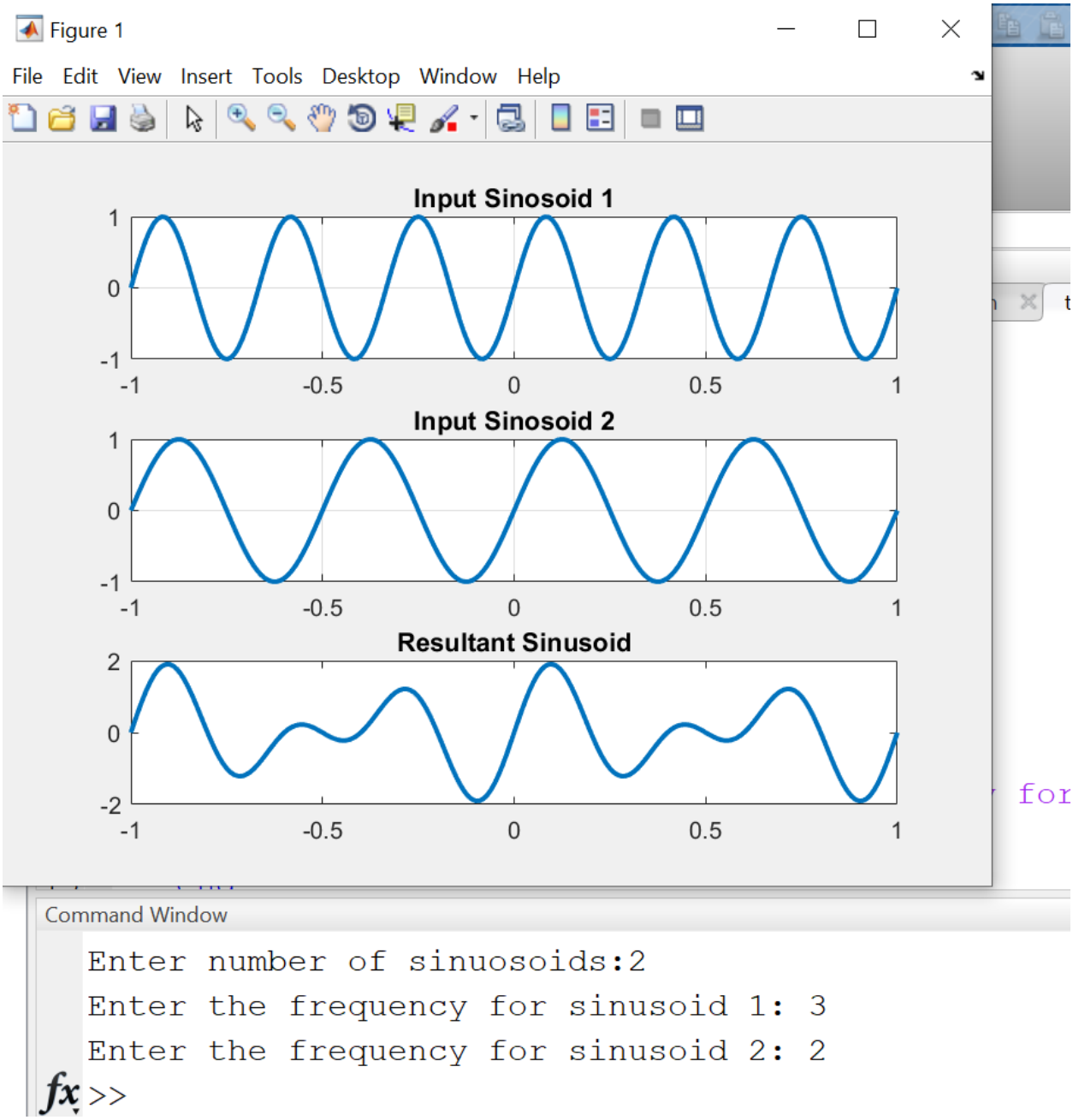
We tend to compare various sinusoids with different frequencies. That can be achieved in MATLAB.

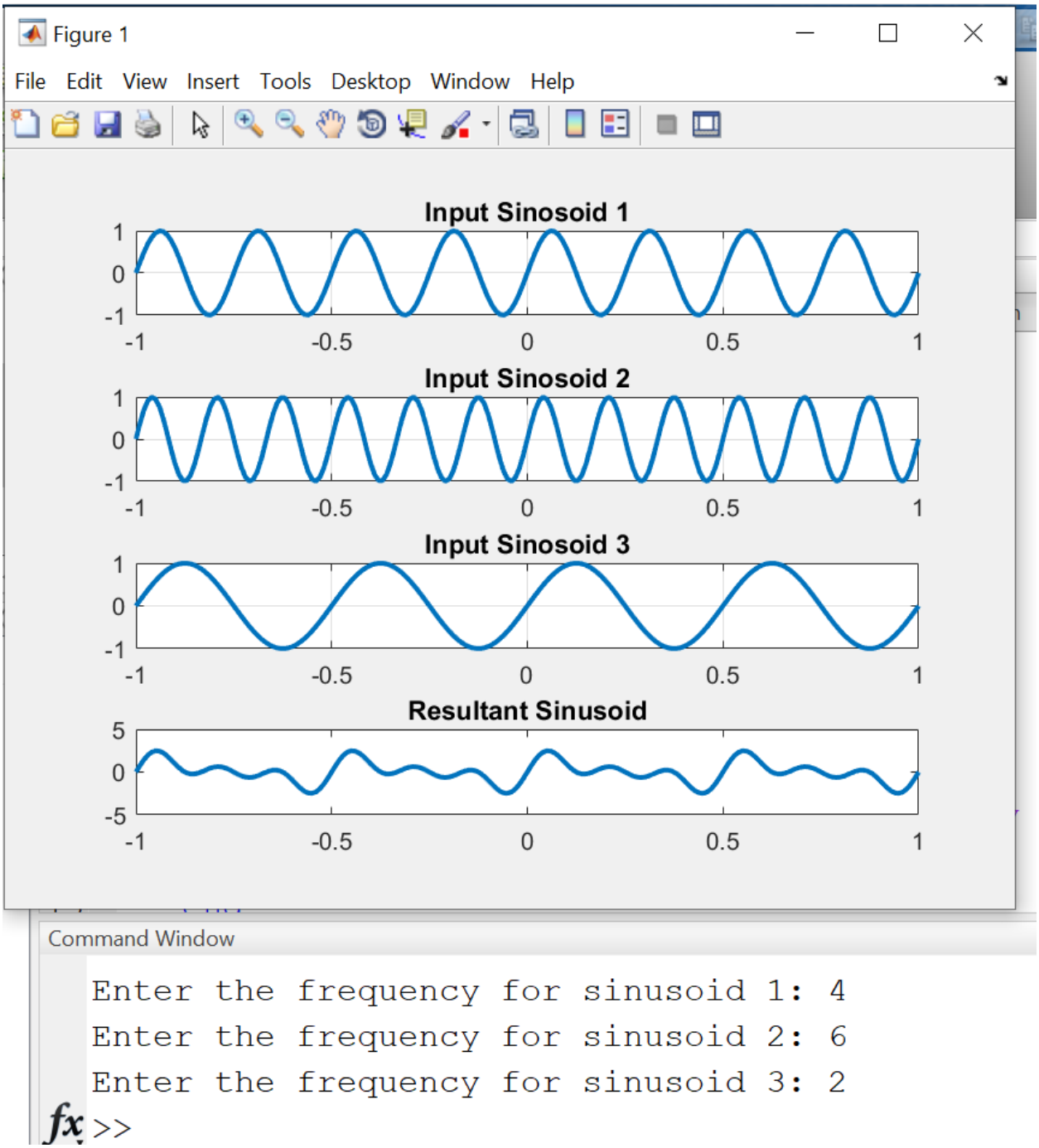
Algorithm:

- Take in no, of sinousoidals.
- Input their varying parameters.
- Create these sinousoidals and subplot them.

Code:

Output / Graphs / Plots / Results:





Discussion and Conclusion:

We frequently contrast distinct sinusoids with varying frequency. In MATLAB, that is achievable.

Task # 09:

Write a general program that takes ‘n’ sinusoids from user of same amplitude and frequency with varying phases. Take each phase from user on run time. Plot the individual sinusoids & the resultant using subplot function on same figure. Do perform proper labeling. Note: Take the amplitude and frequency given in example of case 4. Run the code for different values of n and state the result on paper.

Problem Analysis:

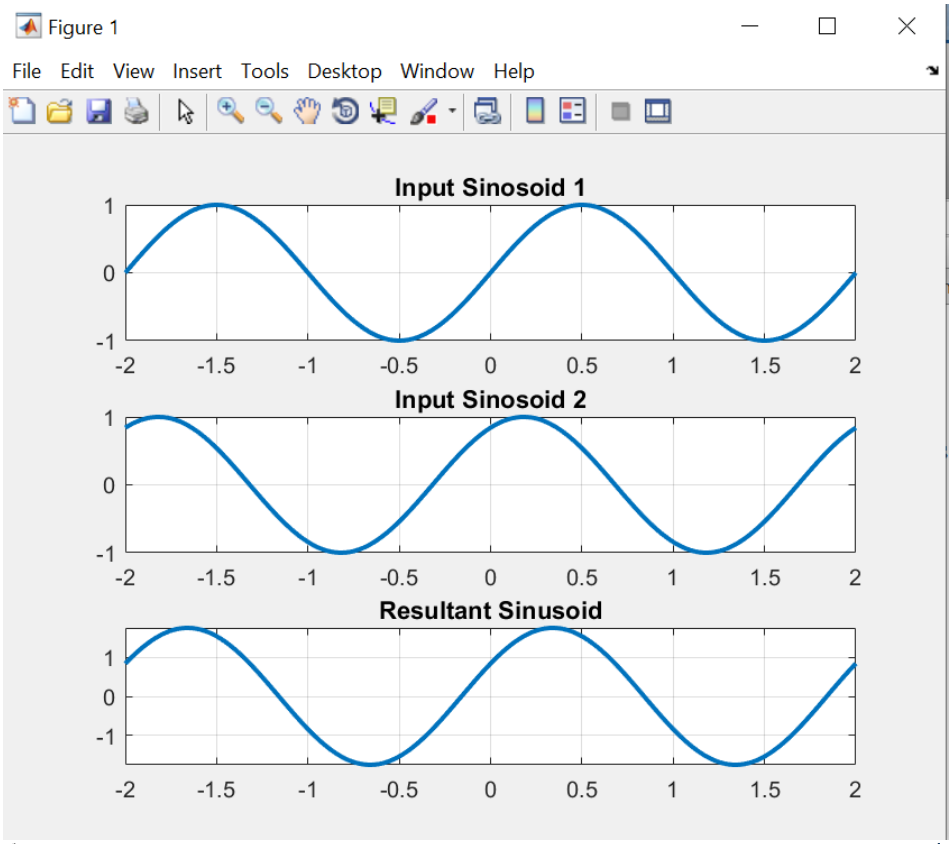
We are frequently able to compare different sinusoids. That can be done in MATLAB.

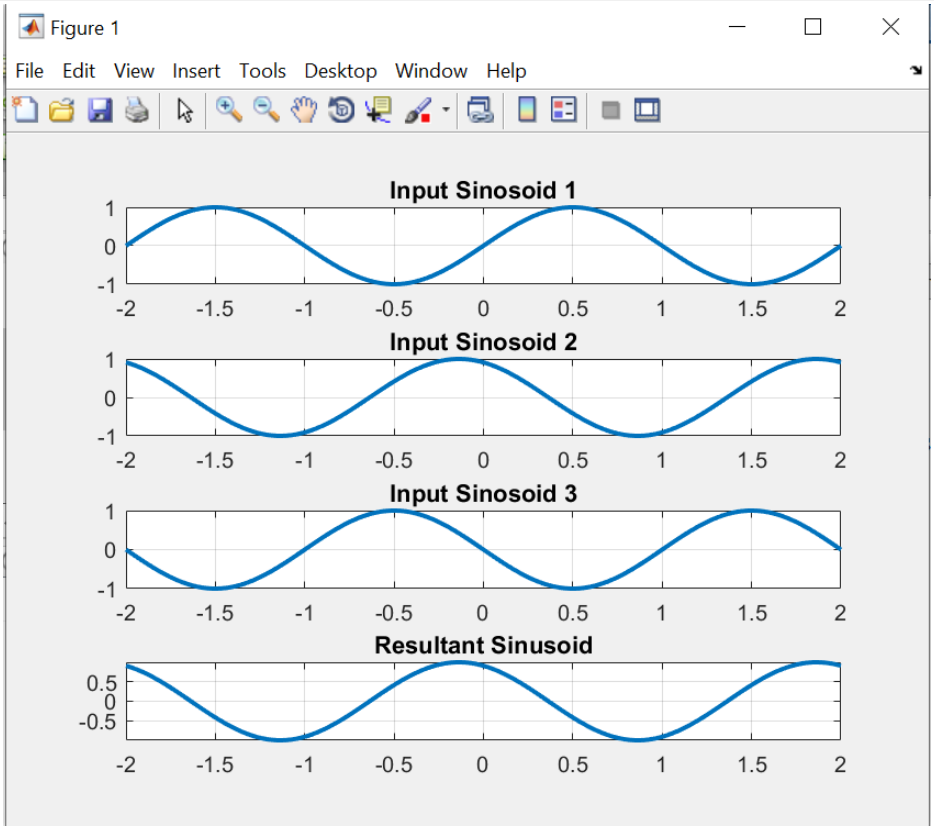
Algorithm:

- Input no of sinusoidal.
- Create those sinusoidal with user input various phase shifts.
- Subplot all signals.

Code:

Output / Graphs / Plots / Results:





Discussion and Conclusion:

In MATLAB, we can quickly contrast various sinusoidal.