

Designing Interactive GUI with MATLAB[®]

Abraham Michelen
Hudson Valley Community College
Troy, New York 12180
micheabr@hvcc.edu

Abstract

In this paper we introduce a tutorial on the use of the *Guide* facility of MATLAB. *Guide* is a development platform composed of several interrelated tools within MATLAB that allows the user to build complex graphical user interfaces that can be used as teaching tools or as automated programs. Several examples related to engineering disciplines are presented.

I. Introduction

MATLAB is a high-level mathematical language for technical computing. In many universities and colleges it is the standard instructional tool for courses in mathematics, engineering, technology, and science. In industry it is widely used as a tool for analysis, development and research. MATLAB integrates computation, visualization and programming in an environment that is easy to learn and use.

From the point of view of its usage, MATLAB is mainly an interactive system. At the command line the user types statements expressed in familiar mathematical notation, and the system responds immediately with the solution of the request. For instance, to create a plot of the function $t*\sin(t)$ for the time interval 0 to 20 in steps of 0.05, we would type

```
t = -10:0.05:10;  
plot(t, sin(t)./t)  
grid on;
```

MATLAB will respond by generating the plot shown in Fig. 1.

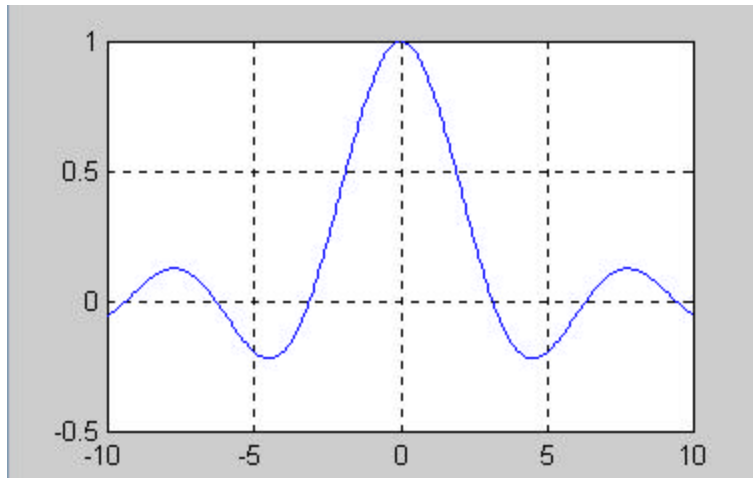


Figure 1: Plot of $\sin(t)/t$

If, however, we need to generate a new plot by changing the interval or any other parameter of the plot we would have to type again the same command lines with different values. In our case this may not be a big problem because we would have to re-type only two lines. However, for more complicated engineering problems the user may find this process quite an ordeal. To minimize this effort, a better approach is to build graphical user interfaces (GUIs) using the *Guide* facility of MATLAB where the user can interact with the GUI by using graphical objects. In this way, to change a parameter in a particular problem and repeat an operation with the new parameter can be accomplished by clicking a push button or by dragging an interface.

Guide is a set of tools designed to build GUIs from within MATLAB. In a matter of minutes the experienced user can create complex graphical tools that can be used to teach the behavior of a system when one or several parameters are changed; even full simulation laboratory exercises can be created without a big effort. The author have developed simulation labs for electronics and telecommunication courses that allow the students to study the behavior of systems without the need to use expensive or complicated equipments. In the next section we introduce an example of a finished problem, and in subsequent sections a tutorial is presented on how to build the GUI using the *Guide* facility of MATLAB.

II. Example: Amplitude Modulation

The following example illustrates the effect on the amplitude-modulated signal when the coefficient of modulation is changed. Mathematically, the amplitude modulation (AM) process can be expressed as

$$y(t) = [1 + n_a x(t)] \cos(2\delta f_c t) \quad (1)$$

where $\cos(2\pi f_c t)$ is the carrier and $x(t)$ is the input signal (data). The parameter n_a is the known as the modulating index or coefficient of modulation. This index is defined as the ratio of the amplitude of the input signal to the amplitude of the carrier. The envelope of the resulting AM signal is $[1 + n_a x(t)]$. Eq. (1) represents the scheme known as double-sideband transmitted carrier (DSBTC). As long as $n_a < 1$, the envelope is an exact replica of the input signal. If, on the other hand, $n_a > 1$, the envelope will cross the time axis and information will be lost because it will not be possible to demodulate the signal at the receiver and obtain the original signal.

With the help of the Guide facility of MATLAB the GUI for this problem was created. As it is illustrated in Figures 2 to 4, this interface consists of a plot area and several push buttons to allow the user to select from several values of the modulation index, and buttons to select or deselect the grid option, and one button to terminate the program. Figure 2 shows the effect of selecting a coefficient of modulation equal to 0.5.

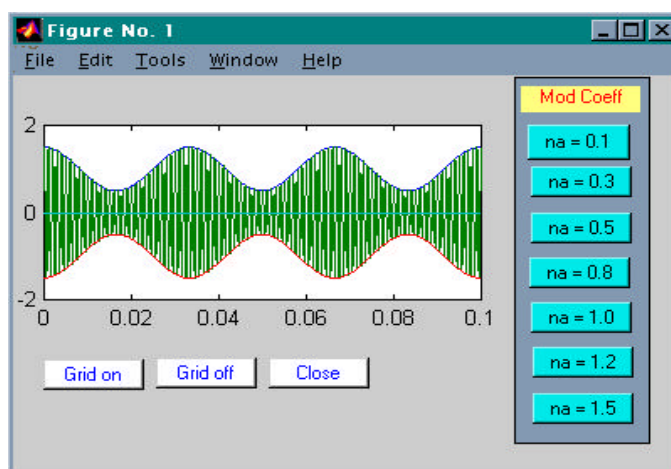


Figure 2: Amplitude modulation with $n_a = 0.5$

By clicking the button for $n_a = 1.0$, Figure 3 will be generated without delay. The student will concentrate in learning the effect of changing the coefficient of modulation rather than in the details of re-typing the statements of the problem.

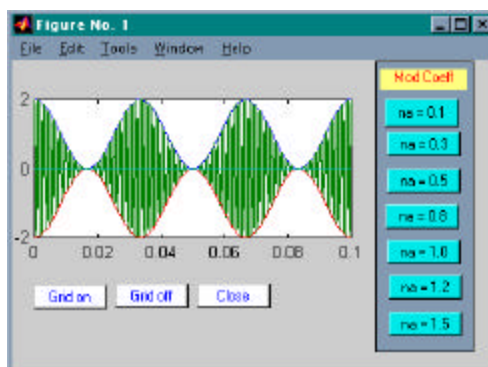


Figure 3: Amplitude modulation with $n_a = 1.0$

Similarly, by clicking the button for $n_a = 1.5$ Fig. 4 will show the effect of aliasing in the AM signal.

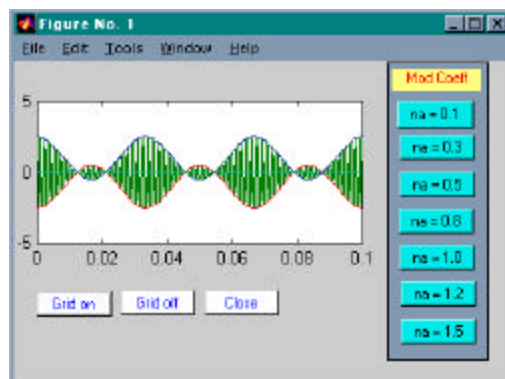


Figure 4: Amplitude modulation with $n_a = 1.5$

III. The *Guide* facility

The *Guide* is a set of five MATLAB tools that allows the user to graphically create GUIs by clicking and dragging objects with the mouse. The *Guide* tools are:

1. The **Control Panel**, among other features, is a container for the other four tools and the **object palette** from where the user selects the objects to be used for the design. There are ten different kinds of objects or controls available:
 - Push buttons
 - Radio buttons
 - Checkboxes
 - Editable text control
 - Sliders
 - Frames
 - Listboxes
 - Pop-up menus
 - Axes (Coordinate System)
2. The **Property Editor**, which is the tool that is used to set the properties of the objects. Properties include the color of the object, its size, the caption that distinguishes the object, the visibility, the initial data assigned to the object, etc.
3. The **Callback Editor** lets the user assign a set of MATLAB commands or functions to an object. When the object is activated (by clicking the object

with the mouse, for instance) the callback commands will be executed. The Callback Editor is also used to make changes to the code assigned to the object in real time, if desired.

4. The **Alignment Tool** allows you to align selected objects using a set of alignment menus. While in design mode, a collection of objects can be aligned vertically, horizontally,
5. The **Menu Editor** is used to add and edit user-created pull-down menu items associated with a project.

IV. A complete Example

To illustrate how to use the *Guide* to create an interactive GUI, we will create a system to plot several different types of graphs. Our design will consist of a coordinate system (an axis) and several push buttons, each one labeled with the type of plot to be generated when the button is clicked with the mouse. The final version of our design will look like Figure 5.

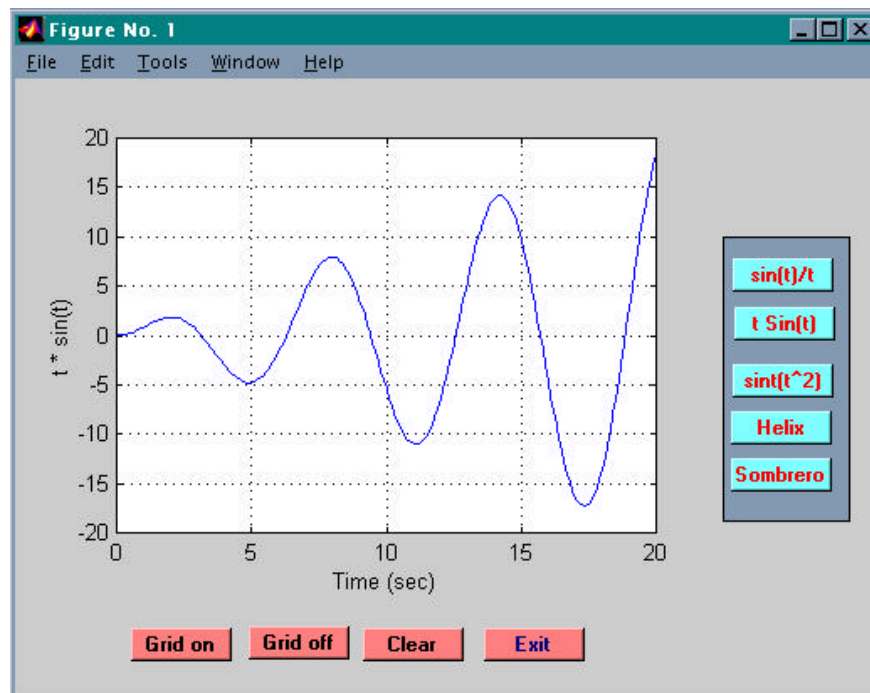


Figure 5: A plotting System

As you can see from the figure, we have the option of plotting five different graphs by clicking the push buttons located at the right-hand side of the axis. Also, we have other buttons to generate or delete the grid of the figure, to clear the figure and to exit the plotting system. Figure 6 shows the result obtained after clicking the button labeled *Sombrero* (the button runs the script that generates a 3D hat).

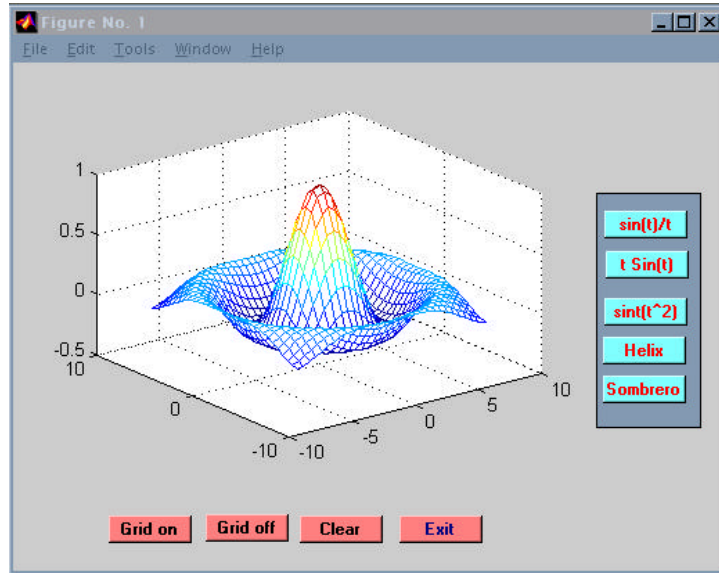


Figure 6: The Sombbrero graph.

Now let's use the *Guide* to build this plotting system from the beginning.

In order to use the *Guide* we must start MATLAB and type *guide* at the command line, as we would when normally working with MATLAB. This command generates a new figure (a window object) and the Guide Control Panel will appear.

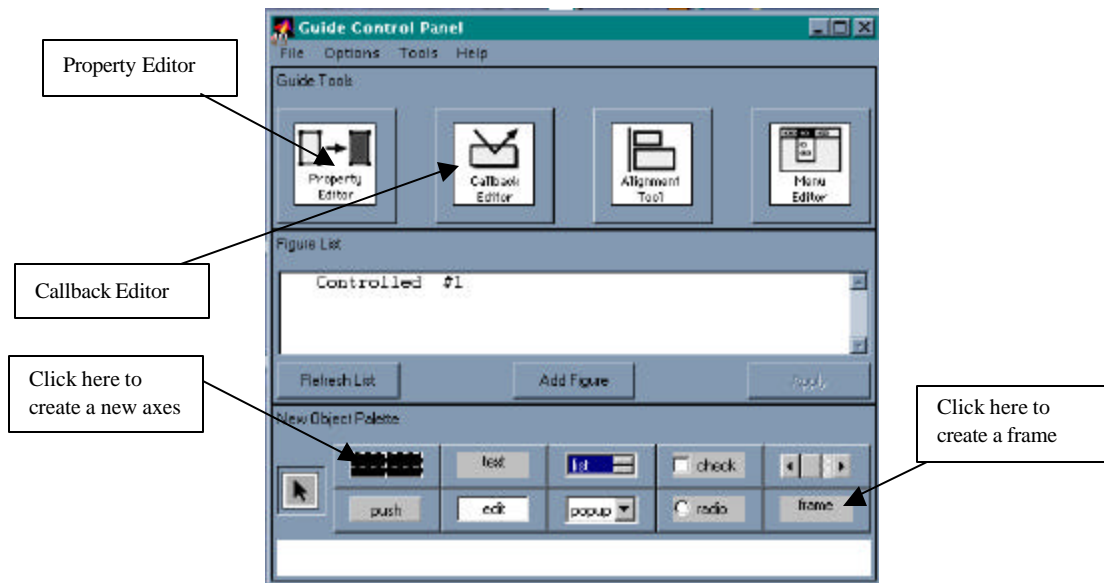


Figure 7: The control Panel

Fig. 7 shows the Control Panel, and Fig. 8 shows the new figure or window to be used for our design.

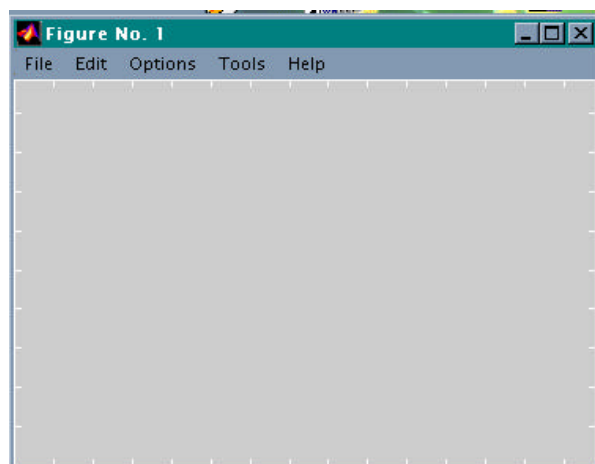


Figure 8: An empty figure or window

The next step in the design is to position and resize an axes object. We do this by clicking the axes button on the Control Panel and then clicking the figure or window. This process will put an axes object on the figure. By dragging the handles of the axes object we can resize and position the axes. Figure 9 shows the result of this procedure.

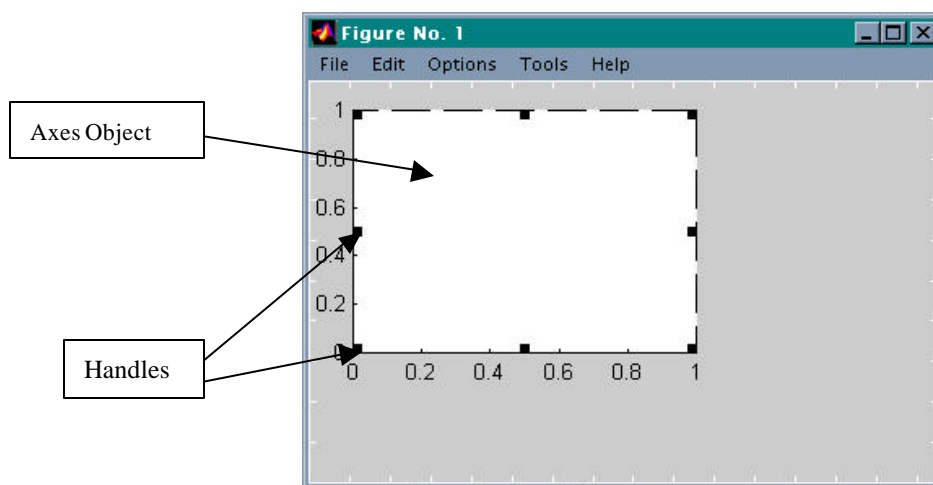


Figure 9: The Axes object

Now, a frame object will be created at the right-hand side of the axes. This frame will serve the purpose of being a container for the graphs push buttons. In the Control Panel (see Figure 7) click the frame object and then click the inside of the window shown in

Fig. 9. Use the handles of the frame to position it in the location you wish. Figure 10 shows the outcome of this action. To position a push-button on the window we proceed in a similar fashion: click the push-button on the Control Panel, click the figure to put the push-button on the window and resize as desired. Figure 11 shows the final position of the push-button that will be used to put a grid on the graph.

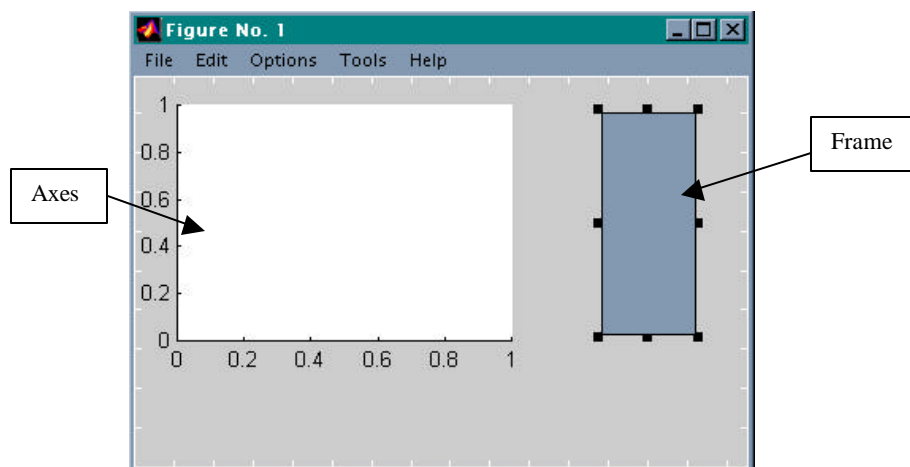


Figure 10: A Frame

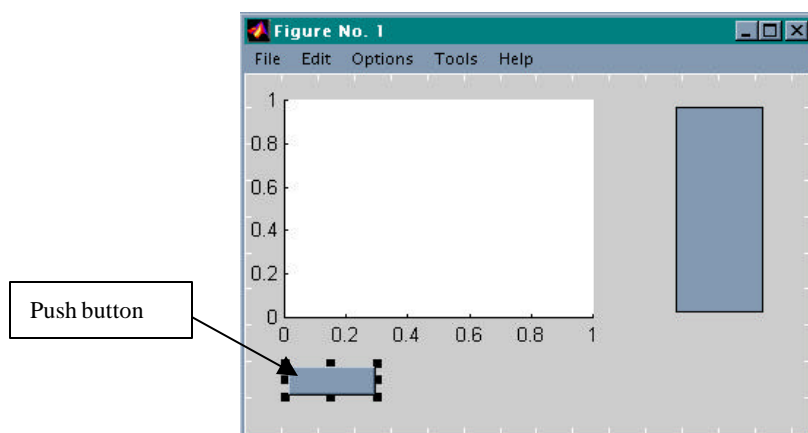


Figure 11: A push-button

In order to change the color of the push-button and to put a caption on top of it, we use the Property Editor of the Control Panel. This is accomplished by following steps:

1. Select the object (the push-button in this case) by clicking it in the figure.
2. Click the icon **Property Editor** on the Control Panel. This will open the Property Editor for the selected object. The Property Editor lists all the properties that can be assigned to the selected object.

- Once the Property Editor is open you can change any of the properties listed. To change color of the selected object, for instance, find the color property, double click it, and a color palette will appear where you can select the desired color. To change or to add a caption, locate the *String* property, double click it and type the caption in the entry box of the Property Editor. Figure 12 shows this process.

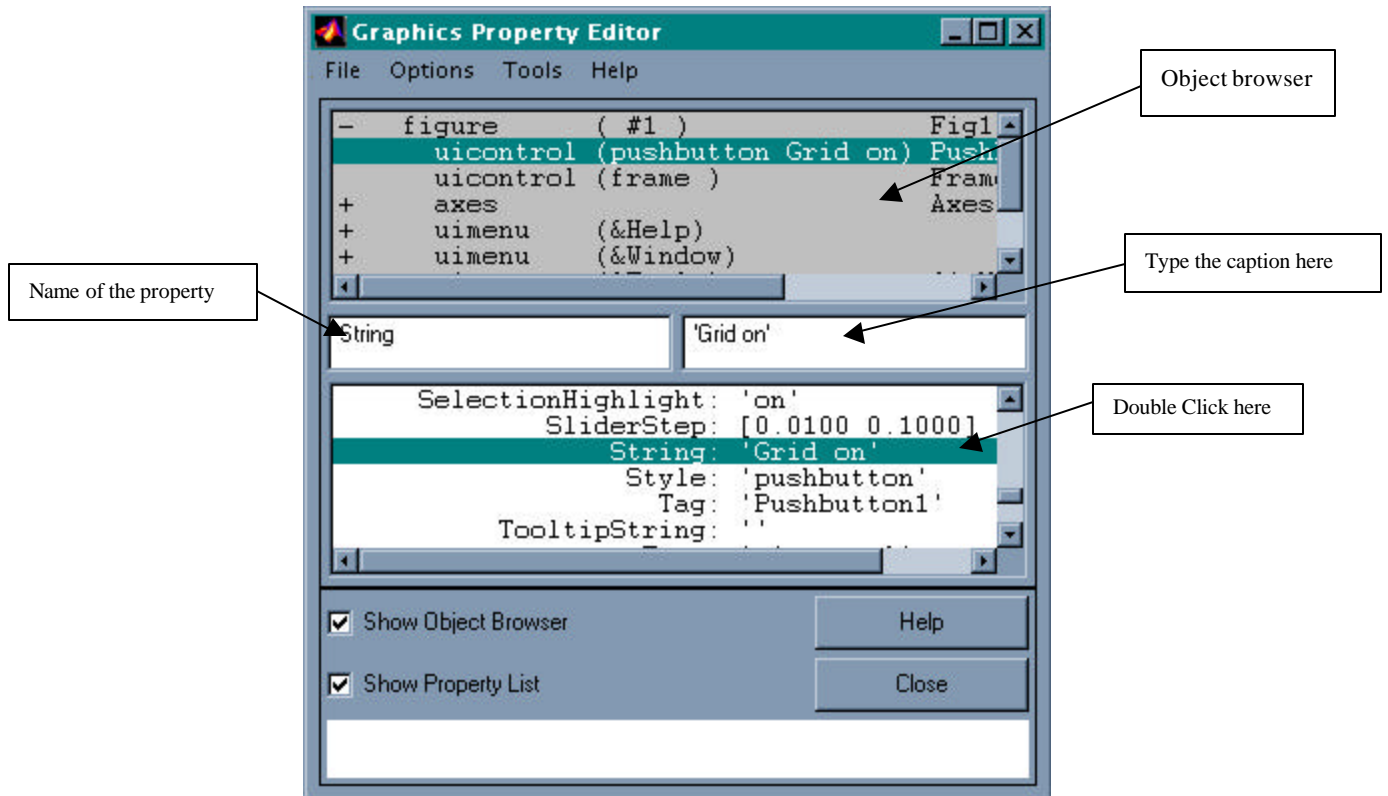


Figure 12: The Property Editor

Figure 13 shows the resulting figure after using the Property Editor.

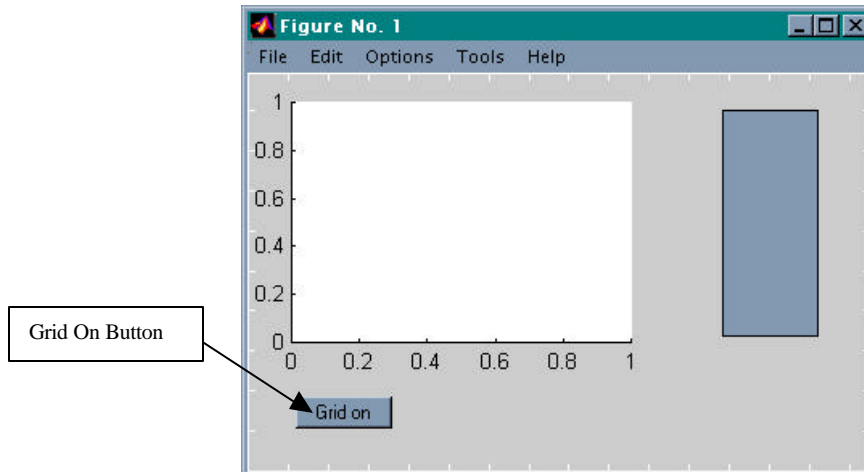


Figure 13: The *Grid On* button

Now that the push-button labeled “*Grid on*” is in the figure, we must attach to it the MATLAB code that will be executed when the *Grid on* button is pressed. To assign commands to an object perform the following steps:

1. Select the object – in this case the *Grid on* button – which you want to attach MATLAB code. This can be accomplished by clicking the name of the object in the Object Browser (see Fig. 12), or by clicking the object in the GUI window (see Fig. 13).
2. In the Control Panel click the Callback Property (Fig. 7). This will open the Callback Editor where you can type the necessary code. In the case at hand the needed code is simply the line: **grid on;** Figure 14 shows the Callback Editor after the code is entered.

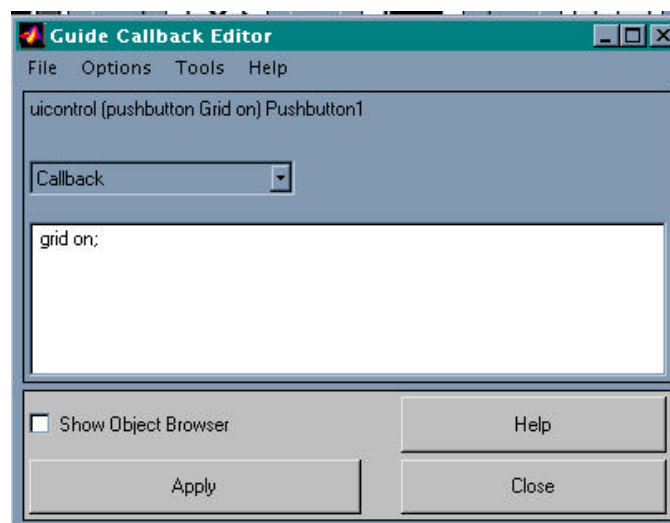


Figure 14: The Callback Editor

To complete the layout presented in Fig. 5, we follow the same principle presented for the creation and activation of the *Grid On* button. Each one of the push buttons in the figure has a different callback MATLAB code attached to it. For example, the code associated to the button labeled **Sin (t)/t** is as follows:

```
t = [-10.001:.05:10.00];
plot(t, sin(t)./t)
grid on;
xlabel('time(sec)');
ylabel('Sin(t)/t');
```

Pressing this button with the mouse (when the system is active) produces the output shown in Fig. 15.

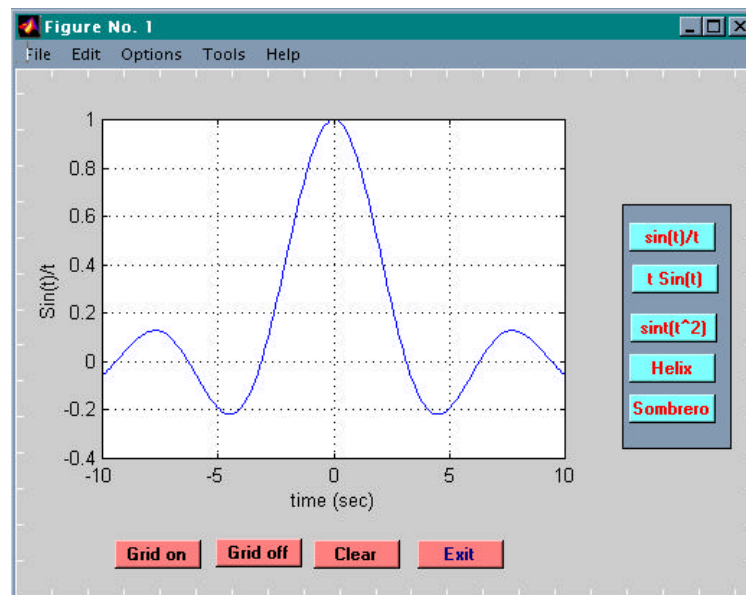


Figure 15: Sin(t)/t

Similarly, the sombrero picture shown in Fig. 6 is easily produced by attaching the following callback code to the corresponding button:

```
x = [-7.5:0.5:7.5];
y=x;
[X,Y] = meshgrid(x,y);
R = sqrt(X.^2,Y.^2+eps);
Z = sin(R./R);
Mesh(X,Y,Z);
grid on
```

V. A Telecommunication Example

A typical application for the *Introduction to Telecommunication* course at Hudson Valley Community College is the concept of the Amplitude Modulation and its several flavors. For this purpose the author developed a GUI that the students use to study AM, Spectrum generation and de-modulation. Figure 16 shows the completed GUI.

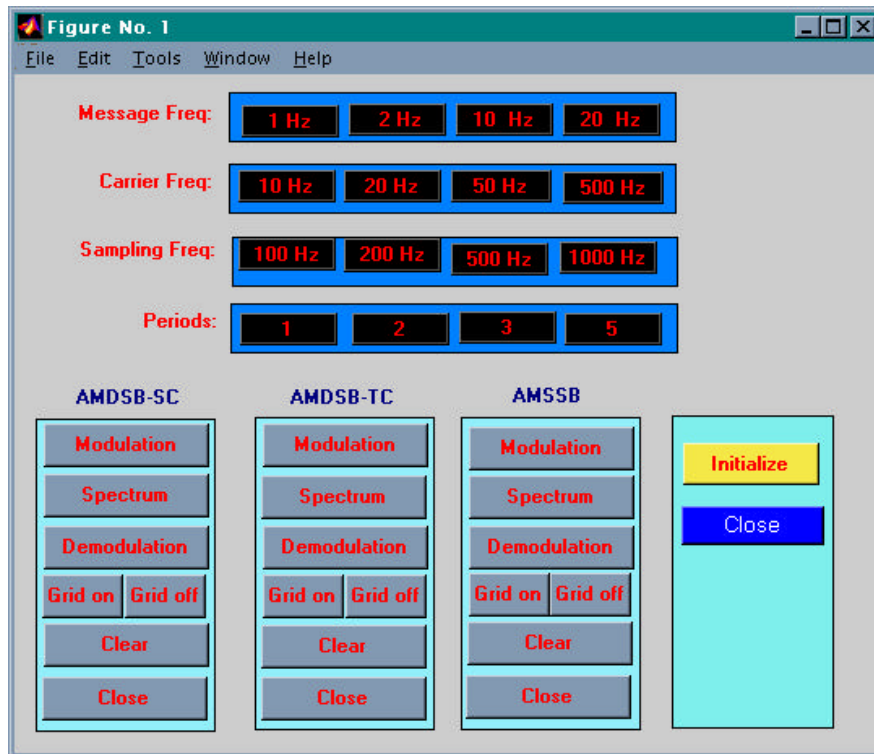


Figure 16: Amplitude Modulation

As you can see, the user selects the message frequency, the carrier frequency, the sampling rate and the number of periods to plot. After pressing the Initialize button, three types of AM can be simulated, namely Double-Side Band Suppressed Carrier, Double-Side Band Transmission Carrier, and Single-Side Band Suppressed Carrier. In the same figure you can generate the corresponding spectrum and the result of the de-modulation. Figures 17, 18, and 19 show the results for a message of frequency of 2 Hz, a carrier frequency of 50 Hz, a sampling rate of 500 Hz, and plot for 2 periods¹².

¹ In order to develop this application you must have the Communication Toolbox of MATLAB.

² If the reader is interested in the source code of this application, send me an e-mail requesting it.

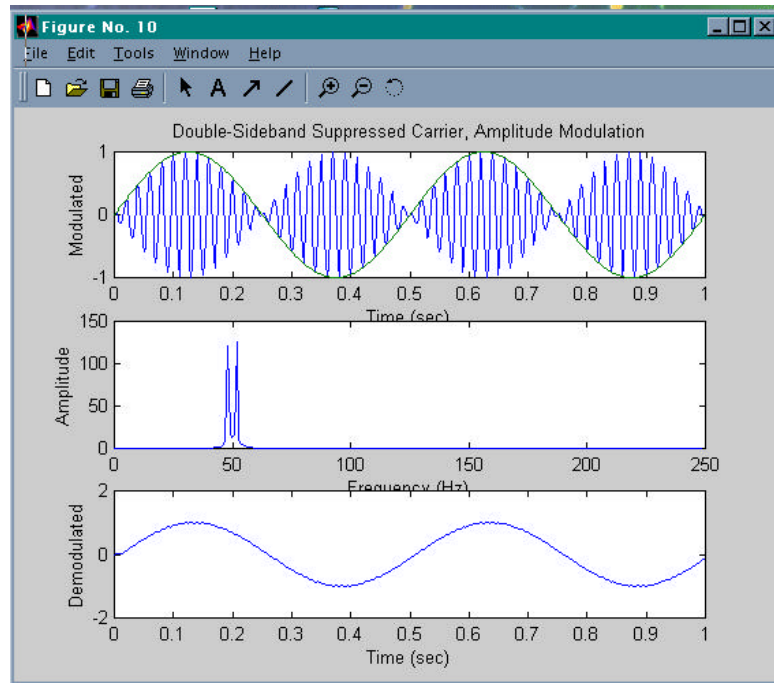


Figure 17: AMDSB-SC AM Modulation

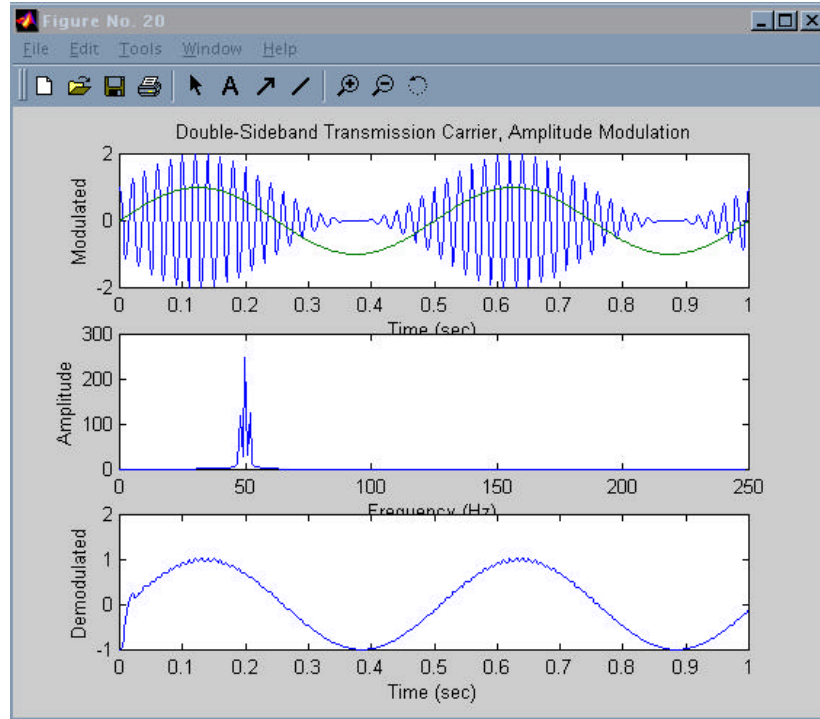


Figure 18: AMDSB-TC AM Modulation

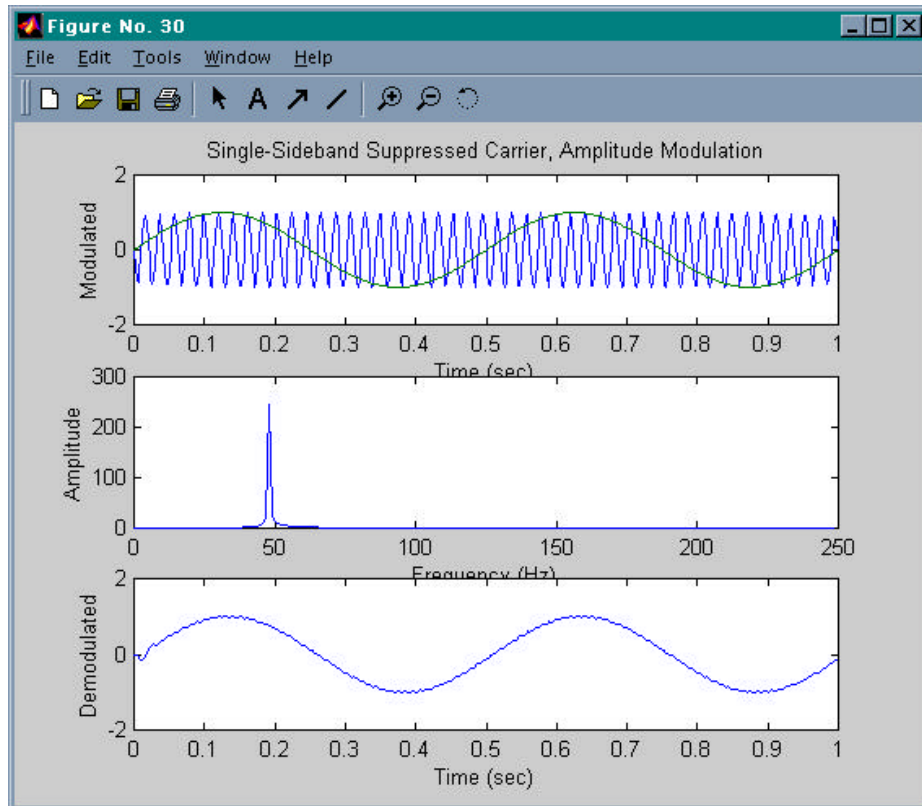


Figure 19: AMSSB-SC AM Modulation

VI. Conclusion

The author has used this MATLAB facility to develop many laboratory exercises for the following courses at Hudson Valley Community College:

- *1930 – Introduction to Telecommunications.*
 - AM Generation
 - AM Detection
 - FM Generation
 - AM Detection
 - FM Transmitter
 - Sinc Filter
 - And others
- *1931 – Data Communication.*
 - Quadrature Amplitude Modulation and Demodulation
 - Pulse Code Modulation (PCM)
 - Phase Shift Keying Modulation (FSK) Techniques
 - Bit-error Rate Simulation
 - Communication System Simulation
 - Eye Pattern and Scatter Plots
 - And others

These exercises do not take too long to develop because all the needed telecommunication functions like filters, modulation and demodulations, bit error rate, DSB-SC, FSK, PCM, etc. are included in a MATLAB package called Communication Toolbox. The toolbox contains hundreds of ready to use communication functions that the author uses to develop the interactive exercises. The GUI is the missing link.

The labs have been used during the last two years in these courses. The author has found that, even though there is no substitution for laboratory exercises using real equipments, these labs have served at least two functions in a small College like HVCC:

1. They are an acceptable method to overcome the prohibited cost of maintaining an updated Telecommunications Lab. By using simulations, the students learn techniques that otherwise would not be available to them.
2. These lab exercises are kept installed in computers available to students in study rooms (learning center rooms). The students can repeat the simulations at any time. This process helps the students to learn and practice the theoretical background of the courses.

Even though, the author does not have a statistical analysis to quantify the degree of usefulness of these tools, we have seen a definite and measurable improvement in the learning process of the students. Test grades are better and the general comprehension of the subject matter has increased in comparison with the students who did not use these exercises.

The Guide facility of MATLAB can be of importance for the development of teaching aids in engineering courses. In this tutorial we presented a step-by-step procedure for the development of GUIs for this purpose. Several examples – some of them with the source code – were presented to demonstrate the usefulness of this facility. We recommend the Guide package for the development of teaching tools in an academic environment.

VII. Bibliography

1. The MathWorks, Inc., “MATLAB, The Language of Technical Computing: Building GUIs with MATLAB” (1999).
2. The MathWorks, Inc., “Simulink: Communications Toolbox” by Weizheng Wang, 1999.
3. William Stallings, “Data and Computer Communications”, Prentice Hall. (1998)
4. Warren Hioki, “Telecommunications”, Prentice Hall. (1999)

Abraham Michelen

Abraham Michelen is currently an Electrical Engineering Technology and Telecommunication Technology professor at Hudson Valley Community College in Troy, New York. He received his Ph. D. in 2000 from Rensselaer Polytechnic Institute in Troy, New York. Abraham main interests are in Petri Nets, Computer Simulations, and Computer Programming. In addition to his teaching career, Abraham is a Development Engineer with the Internet Company GlobalSpec, Inc. (www.globalspec.com) in Troy, New York.