VIBRATION SIGNAL ANALYSIS USING MATLAB GUI

A Project Report

Submitted In Partial Fulfilment For The

Award Of The Degree Of

Bachelor of Technology

IN

Electronics and Communication Engineering

Submitted by

Saikat Banerjee

Md Shahid

Subham Shaw

Under the Guidence of...

Prof. Nilanjan Chattaraj

NATIONAL INSTIUTE OF TECHNOLOGY DURGAPUR

DEPT. OF ELECTRONICS AND COMMUNICATION ENGINEERING

DECLARATION

We hearby declare that this submission is own work to the best of our knowledge and belief, it contains no material previously or written by any other person nor material whose substantial extent has been accepted for the award of any degree or diploma of the university or other institute of higher learning,except where due acknowledgement and reference has been made in the text.

NAME

SAIKAT BANERJEE

MD SHAHID

SUBHAM SHAW

ACKNOWLEDGMENT

First of all, we would like to acknowledge Professor Nilanjan Chattaraj's support that enabled ours and gave this opportunity to innovate and work on the project. we would also like to thank National Instiute of Technology, Durgapur for providing ours with the suitable environment to work on the project. We would like to acknowledge the researchers for providing vast library of information, which helped ours in every step of the way, pursuing this project.

TABLE OF CONTENTS

ABSTRACT

A **damped sine wave** is a sinusoidal function whose amplitude approaches zero as time increases.

Damped sine waves are commonly seen in science and engineering wherever a harmonic oscillator is losing energy faster than it is being supplied.

Sine waves describe many oscillating phenomena. When the wave is damped, each successive peak decreases as time goes on.

A true sine wave starting at time = 0 begins at the origin (amplitude = 0). A cosine wave begins at its maximum value due to its phase difference from the sine wave. In practice a given waveform may be of intermediate phase, having both sine and cosine components.The term "damped sine wave" describes all such damped waveforms, whatever their initial phase value.

The most common form of damping and that usually assumed are exponential damping ,in which the outer envelope of the successive peaks is an exponential decay curve.

The project is based on the MATLAB GUI(graphical user interface).We have plotted damped sine wave by fetching data from excel file,finding successive peak values and also get time corresponding to peak value with the help of MATLAB GUI. We have calculated logarithmic decrement with the help of any two successive peak values and finally calculated damping ratio(zita).Hence we could calculate that how much the amplitude of the vibrational signal decays.

INTRODUCTION

Before going into details about MATLAB GUI let us discuss brief theory of vibration signal and get acquainted with few important terms about MATLAB GUI which we will be using later on.

Brief Theory:

Damped Sine Wave:

A **damped sine wave** is a sinusoidal function whose amplitude approaches zero as time increases.

Damped sine wave represented as

Y=`Amplitude*exp(-Time*0.1).*sin(2*pi*Frequency*Time)`

Where damping factor is 0.1

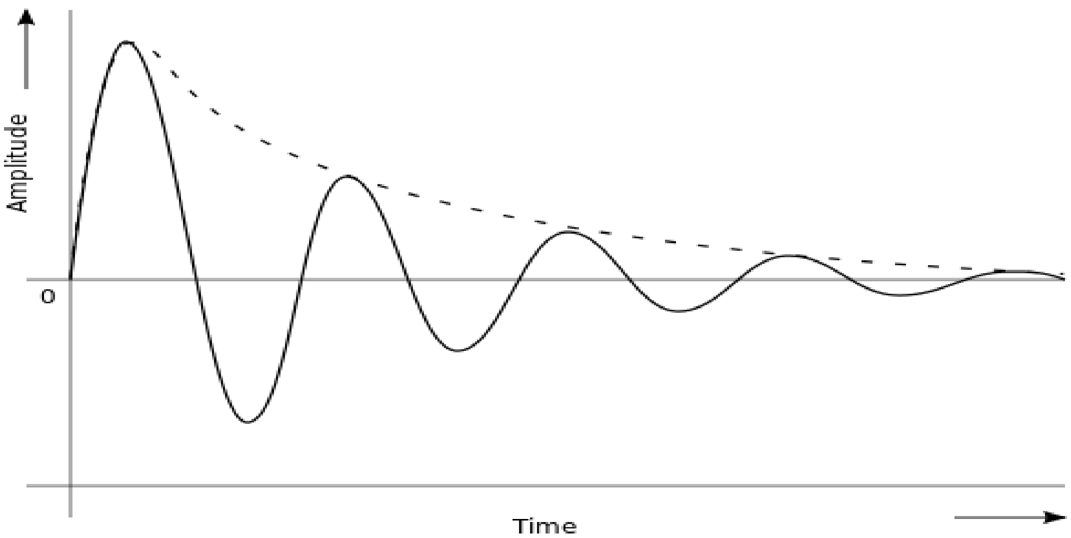


Figure 1: Plot of damped sine wave

Actual Signal:

The term **Actual Signal** is used primarily to denote signals.The signal which is defined as electromagnetic wave used to carry original information and it is free from any noise

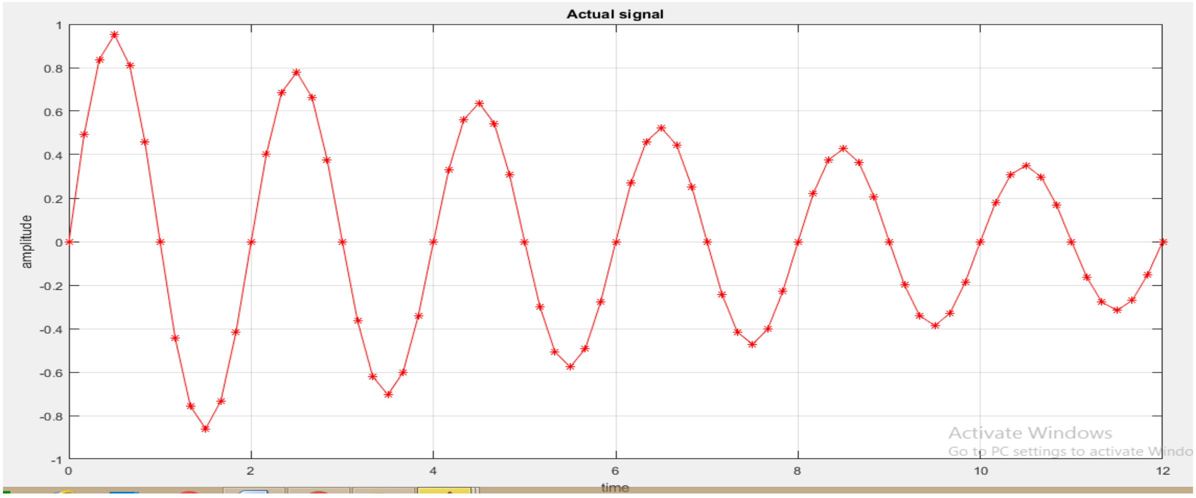


Figure 2: Plot of actual signal

Random signal:

The term **Random Signal** is used primarily to denote signals.which have random in nature,As example we can mention thermal noise,which is created by random movement of electrons in an electric conductor.

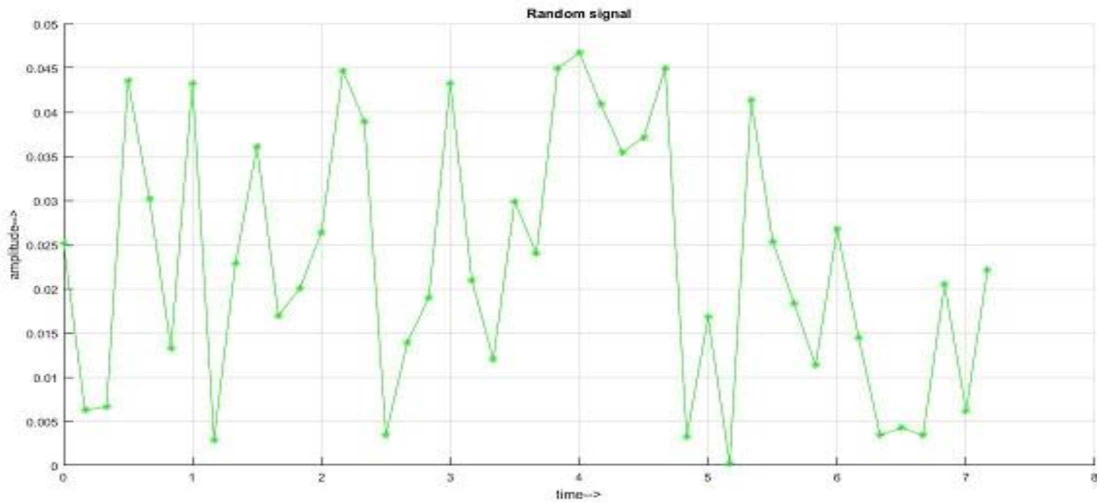


Figure 3: Plot of random signal

Resultant Signal:

The **Resultant Signal** is the algebraic sum of Actual signal and Random signal.

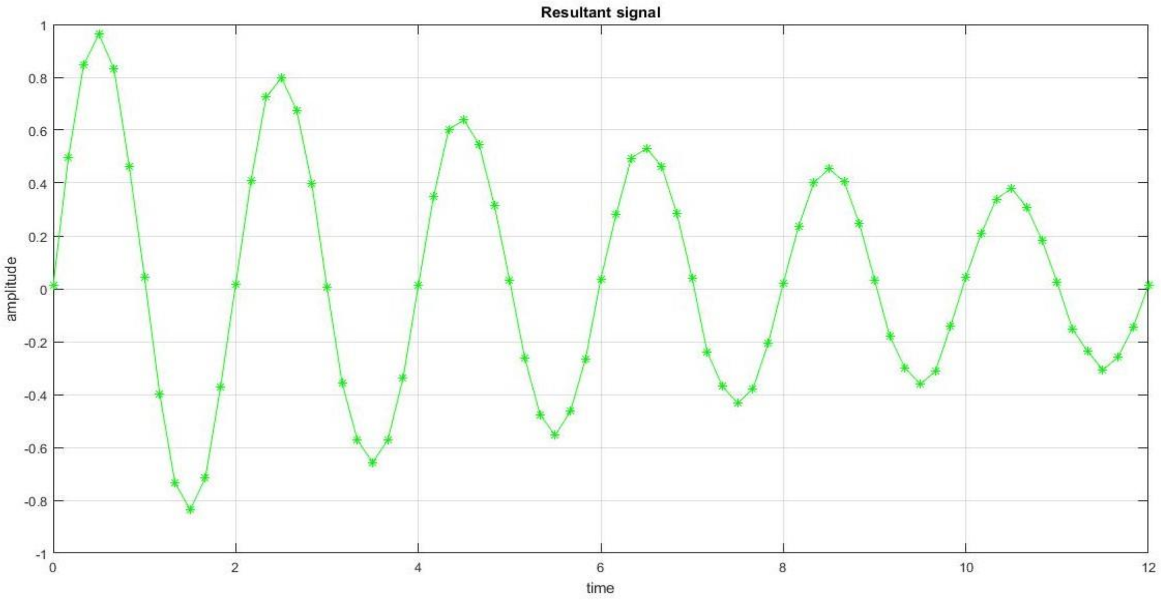


Figure 4: Plot of resultant signal

Logarithmic Decrement:

The **Logarithmic Decrement** represents the rate at which the amplitude of a free damped vibration decreses .It is defined as the natural logarithm of the ratio of any two successive amplitude .It is found from the time response of under damped vibration(oscilloscope or real time analyzer).

Logarithmic decrement is used to find the damping ratio of an under damped system in the time domain.

The method of logarithmic decrement becomes less and less precise as the damping ratio increases past about 0.5.It does not apply at all for a damping ratio greater than 1.0 because the system is over damped.

The logarithmic decrement is defined as the natural log of the ratio of the amplitudes of any two successive peaks.Let us assume that there are two successive peak values x1(t) and x2(t) respectively.

Logarithmic Decrement(w) = ln{(x1(t)) / (x2(t)}

Damping:

Damping is an influence within or upon an oscillatory system that has the effect of reducing, restricting or preventing its oscillations. In physical systems, damping is produced by processes that dissipate the energy stored in the oscillation. Examples include viscous drag in mechanical systems, resistance in electronic oscillators, and absorption and scattering of light in optical oscillators. Damping not based on energy loss can be important in other oscillating systems such as those that occur in biological systems and bikes.

Damping Ratio:

The **Damping Ratio** is a dimensionless measure describing how oscillations in a system decay after a disturbance. Many systems exhibit oscillatory behavior when they are disturbed from their position of static equilibrium. A mass suspended from a spring, for example, might, if pulled and released, bounce up and down. On each bounce, the system tends to return to its equilibrium position, but overshoots it. Sometimes losses (e.g. frictional) damp the system and can cause the oscillations to gradually decay in amplitude towards zero or attenuate. The damping ratio is a measure describing how rapidly the oscillations decay from one bounce to the next.

The **Damping Ratio** is a system parameter, denoted by zeta(ζ), that can vary from undamped ($\zeta = 0$), under damped ($\zeta < 1$) through critically damped ($\zeta = 1$) to over damped ($\zeta > 1$).

The damping ratio is then found from the logarithmic decrement by:

$$\boxed{\zeta = \frac{1}{\sqrt{1 + (\frac{2\pi}{\delta})^2}}}$$

Damping Factor:

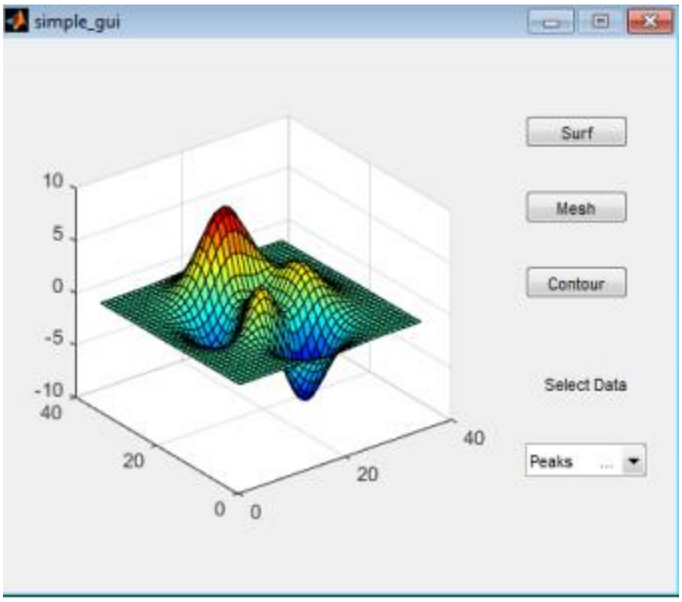
When a vibrating system is **damped**, its energy gets continually dissipated. The amplitude of vibration decreases regularly and the system finally comes to rest. Damping factor It is also known as damping ratio. It is a dimensionless quantity. It can be defined as "The ratio of actual damping to the critical damping."

Calculation of Damping Factor and Damping Coefficient for system:

The first step is to calculate the ratio of amplitudes. The second step is to calculate the logarithmic decrement. The logarithmic decrement is equal to the natural log of the ratio of the amplitudes divided by the number of cycles.

Some Fact About MATLAB GUI

A user interface (UI) is a graphical display in one or more windows containing controls, called components, that enable a user to perform interactive tasks. The user does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user does not need to understand the details of how the tasks are performed. UI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders—just to name a few. UIs created using MATLAB® tools can also perform any type of computation, read and write data files, communicate with other UIs, and display data as tables or as plots. The following figure illustrates a simple UI that you can easily build yourself.



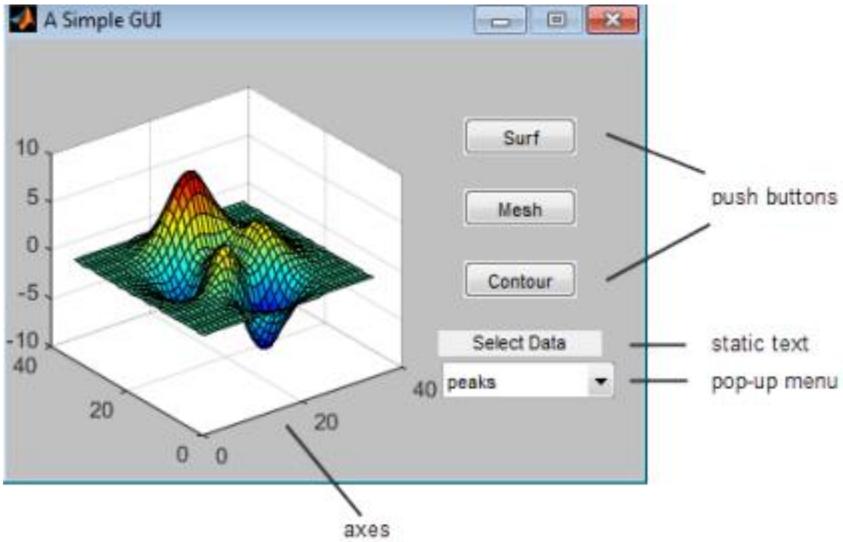
The UI contains these components:

- An axes component
- A pop-up menu listing three data sets that correspond to MATLAB functions: peaks, membrane, and sinc.
- A static text component to label the pop-up menu
- Three buttons that provide different kinds of plots: surface, mesh, and contour When you click a push button, the axes component displays the selected data set using the specified type of 3-D plot.

How to Create a UI with GUIDE

Create a Simple UI Using GUIDE

This example shows how to use GUIDE to create a simple user interface (UI), such as shown in the following figure.



Subsequent topics guide you through the process of creating this UI.

If you prefer to view and run the code that created this UI without creating it, set your current folder to one to which you have write access. Copy the example code and open it in the Editor by issuing the following MATLAB commands:

copyfile(fullfile(docroot, 'techdoc','creating_guis',...
'examples','simple_gui*.*')),

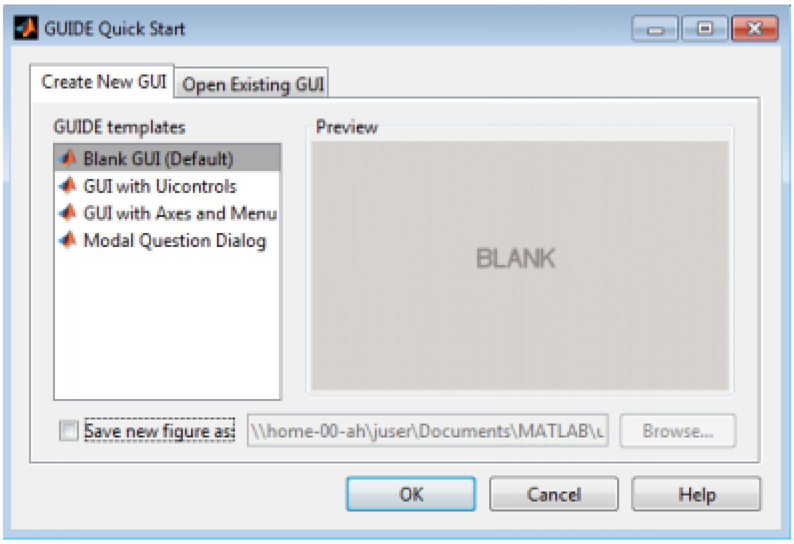
fileattrib('simple_gui*.*', '+w');

guide simple_gui.fig;

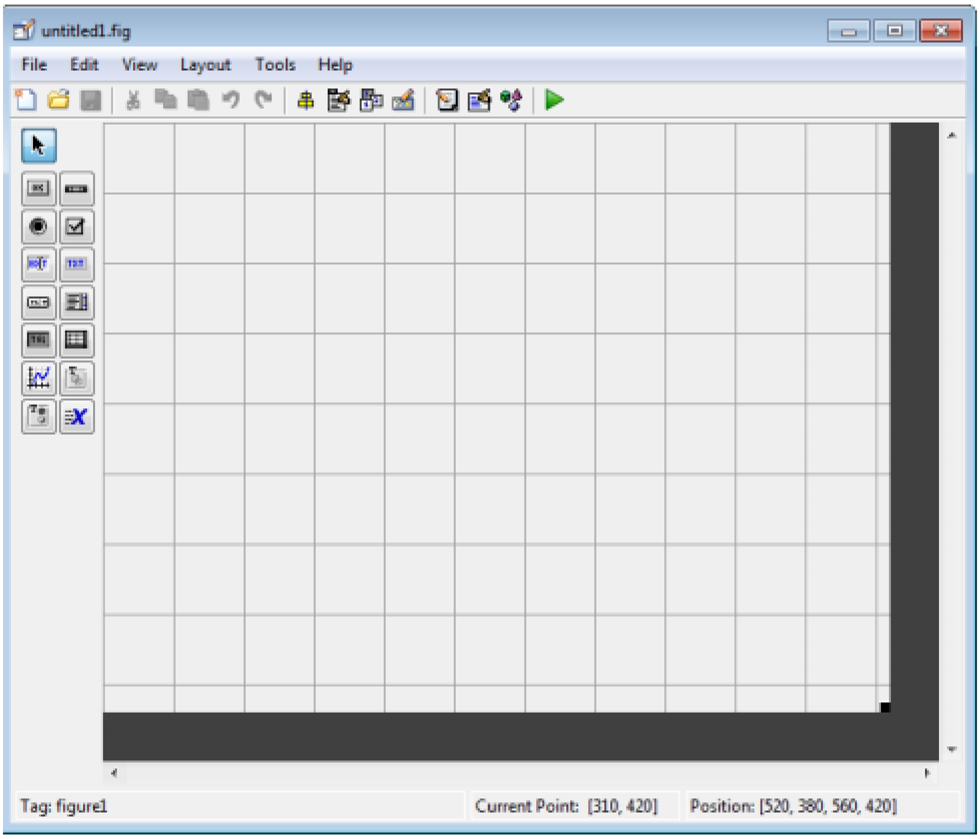
edit simple_gui.m To run the UI, on the Editor tab, in the Run section, click Run .

Open a New UI in the GUIDE Layout Editor

1. Start GUIDE by typing guide at the MATLAB prompt.



2. In the GUIDE Quick Start dialog box, select the Blank GUI (Default) template, and then click OK.

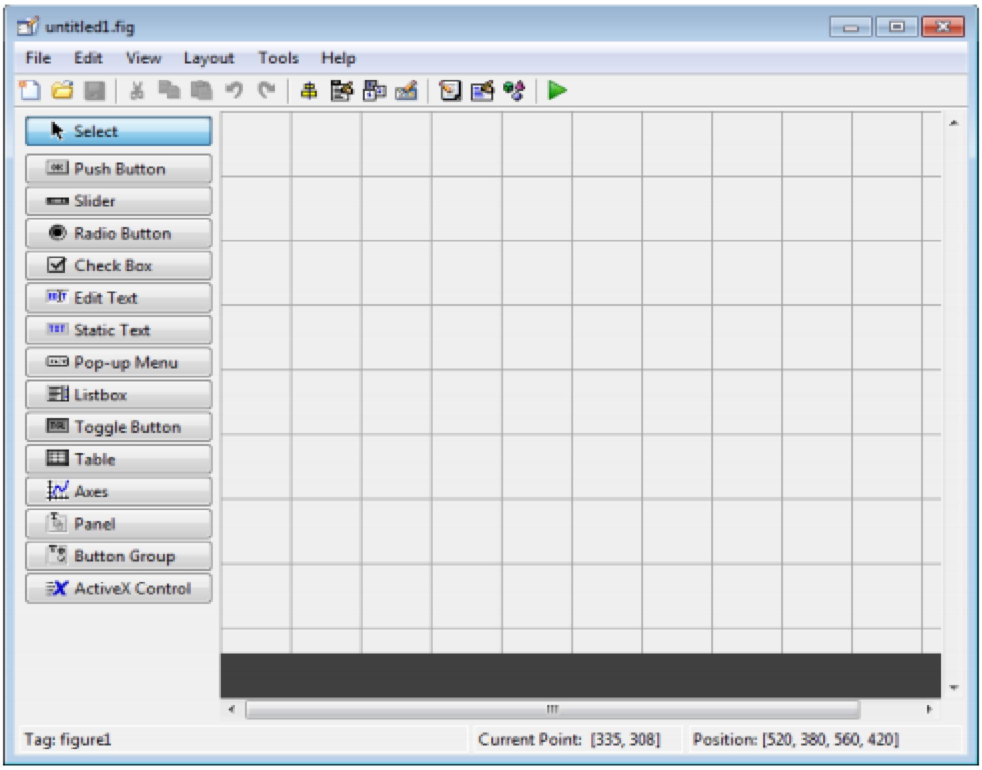


3.Display the names of the UI components in the component palette:

a Select File > Preferences > GUIDE.

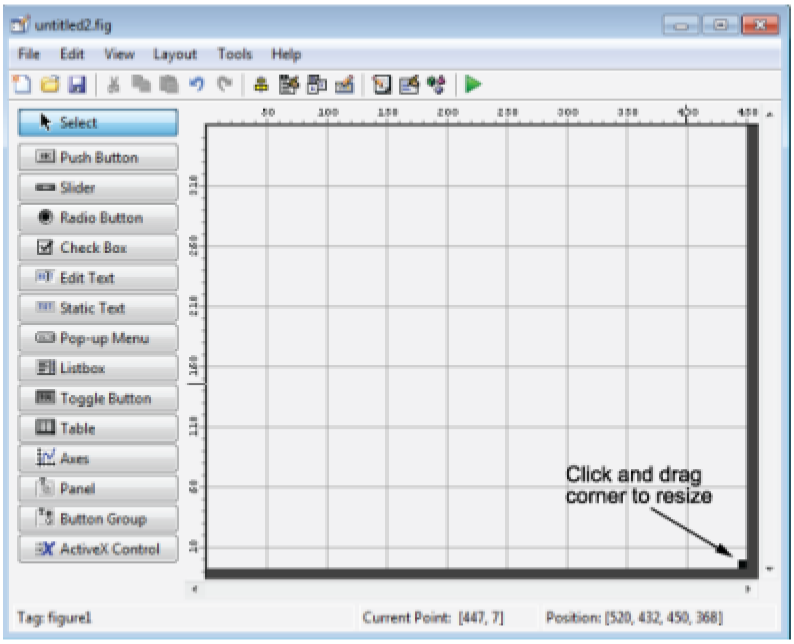
b Select Show names in component palette.

c Click OK.



Set the Window Size in GUIDE

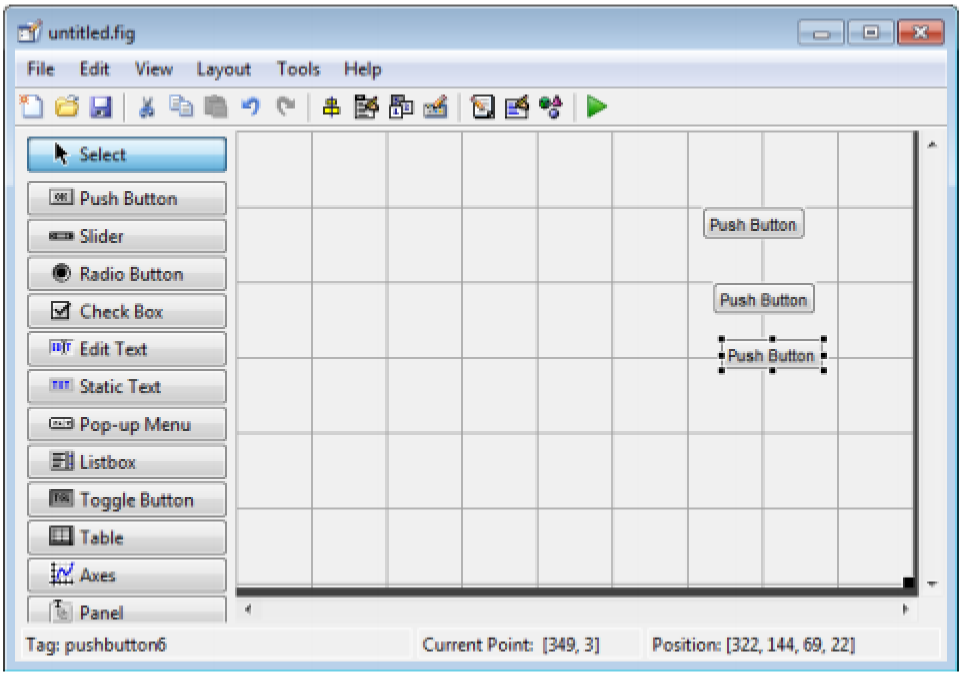
Set the size of the UI window by resizing the grid area in the Layout Editor. Click the lower-right corner and drag it until the canvas is approximately 3 inches high and 4 inches wide. If necessary, make the canvas larger.



Layout the Simple GUIDE UI

Add, align, and label the components in the UI.

1 Add the three push buttons to the UI. Select the push button tool from the component palette at the left side of the Layout Editor and drag it into the layout area. Create three buttons, positioning them approximately as shown in the following figure.

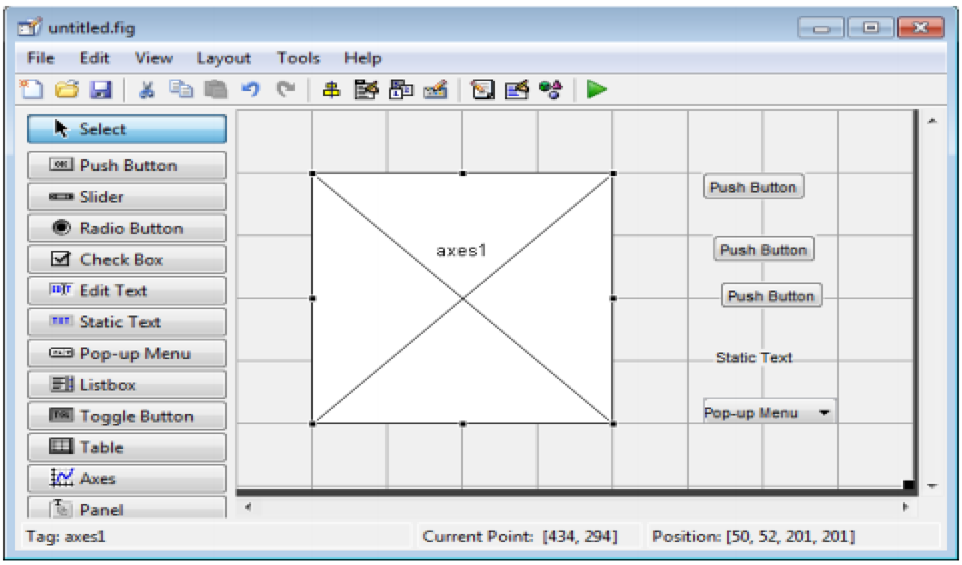


2 Add the remaining components to the UI.

• A static text area

• A pop-up menu

• An axes Arrange the components as shown in the following figure. Resize the axes component to approximately 2-by-2 inches.



Align the Components

If several components have the same parent, you can use the Alignment Tool to align them to one another. To align the three push buttons:

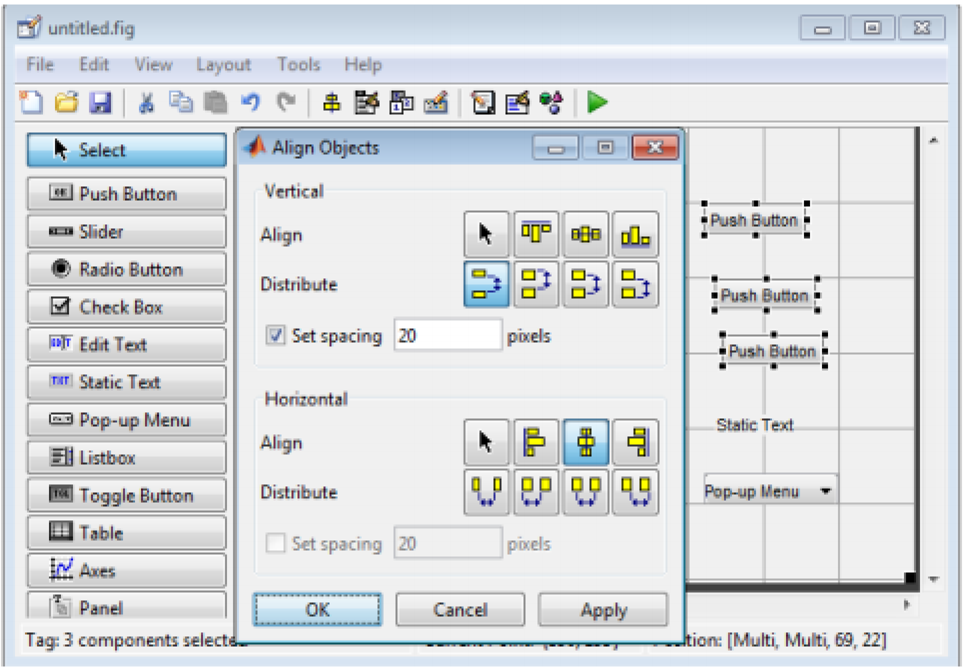
1 Select all three push buttons by pressing Ctrl and clicking them.

2 Select Tools > Align Objects.

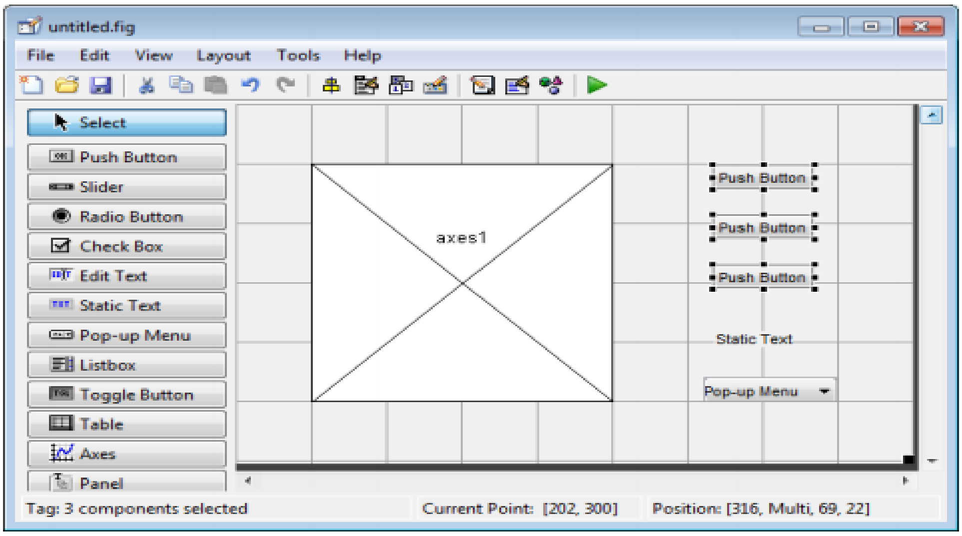
3 Make these settings in the Alignment Tool:

• Left-aligned in the horizontal direction.

• 20 pixels spacing between push buttons in the vertical direction.



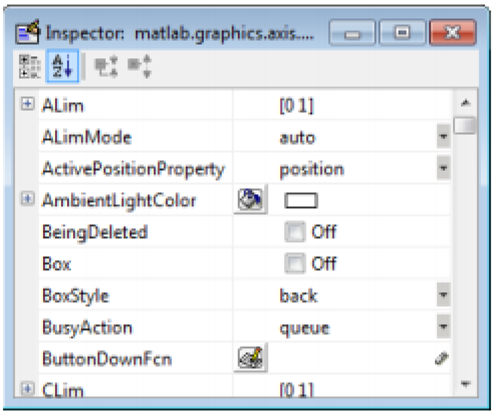
4 Click OK.



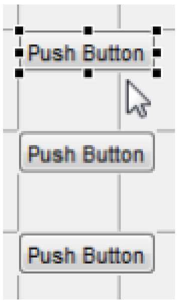
Label the Push Buttons

Each of the three push buttons specifies a plot type: surf, mesh, and contour. This topic shows you how to label the buttons with those options.

1 Select **View > Property Inspector**



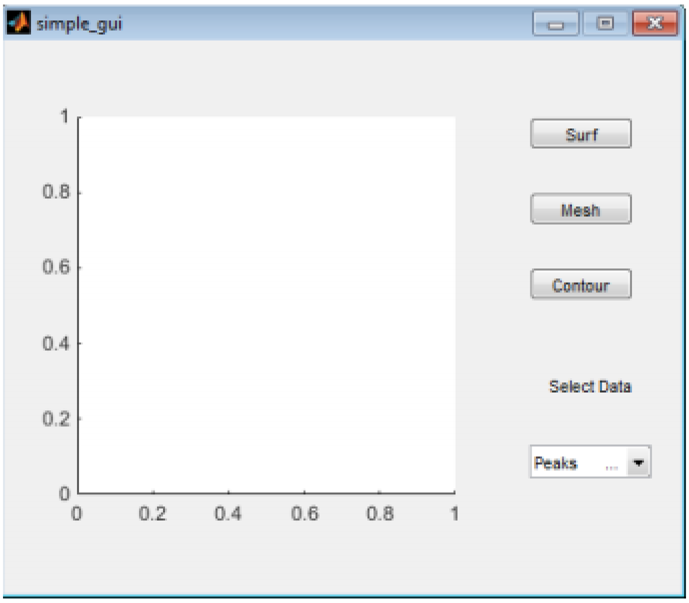
2 In the layout area, click the top push button



3 In the layout area, click the top push button

4 Click outside the String field. The push button label changes to Surf.

5 Click each of the remaining push buttons in turn and repeat steps 3 and 4. Label the middle push button Mesh, and the bottom button Contour.



To run a program created with GUIDE without opening GUIDE, execute its code file by typing its name.

simple_gui

You can also use the run command with the code file, for example,

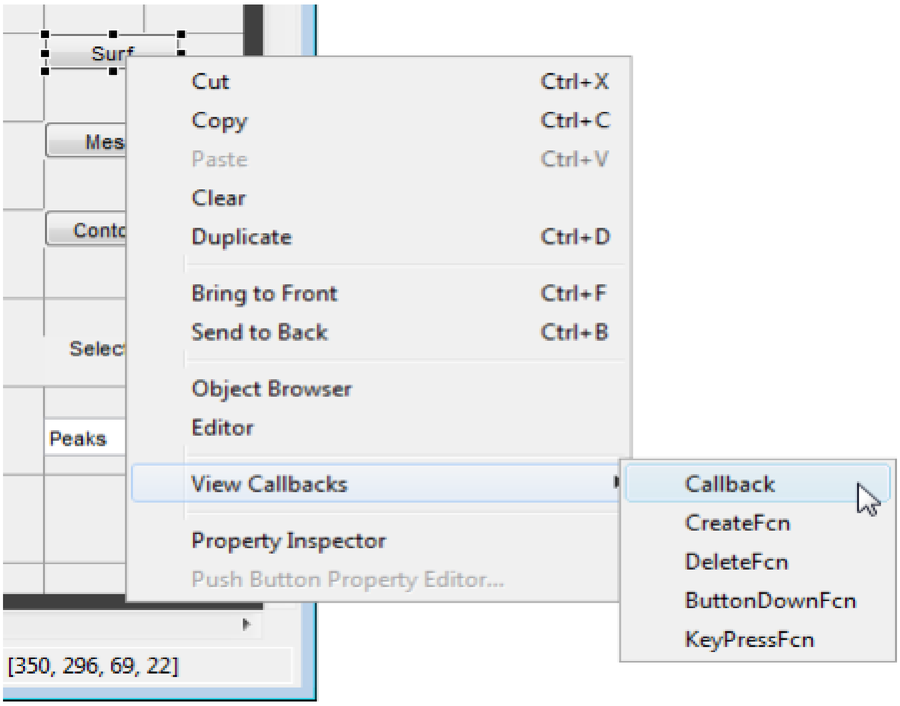
run simple_gui

Note: Do not attempt to run your program by opening its FIG-file outside of GUIDE. If you do so, the figure opens and appears ready to use, but the UI does not initialize and its callbacks do not function.

Code Push Button Behaviour

Each of the push buttons creates a different type of plot using the data specified by the current selection in the pop-up menu. The push button callbacks get data from the handles structure and then plot it.

1. Display the Surf push button callback in the MATLAB Editor. In the Layout Editor, right-click the Surf push button, and then select View Callbacks > Callback.



In the Editor, the cursor moves to the Surf push button callback in the UI code file, which contains this code:

% --- Executes on button press in pushbutton1. function pushbutton1_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

Add the following code to the callback immediately after the comment that begins % handles..

2. % Display surf plot of the currently selected data. surf(handles.current_data);

3. Repeat steps 1 and 2 to add similar code to the Mesh and Contour push button callbacks.

4. Add this code to the Mesh push button callback, pushbutton2_Callback:

5. % Display mesh plot of the currently selected data. mesh(handles.current_data);

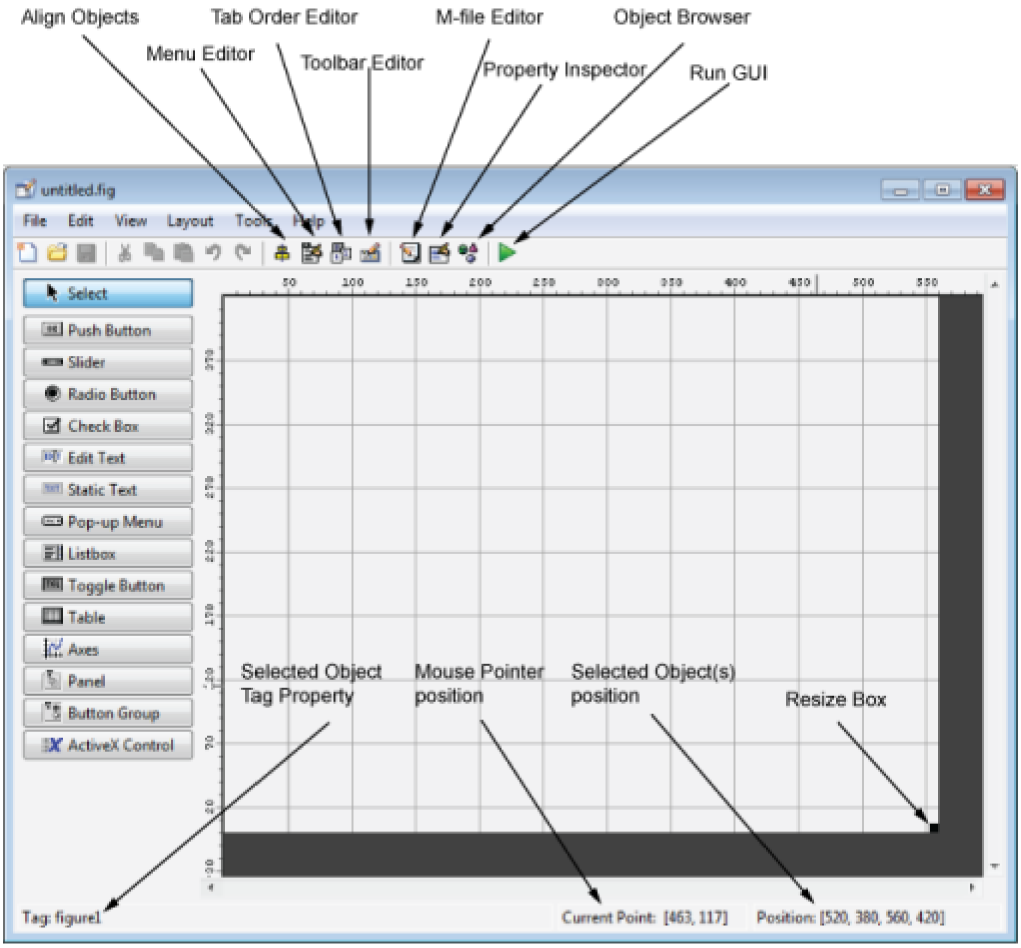
6. Add this code to the Contour push button callback, pushbutton3_Callback:

7.% Display contour plot of the currently selected data. contour(handles.current_data);

8. Save your code by selecting File > Save.

GUIDE Tools Summary

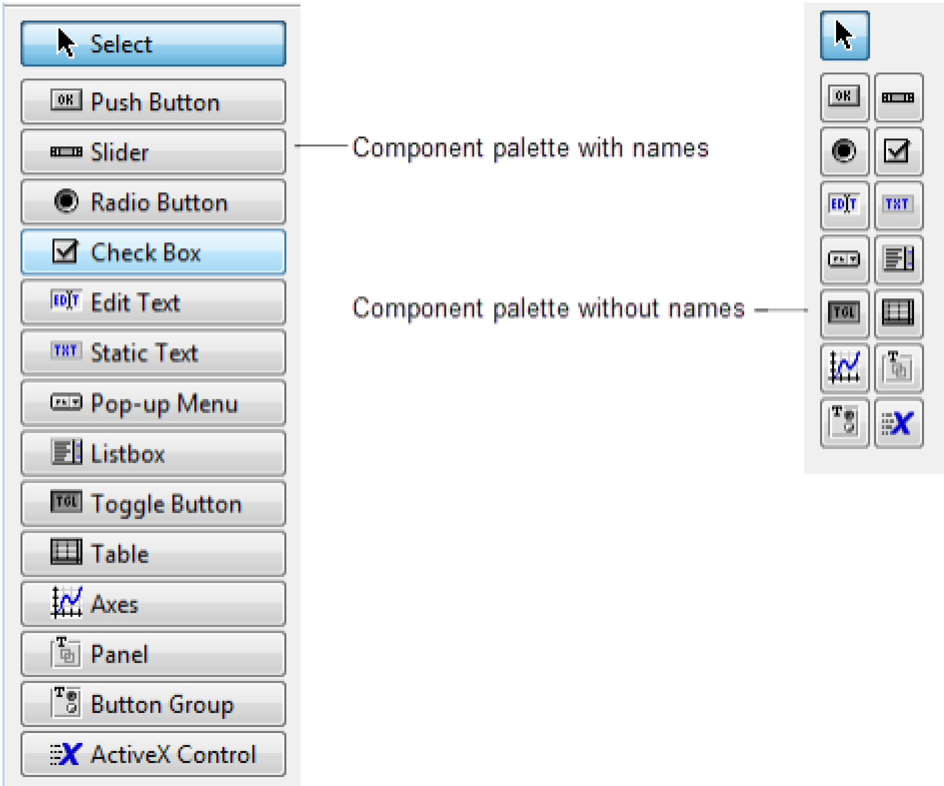
The GUIDE tools are available from the Layout Editor shown in the figure below. The tools are called out in the figure and described briefly below. Subsequent sections show you how to use them.



Use This Tool...	To...
Layout Editor	Select components from the component palette, at the left side of the Layout Editor, and arrange them in the layout area. See "Add Components to the GUIDE Layout Area" on page 6-17 for more information.
Figure Resize Tab	Set the size at which the UI is initially displayed when you run it. See "Set the UI Window Size in GUIDE" on page 6-11 for more information.
Menu Editor	Create menus and context, i.e., pop-up, menus. See "Create Menus for GUIDE UIs" on page 6-91 for more information.
Align Objects	Align and distribute groups of components. Grids and rulers also enable you to align components on a grid with an optional snap-to-grid capability. See "Align GUIDE UI Components" on page 6-79 for more information.
Tab Order Editor	Set the tab and stacking order of the components in your layout. See "Customize Tabbing Behavior in a GUIDE UI" on page 6-88 for more information.
Toolbar Editor	Create Toolbars containing predefined and custom push buttons and toggle buttons. See "Create Toolbars for GUIDE UIs" on page 6-108 for more information.
Icon Editor	Create and modify icons for tools in a toolbar. See "Create Toolbars for GUIDE UIs" on page 6-108 for more information.
Property Inspector	Set the properties of the components in your layout. It provides a list of all the properties you can set and displays their current values.
Object Browser	Display a hierarchical list of the objects in the UI. See "View the GUIDE Object Hierarchy" on page 6-119 for more information.
Run	Save the UI and run the program.
Editor	Display, in your default editor, the code file associated with the UI. See "Files Generated by GUIDE" on page 2-24 for more information.
Position Readouts	Continuously display the mouse cursor position and the positions of selected objects

Show Names in Component Palette

Displays both icons and names in the component palette, as shown below. When unchecked, the icons alone are displayed in two columns, with tooltips.



Show File Extension in Window

Title Displays the Figure-file file name with its file extension, .figure, in the Layout Editor window title. If unchecked, only the file name is displayed.

Show File Path in Window Title

Displays the full file path in the Layout Editor window title. If unchecked, the file path is not displayed.

Add Comments for Newly Generated Callback Functions

Callbacks are blocks of code that execute in response to actions by the user, such as clicking buttons or manipulating sliders. By default, GUIDE sets up templates that declare callbacks as functions and adds comments at the beginning of each one. Most of the comments are similar to the following.

% --- Executes during object deletion, before destroying properties. function figure1_DeleteFcn(hObject, eventdata, handles)

% hObject handle to figure1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

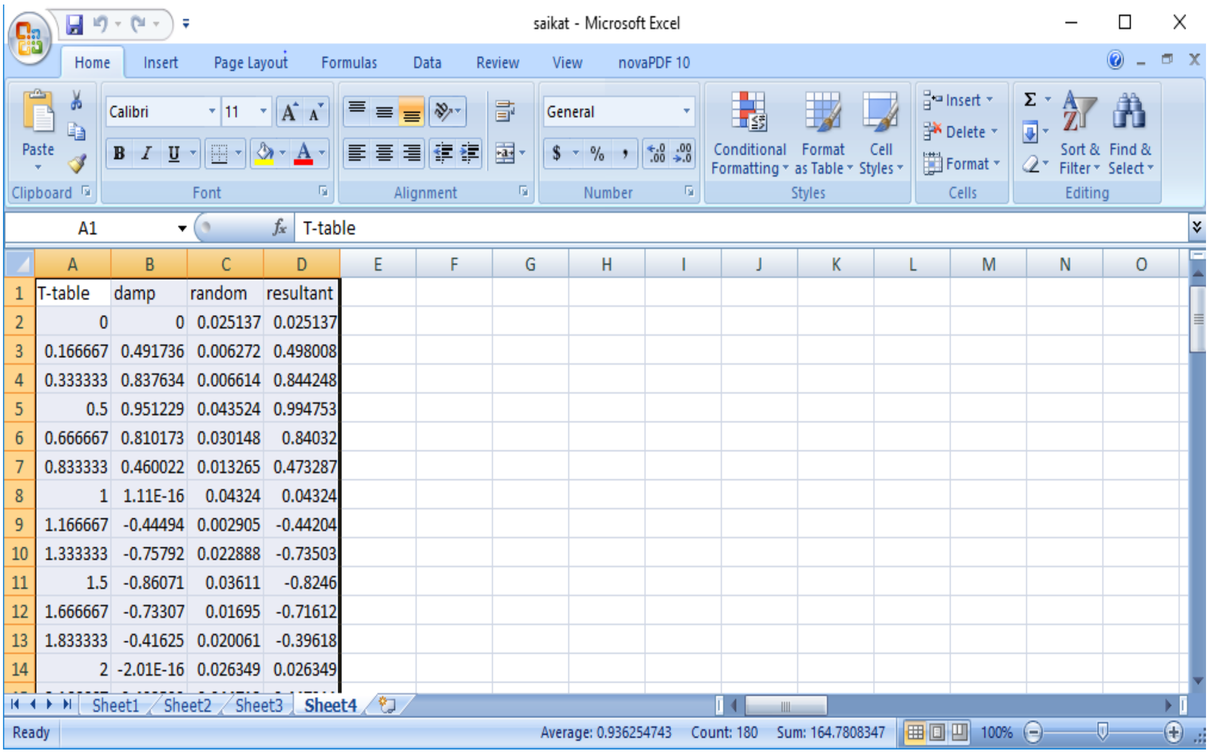
Some callbacks are added automatically because their associated components are part of the original GUIDE template that you chose. Other commonly used callbacks are added automatically when you add components. You can also add callbacks explicitly by selecting them from **View > View Callbacks** menu or on the component's context menu.

If you deselect this preference, GUIDE includes comments only for callbacks that are automatically included to support the original GUIDE template. GUIDE does not include comments for callbacks subsequently added to the code.

SYNTHESIS OF VIBRATION SIGNAL WITH THE HELP OF MATLAB GUI

Basic Assumption:

1. Take an excel sheet ('*.xlsx') from the user.
2. Read the excel sheet data.
3. In excel sheet,1st column must be the time value.
4. In excel sheet,2nd column must be the amplitude data of the actual signal.
5. In excel sheet,3rd column must be the amplitude data of the random signal.
6. In excel sheet,4th column must be the amplitude data of the resultant(addition of actual signal and random signal) signal.



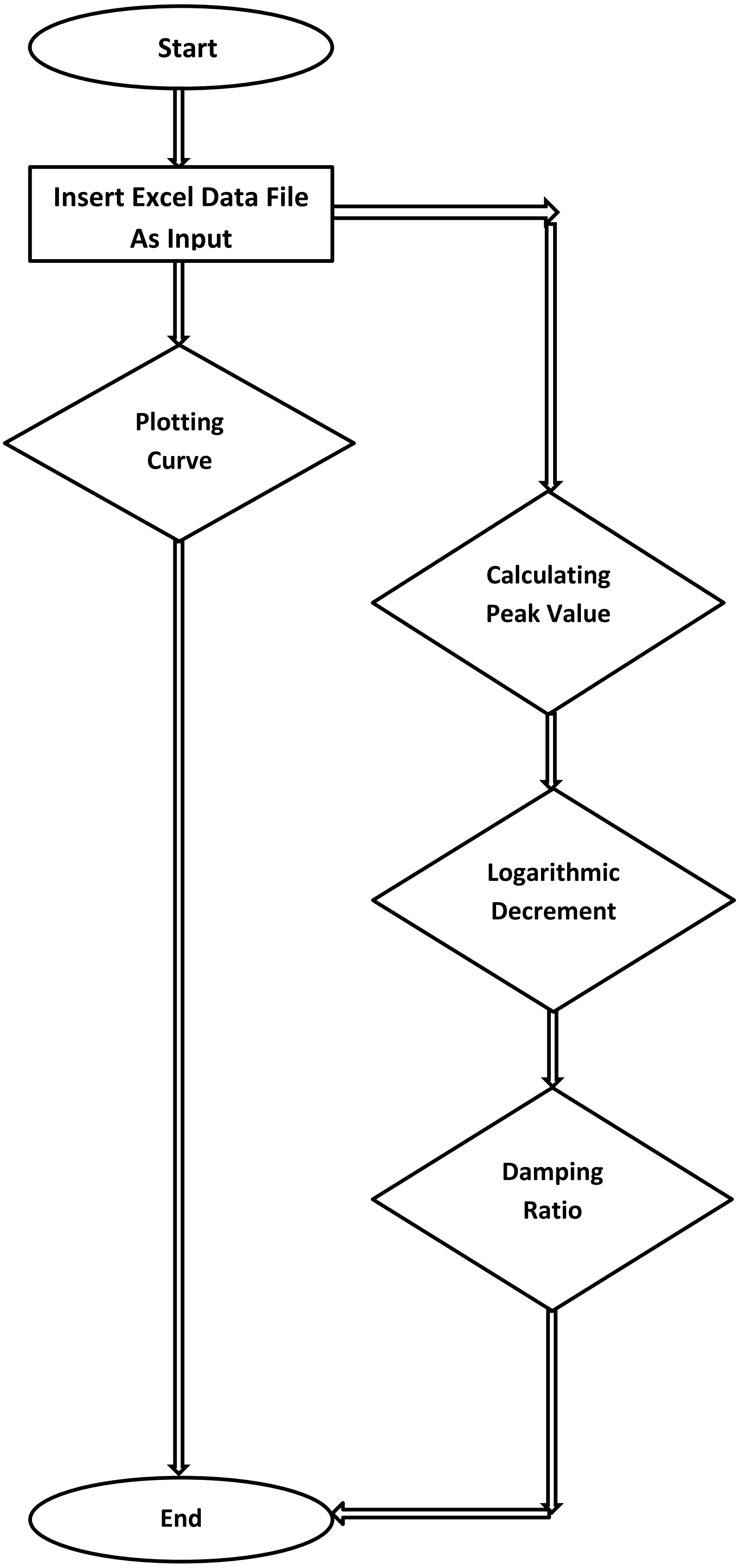
T-table	damp	random	resultant
0	0	0.025137	0.025137
0.166667	0.491736	0.006272	0.498008
0.333333	0.837634	0.006614	0.844248
0.5	0.951229	0.043524	0.994753
0.666667	0.810173	0.030148	0.84032
0.833333	0.460022	0.013265	0.473287
1	1.11E-16	0.04324	0.04324
1.166667	-0.44494	0.002905	-0.44204
1.333333	-0.75792	0.022888	-0.73503
1.5	-0.86071	0.03611	-0.8246
1.666667	-0.73307	0.01695	-0.71612
1.833333	-0.41625	0.020061	-0.39618
2	-2.01E-16	0.026349	0.026349

- ✓ User have to noticed the name of the excel sheet which sheet they have stored the whole data.[e.g.sheet1,sheet2,sheet3,etc]
- ✓ User must be known the range of data which they have used.[e.g.A1:D24,A1:D34,etc]

7. Plot (time vs amplitude) graph of the vibrational signal by fetching the excel sheet data.

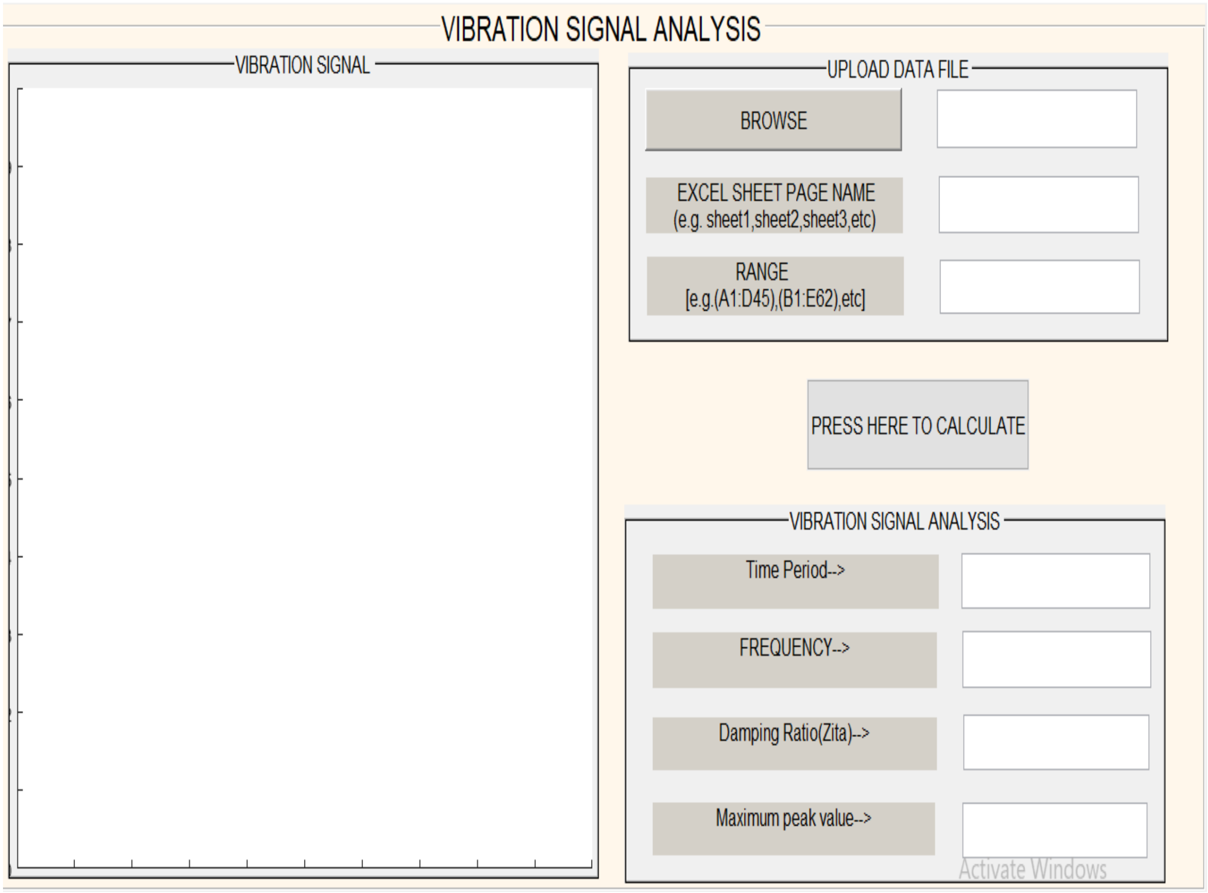
8. Hence we have calculated the maximum peak values of resultant signal,Logarithmic Decrement and Damping Ratio(Zita).
9. Also calculate the time period and frequency.

Flowchart Representing MATLAB GUI Algorithm



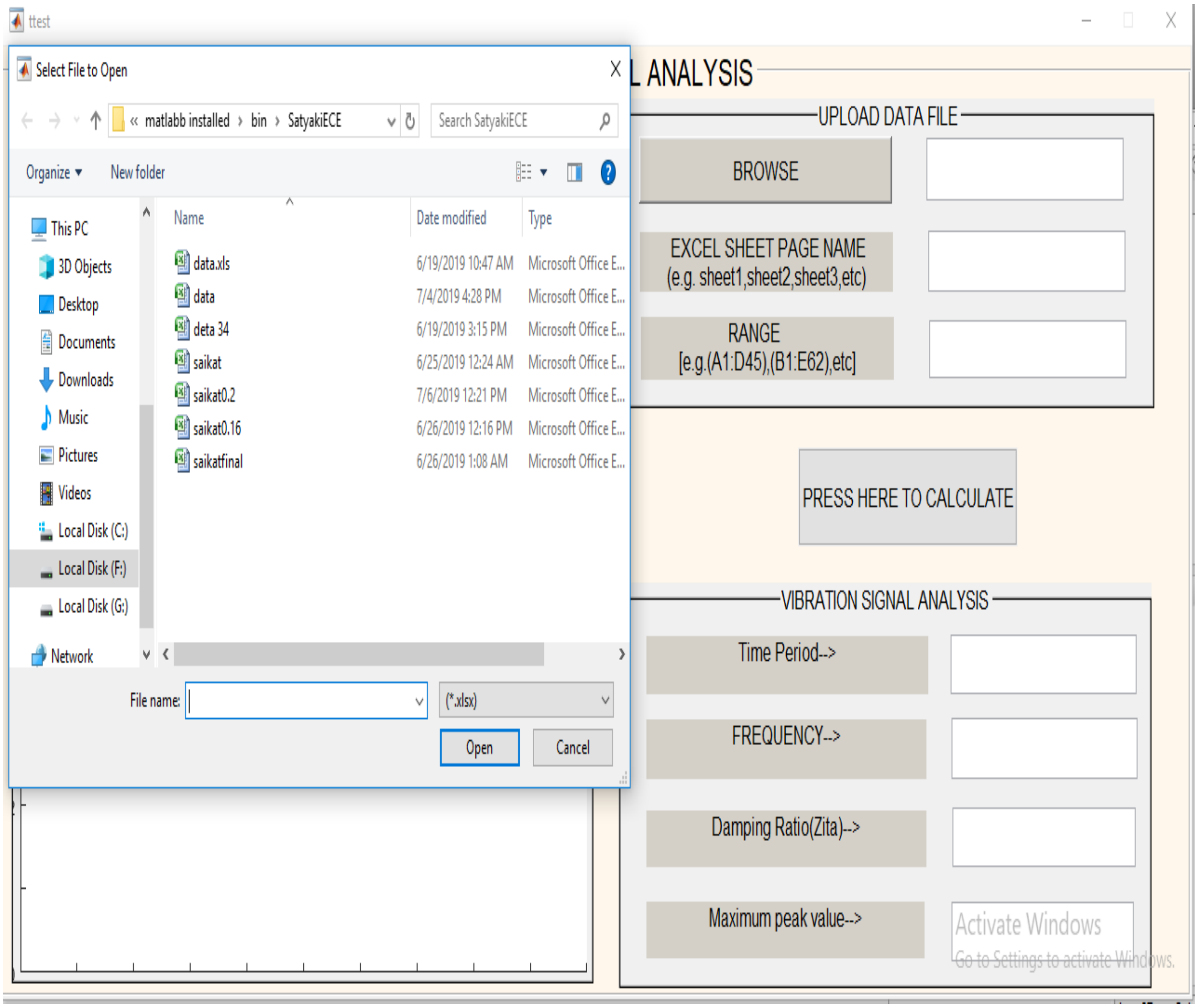
METHODOLOGY 1:

As users have many excel file('*.xlsx') saved in editor file in matlab. When user click the run option after the process of compile the m-file code there is come a user's interface of vibration signal analysis,in front of their screen. There is a browse option to uplod excel file. After fetching excel file we are plot the vibrational signals (Actual,Random and Resultant Signal) and calculate peak values and time corresponding peak values.We have find out logarithmic decrement with the help of two successive peak values of the resultant signal. Hence we calculate damping ratio(zita).We have shown the user interface of the vibration signals analysis given below.



METHODOLOGY 2:

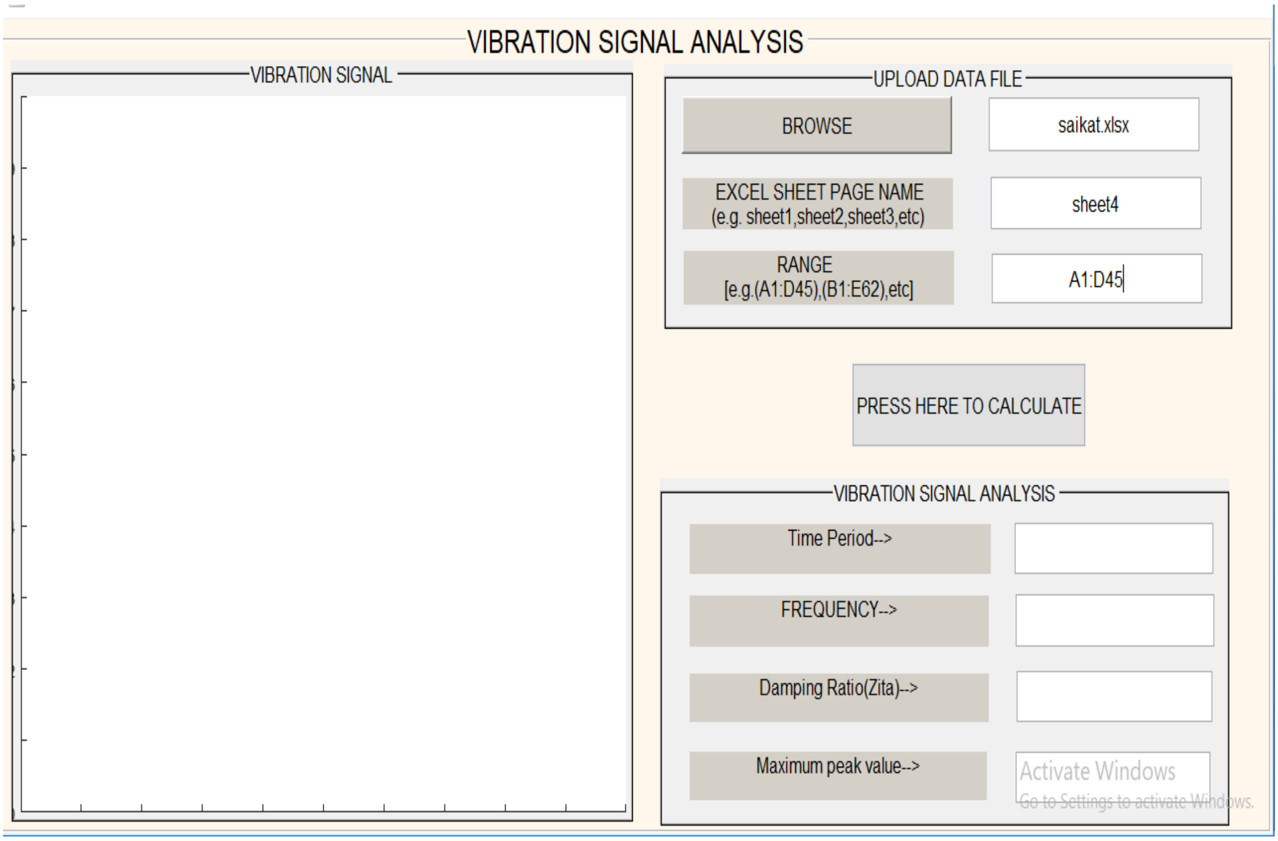
When any user press the **BWROSE** button therefore the screen be like:



Then user have to choose any excel file ('*.xlsx') which they have been already saved .

METHODOLOGY 3:

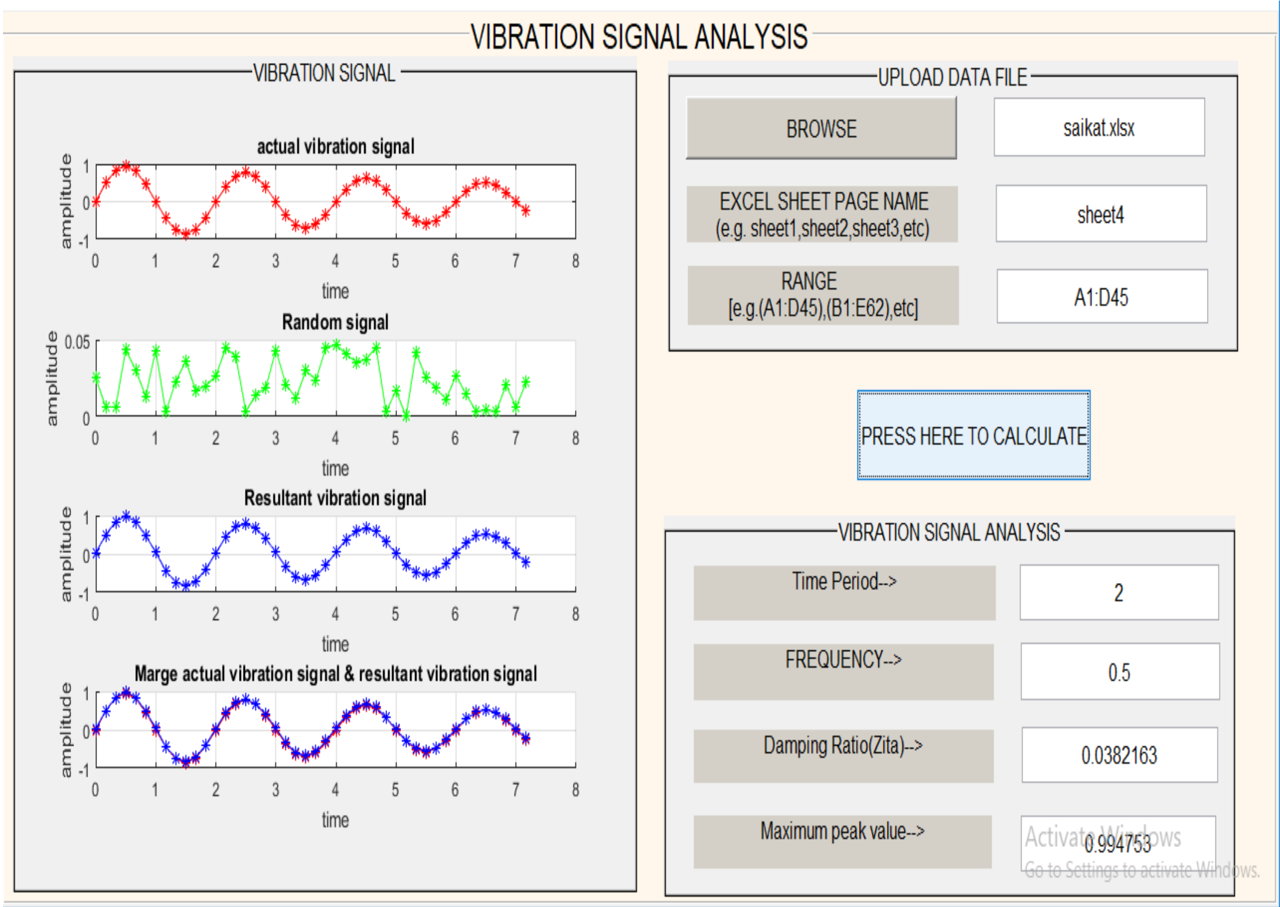
There after user have to give the sheet name (e.g.sheet1,sheet2,etc) in the blank portion beside the **Excel Sheet Page Name,** and also give the excel sheet (data)range beside the **Range** in this figure.



Therefore user have to press the **PRESS HERE TO CALCULATE** button and final output given below:

FINAL RESULTANT FIGURE:

After pressing the **PRESS HERE TO CALCULATE** button .The final output be like:



Vibration signals :

Here we have to see the plot of vibration signals (Actual, Random ,Resultant and finally marge the both of Actual and Resultant Signal).The figure given below

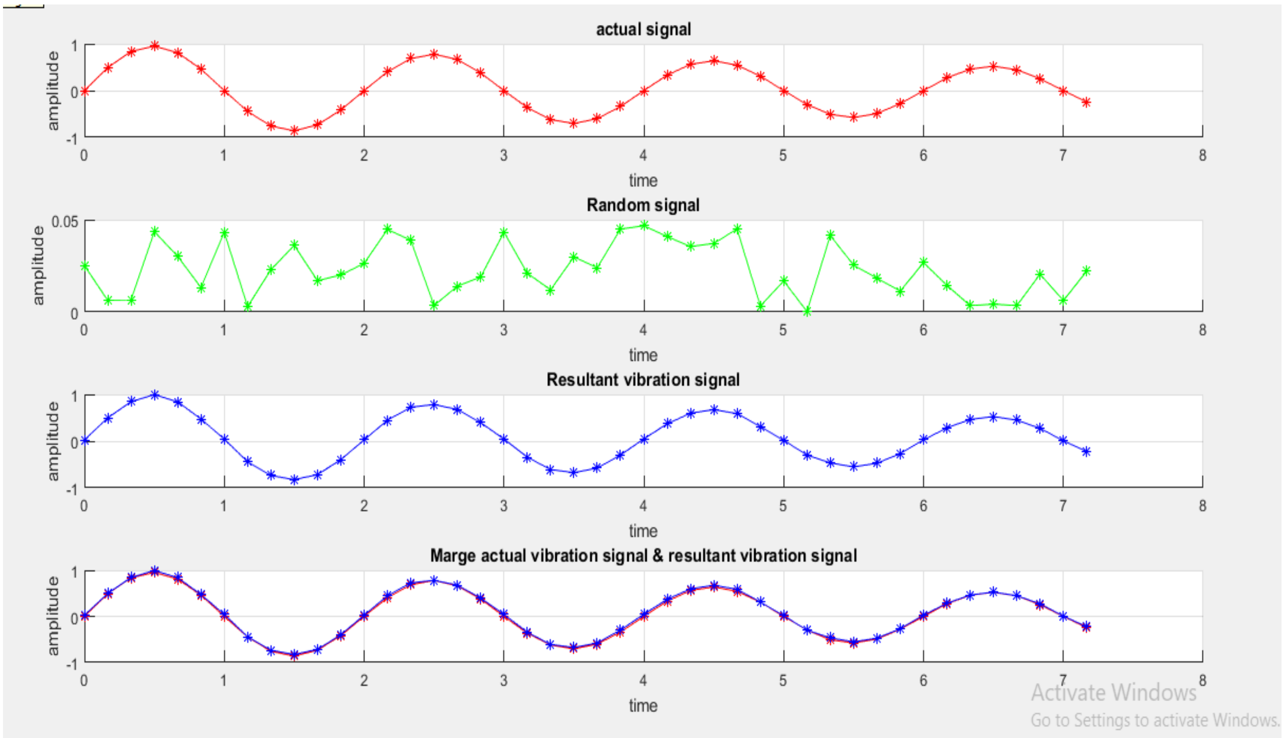
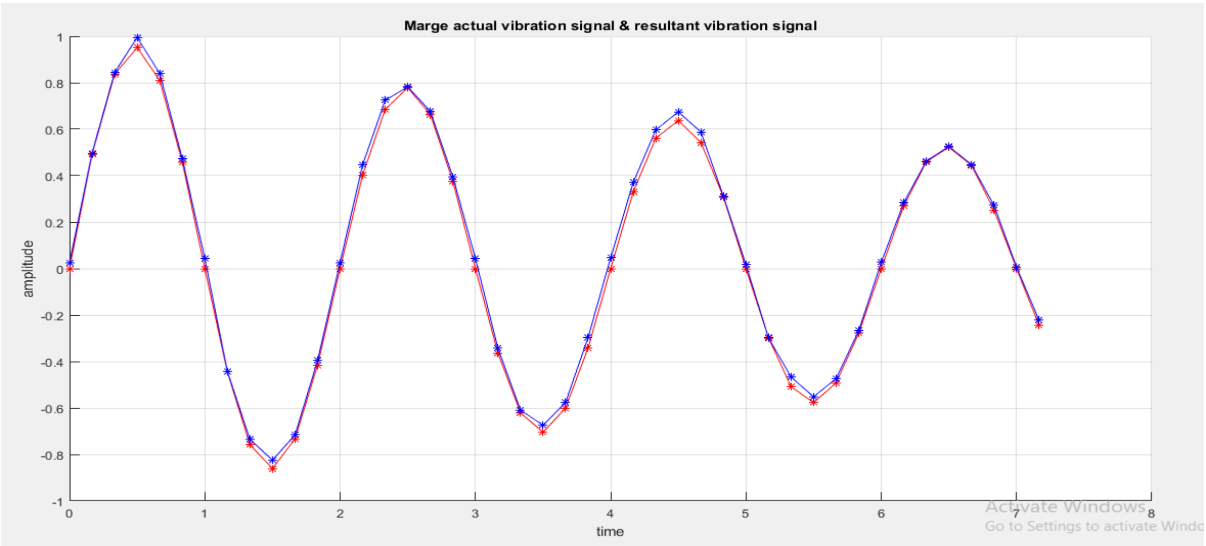
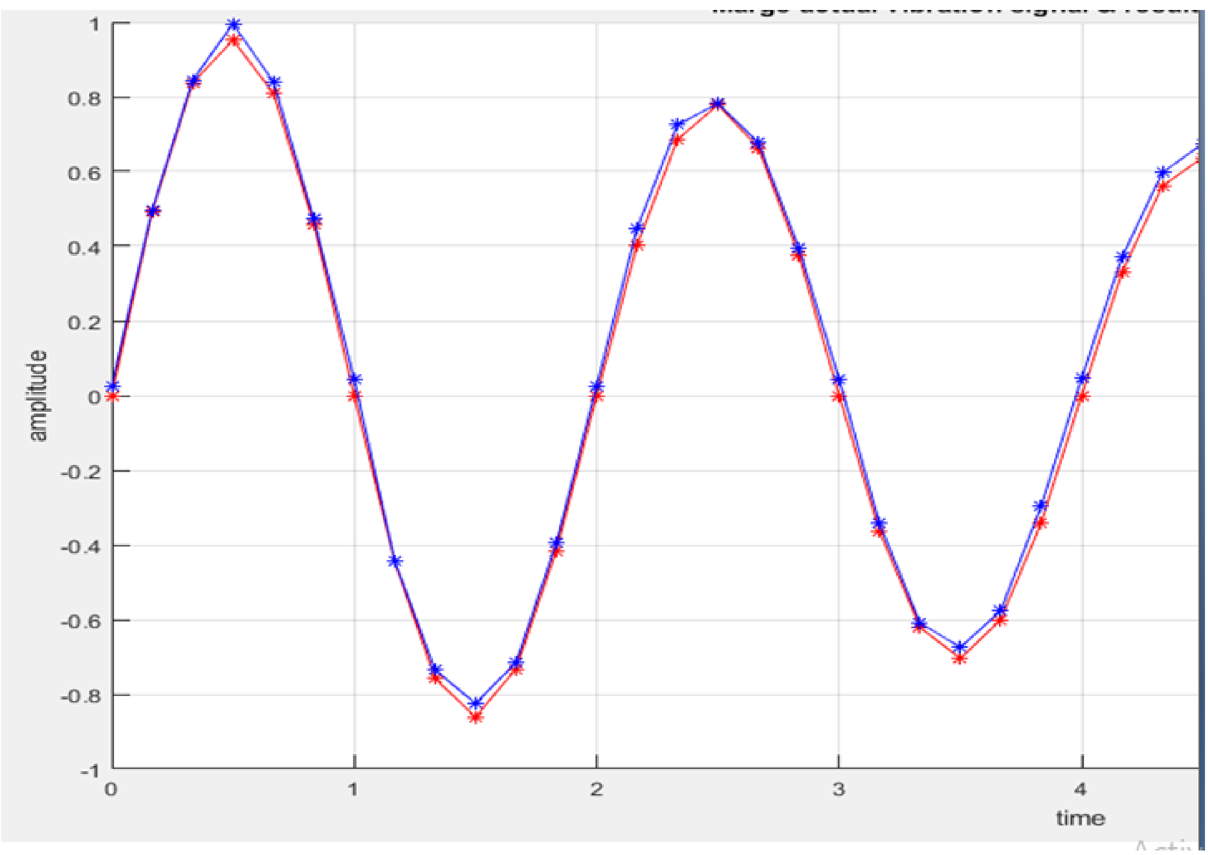


Figure 5: plot of vibration signal.

After we have to plot of the zooming out of this marge actual vibration and resultant vibration signal.Again we have to see how does the random signal affected the amplitude of the actual signal.

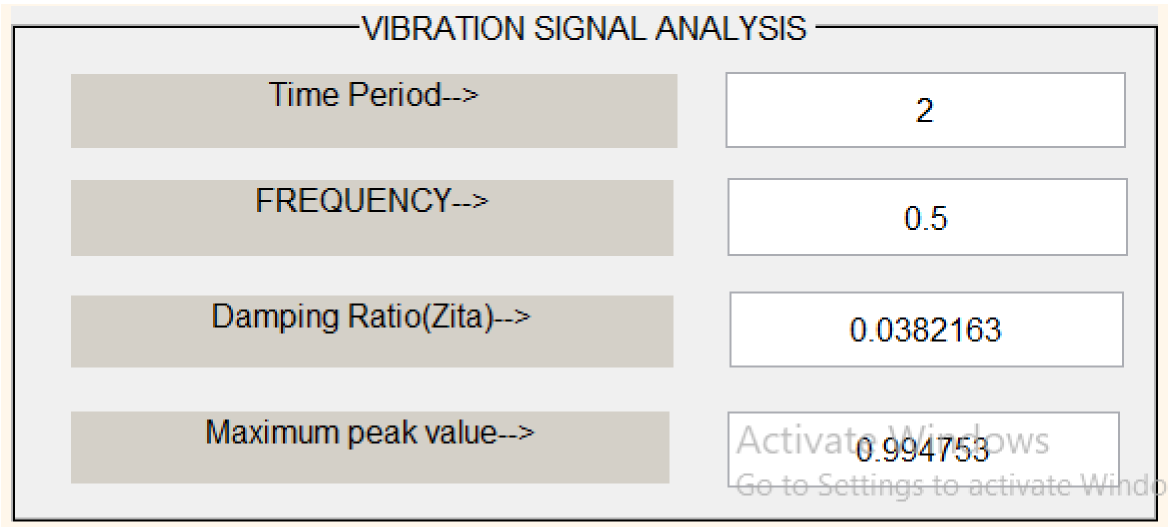




Here red line(__)indicates the actual signal but the blue line(__)indicats how much actual signal affected when we are add the random signal with actual signal.

Vibration Signal Analysis:

We also calculate the time period of those signal ,the maximum value and damping factor(zita).



? INFERENCES:

1. It is clear that fetching the excel file data we have to plot the vibration signal.
2. See how much the actual vibration signal affected when we add the random signal with this.
3. However we have to calculate the maximum peak value of this vibration signal.
4. Used of the two successive peak value we have measure the Damping ratio(zita).
5. If we will change the range of the excel shit data then we get different plotting signal .
6. If we will change the excel shit with the help of BROWSE button therefore we will put the sheet name and the data range in this MATLAB GUI figure then we will get the different output.

CONCLUSION

As a conclusion, we can say that this software offers strong possibilities. The utilisation is pretty for the problem based on the damping ratio of damped sine wave. In this project we have created some algorithm to calculate damping ratio(zita) with the help of two successive peak values.

Understanding the complexity of project,we used MATLAB GUI(Graphical User Interface) in which user have been provided message to upload excel data file with sheet page name and range.Also user have been displayed the plotted curve of the vibrational signals and the numerical value of damping ratio,logarithmic derement,time period and frequency.We have found that the theoretical value of damping ratio is same as the experimental value.

REFERENCES

BOOKS:

1. MATLAB for Beginners: A Gentle Approach By Peter Issa Kattan.
2. MATLAB Programming for Engineers By Stephen J.Chapman.
3. MATLAB and Its Applications in Engineering By Raj Kumar Bansal,Manoj Kumar Sharma, Ashok K. Goel.
4. MATLAB, An Introduction With Applications By Amos Gilat.
5. Graphics and GUIs with MATLAB, Second Edition By Patrick Marchand.
6. MATLAB Graphical Programming: Practical Hands-on MATLAB Solutions By C.Perez.

WEBSITES:

1. https://in.mathworks.com/products/matlab.html
2. https://www.tutorialspoint.com/matlab/
3. https://www.mathworks.com/videos/importing-spreadsheets-into-matlab-68987.html
4. https://stackoverflow.com/questions/15099414/access-to-a-matlab-gui-from-the-web
5. https://www.mathworks.com/discovery/matlab-gui.html