International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

MATLAB GUI based Signal Processing for Various Functions

Mohit Sharma[1], Gaurav Sahu[2]

[1]Student, BK Birla Institute of Engineering and Technology, Pilani
[2]Assistant Professor, BK Birla Institute of Engineering and Technology, Pilani

Abstract: ***A graphical user interface provides the user with a familiar environment in which to work. This environment contains pushbuttons, toggle buttons, lists, menus, text boxes, and so forth, all of which are already familiar to the user, so that he or she can concentrate on using the application rather than on the mechanics involved in doing things. However, GUIs are harder for the programmer because a GUI-based program must be prepared for mouse clicks (or possibly keyboard input) for any GUI element at any time. Such inputs are known as events, and a program that responds to events is said to be event driven. A graphical User interface (GUI) is a system of graphical elements that allow a user to interact with software using mouse application. GUI can also display data in tabular form or as plots and can group related components. MATLAB is widely used technical computing software. MATLAB provides graphical design tools such as Simulink and Dials and Gauges Blockset. This paper deals with the implementation of various MATLAB functions present in signal processing like Trigonometric, exponential, logarithmic, Unit Step, Unit Impulse, Unit Ramp functions. It is based on software.***

Keywords: MATLAB-GUI, Continuous Time Function, Discrete Time Function, Graph

1. Introduction

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks [1]. The GUI 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 GUI user need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders, etc. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots [1][2].

2. Classification of Signals

Analog signal processing– Analog signal processing is for signals that have not been digitized, as in legacy radio, telephone, radar, and television systems. This involves linear electronic circuits as well as non-linear ones. The former are, for instance, passive filters, active filters, additive mixers, integrators and delay lines. Non-linear circuits include voltage-controlled filters, voltage-controlled oscillators and phase-locked loops [3]. Therefore it can be concluded as continuous-time signal processing varies with change of continuous domain. The methods of signal processing include time domain, frequency domain, and complex frequency domain. This technology mainly discusses the modeling of linear time-invariant continuous system, setting up system function and the continuous time filtering of deterministic signals [3].

Discrete-time signal processing–Discrete-time signal processing is for sampled signals, defined only at discrete points in time, and as such is quantized in time, but not in magnitude. Analog discrete-time signal processing is a technology based on electronic devices such as sample and hold circuits, analog time-division multiplexers, analog delay lines and analog feedback shift registers[3].

3. System Overview

The complete system is divided into two modules. The first modules deal in designing the Graphical User Interface using Matlab for variety of signal to be processed. This module deals in representing the output waveform for the type of signal which will be selected from the list of functions. Trigonometric, Inverse Trigonometric, Hyperbolic, Unit, Ramp, Impulse, Log & Exponential functions are represented both in continuous and discrete domains [1]. The second modules deals in designing the axis which is used to display the output graph for the various functions. Axis is used in the GUI and is used for the purpose of displaying output. Options are the output buttons which are used in the defining the graph related with every function. It depends upon the user, whichever option user decides and clicks on it the graph related to that function will be plotted on the designed axis.

4. System Designing

A graphical user interface provides the user with a familiar environment in which to work [4]. This environment contains pushbuttons, toggle buttons, lists, menus, text boxes, and so forth, all of which are already familiar to the user, so that the user can concentrate on using the application rather than on the mechanics involved in doing things [4]. MATLAB includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation and presentation graphics. However, GUIs are harder for the programmer because a GUI-based program must be prepared for mouse clicks (or possibly keyboard input) for any GUI element at any time [4]. The image showing various buttons in the GUI is shown in Figure 1.

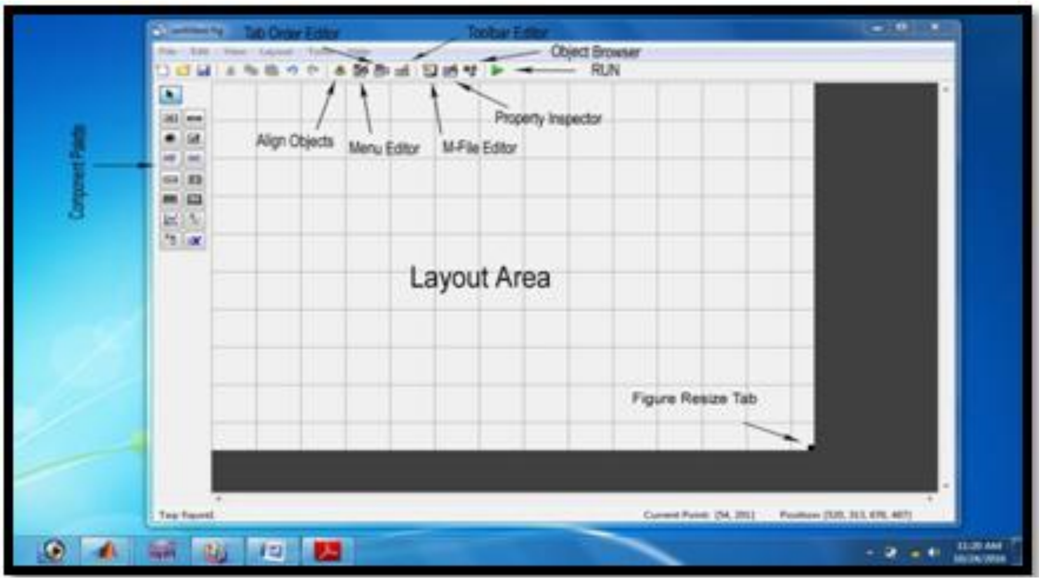

Figure 1: Layout of MATLAB GUI

The main elements required to create MATLAB Graphical User Interface are as follows and is shown in Figure 2.

(a) *Components*- Each item on a MATLAB GUI (pushbuttons, labels, edit boxes, etc.) is a graphical component. The types of components include graphical controls (pushbuttons, edit boxes, lists, sliders, etc.), static elements (frames and text strings), menus, and axes. Graphical controls and static elements are created by the function uicontrol, and menus are created by the functions uimenu and uicontextmenu. Axes, which are used to display graphical data, are created by the function axes [4].

(b) *Figures*-The components of a GUI must be arranged within a figure, which is a window on the computer screen. Empty figures can be created with the function figure and can be used to hold any combination of components [4]. Use graphical elements that perform a dual function: display data and interaction. For example, a plotted line can both display data and can alter data when a user click on the line drag it on a new position. Use mouse downs, drags, and mouse ups anywhere within the figure to perform an action.

(c) *Callbacks* -Finally, there must be some way to perform an action if a user clicks a mouse on a button or types information on a keyboard. A mouse click or a key press is an event, and the MATLAB program must respond to each event if the program is to perform its function. For example, if a user clicks on a button, that event must cause the MATLAB code that implements the function of the button to be executed. The code executed in response to an event is known as a call back. There must be a callback to implement the function of each graphical component on the GUI [4].

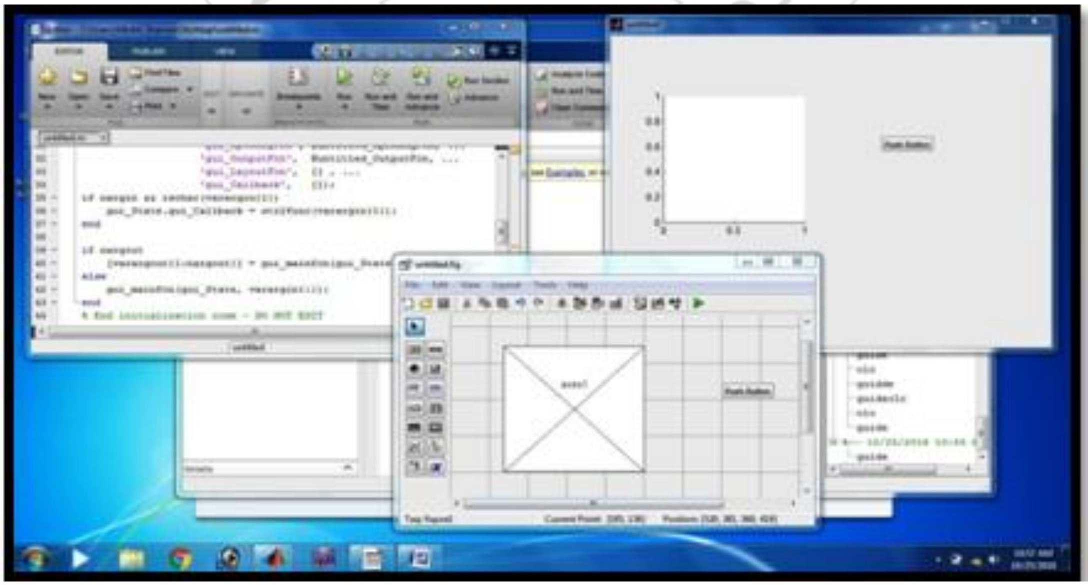

Figure 2: Layout with Components, Figure and Callbacks

4.1 Tools used in Designing the GUI

Table 1: Tools used in designing the GUI

S. No.	*Tools*	*Functions*
1	LayoutEditor	Select components from the component palette, at the left side of the Layout Editor, and arrange them in the layout area. See AddingComponents to the GUI for more information [5].
2	FigureResize Tab	Set the size at which the GUI is initially displayed when you run it. SeeSetting the GUI Size for more information [5].
3	MenuEditor	Create menus and context, i.e., pop-up, menus. See Creating Menus for more information [5].
4	AlignObjects	Align and distribute groups of components. Grids and rulers alsoEnable you to align components on a grid with an optional snap-to-grid Capability. See Aligning Components for more information [5].
5	Tab OrderEditor	Set the tab and stacking order of the components in your layout. SeeSetting Tab Order for more information [5].
6	Property Inspector	Set the properties of the components in your layout. It provides a list ofall the properties you can set and displays their current values [5].
7	ObjectBrowser	Display a hierarchical list of the objects in the GUI. See Viewing theObject Hierarchy for more information [5].
8	Run	Save and run the current GUI. See Saving and Running the GUI for Information [5].
9	M-FileEditor	Display, in your default editor, the M-file associated with the GUI. SeeGUI Files: An Overview for more information [5].

4.2 Developing GUI for Signal Processing

Step 1: Developing Graphical User Interface for Signal Processing using MATLAB. Begin to create MATLAB GUI by creating their figurefile name fig-file. By typing 'GUIDE' at the command window, A GUIDE Quick Start window will prompt out [5].

Figure 3: MATLAB Command Window

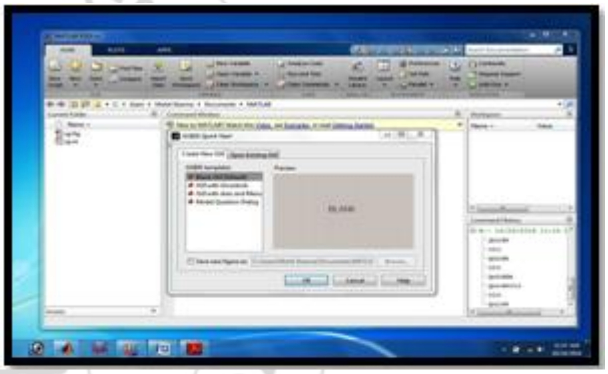

Figure 4: MATLAB GUIDE Quick Start Window

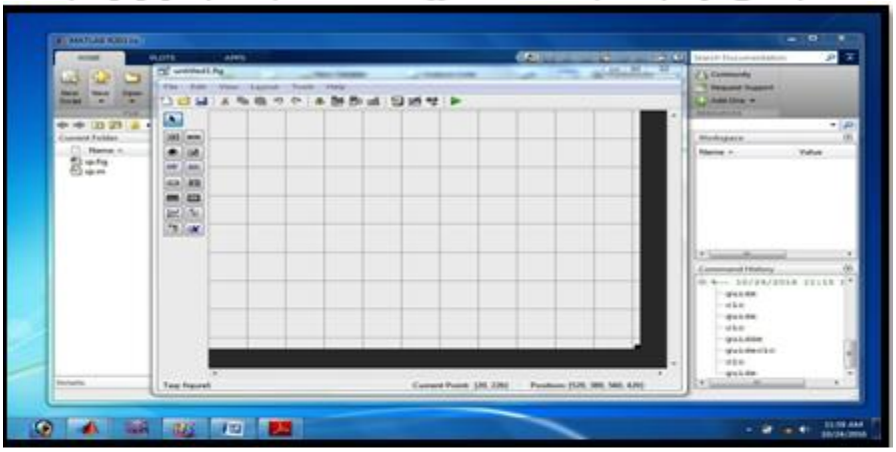

Figure 5: Layout Editor with a blank GUI template display

Step 2: Add function for GUI by dragging component from component palette. There is Push Button, Slider, Radio Button, Check Box, Edit Text, StaticText, Pop-up Menu, List Box, Toggle Button, Axes, Panel, Button Group, and Active X Control [5][6].

For example, if you drag push buttons and axes into the layout area and arrange it and the same arrangements of push buttons and axis is shown in Figure 6.

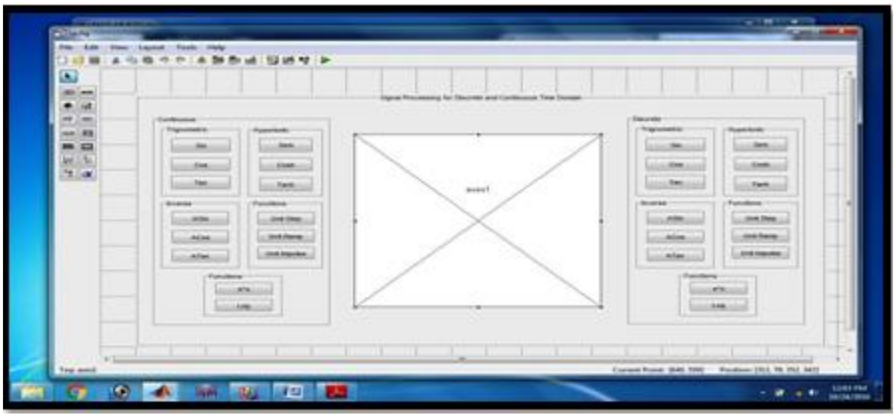

Figure 6: Showing Push Button, Static Text, Axes and Panel at the Lay out area

Step 3: Open the Property Inspector at View > Property Inspector to change the name of pushbutton, Panel and other function name [5].

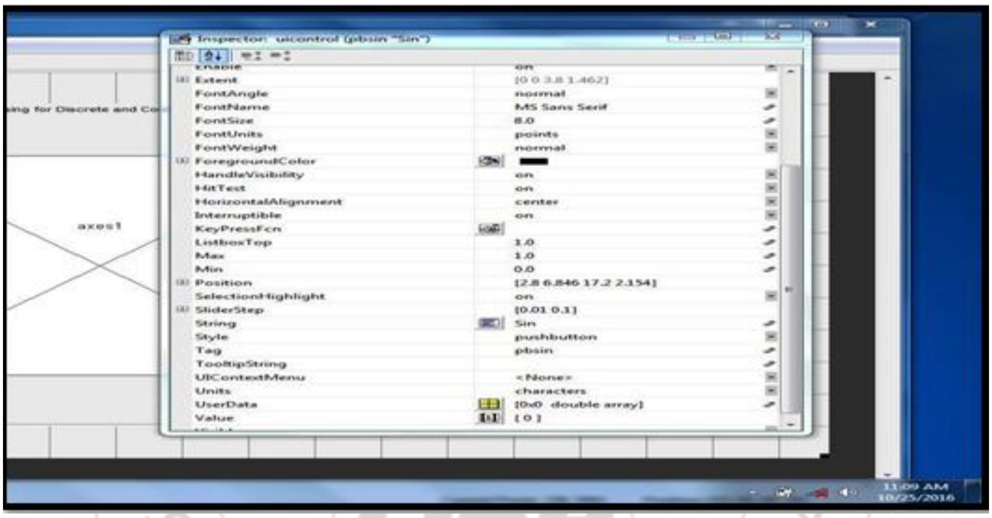

Figure 7: Property Inspector

Step 4: Run the GUI by click the run Button  To run a GUI, select Run from the Tools menu, or click the run button on the toolbar. This display the functioning GUI outside the Layout Editor. GUIDE stores a GUI in two files, which are generated the first time save or run the GUI: FIG-file, with extension .fig, which contains a complete description of the GUI layout and the components of the GUI: push buttons, menus, axes, and so on [5][6].

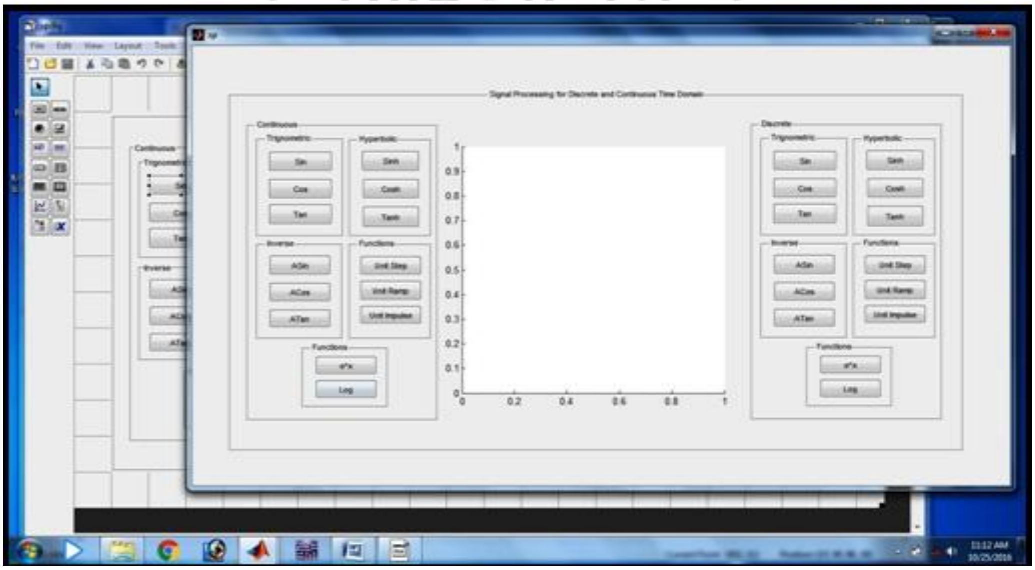

Figure 8: After Click on RUN Button

Step 5: M-file, with extension .m, that contains the code that controls the GUI, including the callbacks for its components. These two files correspond to the tasks of lying out and programming the GUI. When lay out of the GUI in the Layout Editor, The works is stored in the FIG-file. When program the GUI, the work is stored in the M-file [5][6].

Step 6: After laying out your GUI, you can program the GUI M-file using the M-file editor. GUIDE automatically generates this file from your layout the first time you save or run the GUI. The GUI M-file initializes the GUI Contains code to perform tasks before the GUI appears on the screen, such as creating data or graphics contains the callback functions that are executed each time a user clicks a GUI component. Initially, each callback contains just a function definition line. Use the M-file editor to add code that makes the component. To open the M-file, click the M-file Editor icon on the Layout Editor Toolbar. You can view the callback for any of the GUI components by clicking the function icon fx on the toolbar. This displays a list of all the callbacks, as shown in the following figure [5][6].

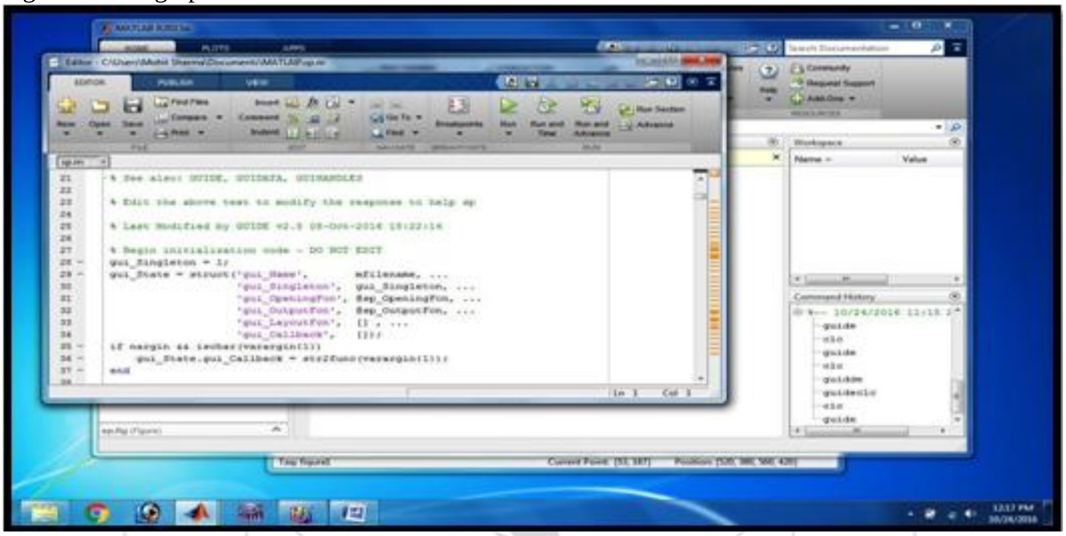

Figure 9: Editor Window with callbacks

5. Results

The designed user interface is shown below in Figure 10.

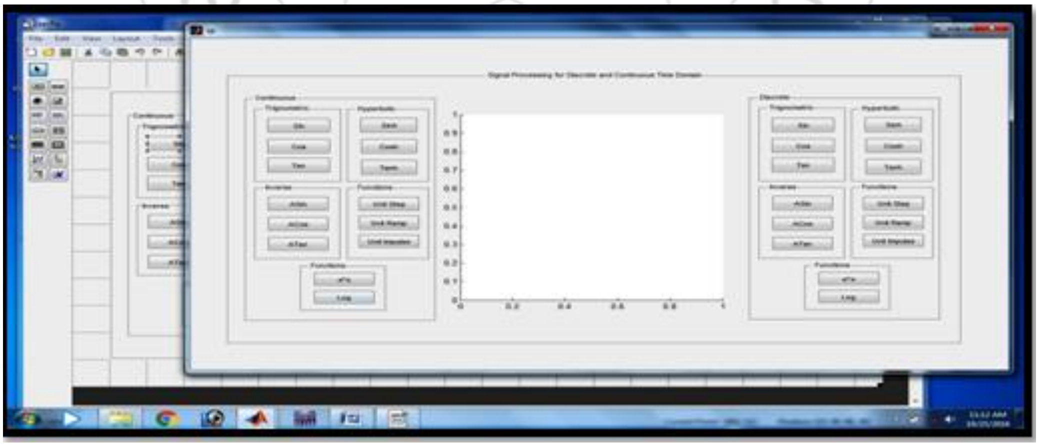

Figure 10: Main GUI Window

To initialize the signals, user can select desired options like Trigonometric, Inverse, Hyperbolic, Unit, Ramp, Impulse, Exponential and Log functions for both continuous and discrete time domain. Some of the results are shown below mentioned figures:

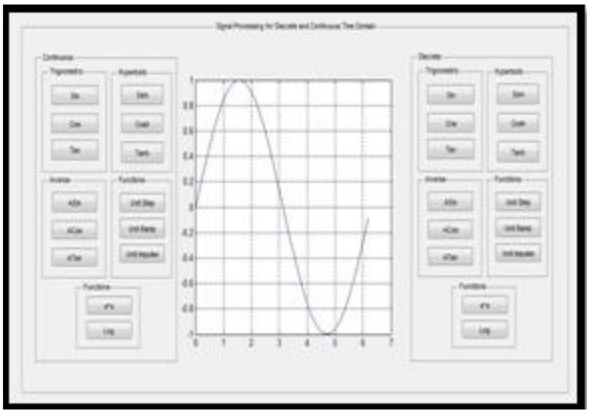

Figure 11: Graph of sin for continuous time domain

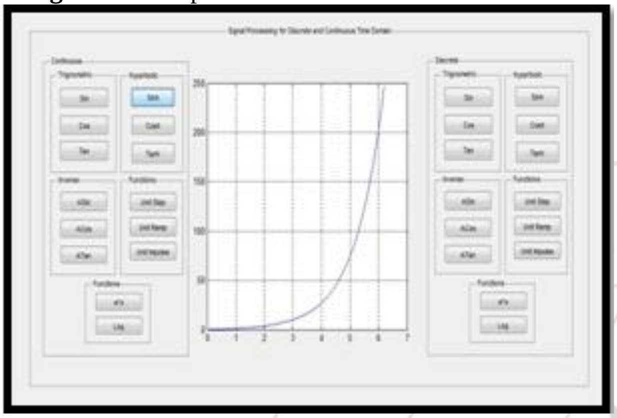

Figure 12: Graph of hyperbolic sin for continuous time domain

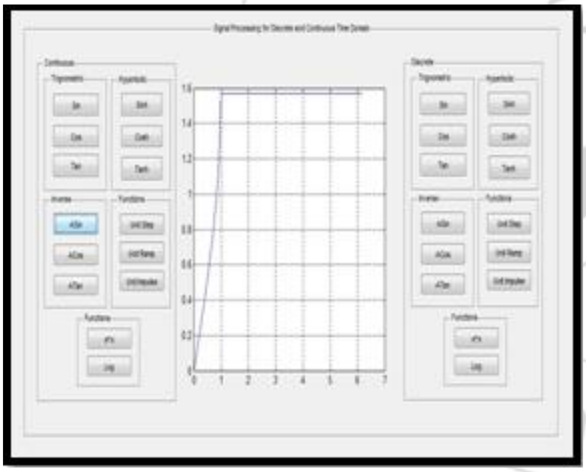

Figure 13: Graph of inverse sin for continuous time domain

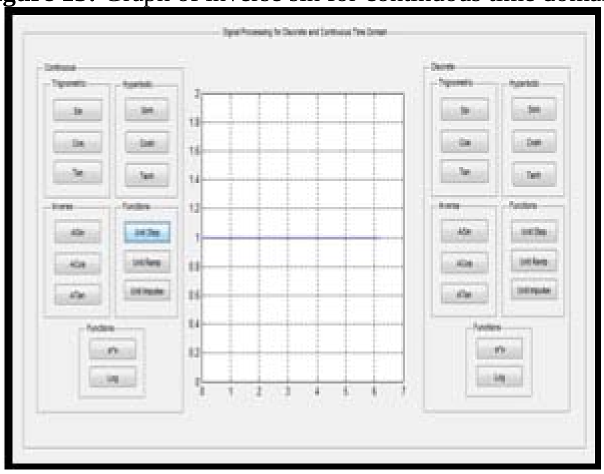

Figure 14: Graph of unit step function for continuous time domain

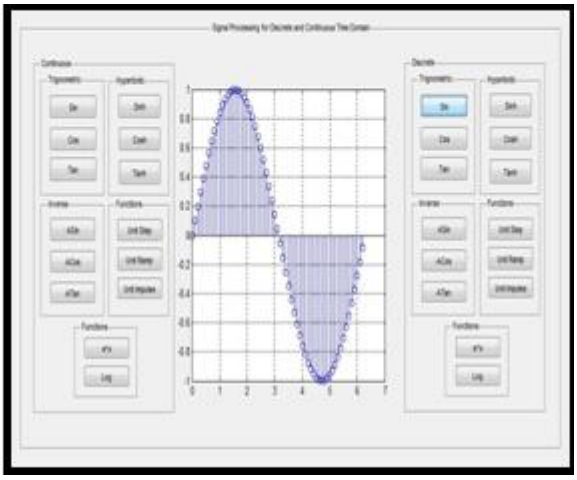

Figure 15: Graph of sin for discrete time domain

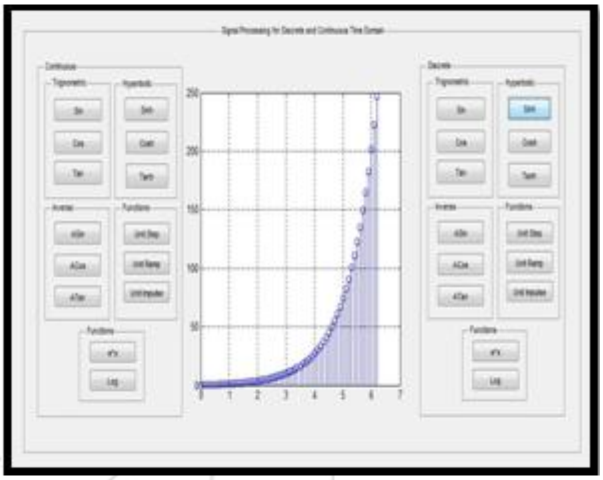

Figure 16: Graph of hyperbolic sin for discrete time domain

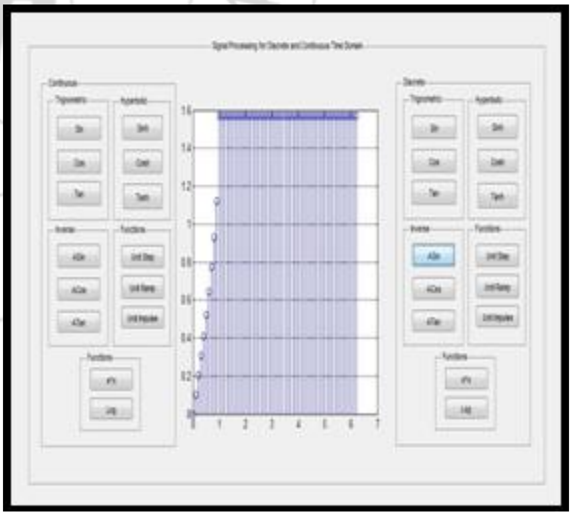

Figure 17: Graph of inverse sin for discrete time domain

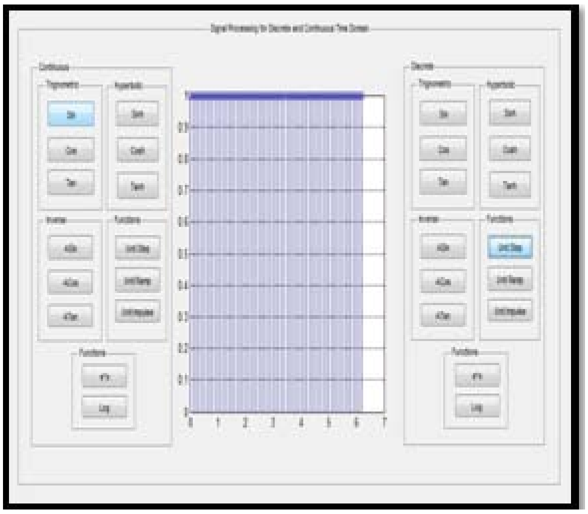

Figure 18: Graph of unit step function for discrete time domain

6. Discussions and Conclusion

In this paper the Matlab based GUI is designed and described for various signals based on signal processing tool. The authors have tried their level best to make the signal processing tool as user friendly as possible. The purpose of the signal processing tool is to bring the various signals functions available in MATLAB tool box under one common platform and to make it easier for the understanding of any user.

References

[1] Refaat Yousef Al Ashi, Ahmed Al Ameri, Prof. Abdulla Ismail Abdulla, Introduction To Graphical User Interface (GUI) Matlab 6.5, College Of Engineering, Electrical Engineering Department, IEEE UAEU Student Branch, UAE University. {4}

[2] Mistry Kaushik P, Prajapati Simpal K, Patel Jenish C, Machhi Devis M, A MATLAB Based GUI for Various Signals Testing and Analysis of Filter (*IJSRD - International Journal for Scientific Research & Development| Vol. 3, Issue 02, 2015 | ISSN (online): 2321-0613).*

[3] Online: https://en.wikipedia.org/wiki/Signal_processing

[4] Stephen J. Chapman, MATLAB Programming for Engineers, 5th Edition, 2008.

[5] Muhammad Lokman Al-Hakim Bin Abu Bakar, Graphical User Interface For Signal Generator, University Malaysia Pahang, Malaysia.

[6] Online: http://www.mathworks.in/videos/creating-a-gui-with-guide-68979.html