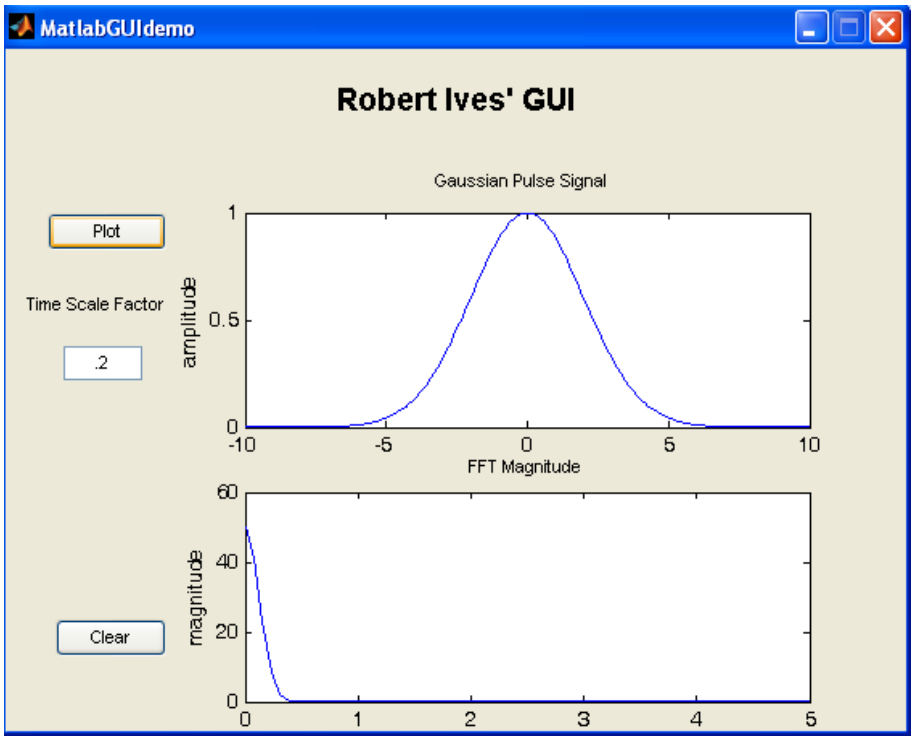
EE432: Digital Signal Processing
MATLAB GUI Tutorial

This tutorial will lead you through the creation of a simple MATLAB GUI. The GUI will allow a user to create and clear a plot of a Gaussian pulse, and to display the magnitude of its FFT. A picture of what your end product should look like is shown to the right. Each GUI is composed of two parts: a .fig file (which determines what the GUI looks like) and an *m*-file (which determines what happens in the GUI). The process starts by creating the layout of the GUI using the .fig file; the functionality is added later in the *m*-file.



1. To begin, go to the course website and download two images, *DefaultPlot.bmp* and *DefaultFFT.bmp*. They will be used in the GUI and you'll be instructed how to use them.

2. Start MATLAB and at the command line, type
   >> guide

   This starts the MATLAB Graphical User Interface Development Environment.

3. Select the type of GUI. For this tutorial, choose "Blank GUI (Default)". This will create a blank GUI (a working .fig file).

4. On the left side of the palette, there are buttons which represent the different types of objects that can be added to the GUI. To begin, drag one of the "Static Text" objects over to the top of the grid. *Static Text* is text that never changes. You will use this for the title of your GUI. Place this object at the top of the grid.

   a. Double-click on this Static Text object, and you will see the properties associated with it (the "Property Inspector" window will appear). The properties associated with this object include settings such as its location on the grid, font type, font size, and other such information.

   b. Using the Property Inspector window, change the font size to 14 and make it bold ("font weight").

   c. Change the "String" property to "xxxx GUI DEMO" instead of "Static Text", where xxx is your group's names…this is the title of your GUI. The font size is too large for the object you originally created, so drag the corner of the *Static Text* object to make it bigger.

4. Drag two “Axes” boxes onto the grid, and enlarge it to what seems a reasonable size. Note the name of this object: “Axes1”. When you add more axes, they would be named “Axes2”, “Axes3”, etc.
5. Drag a “Static Text” box over each axis, and change the “String” property to “Gaussian Pulse Signal” for the upper box, and “FFT Magnitude” for the lower one.

6. Drag a “Static Text” box on the left side, and change the “String” property to “Time Scale Factor.” This just gives a name to the “Edit Text” box you will create in the next step.

7. Drag an “Edit Text” box onto the grid right below the “Time Scale Factor” box you just created. “Edit Text” is different than “Static Text” because it can change. We will use this so that the user can enter a time compression factor to add to widen the Gaussian pulse or make it narrower.

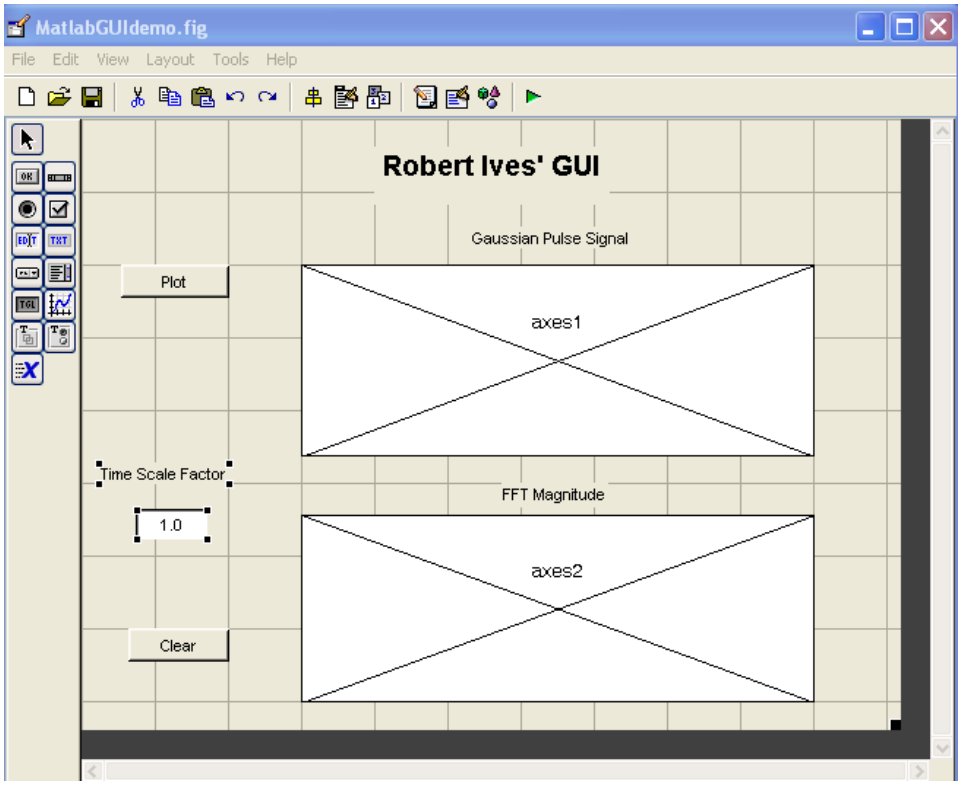
   a. Change its “String” property to “1.0”. This will be the default time scale factor (that is, no time compression).

6. Drag two “Push Buttons” to the grid, one on top of the other. One will be named “pushbutton1” and the other “pushbutton2”.

   a. Double-click on the top push button and change the “String” to “Plot”.
   b. Double-click on the bottom push button and change its “String” to “Clear”.

   c. Elongate the boxes if needed so the text fits.

7. Save the GUI figure (which is currently called “untitled.fig”). Save it as “MatlabGUIdemo.fig”. Your end product should look something like the figure to the right. At this point (after you save the .fig file), MATLAB will create an *m*-file that specifies how the GUI should function: that is, what each button does, what the axes is supposed to show, etc. Now we edit the *m*-file to complete the GUI.



8. Most of the code is background code MATLAB needs to run the GUI, and should not be altered. It consists of a number of MATLAB functions specific to this GUI, and in order to make the GUI work as advertised, we need to add some lines to some of the functions.

9. The function called "GUI_demo_OpeningFcn" (at approximately line 50 of the m-file) controls what happens when the GUI is started. For example, we could have the GUI display a default image on *axes1*. For this GUI, let's initially display an image I made up called "Default-image.bmp". Add the following lines at the end of this function:

```
axes(handles.axes1);
imshow('DefaultPlot.bmp');
axes(handles.axes2);
imshow('DefaultFFT.bmp');
```

10. When the user pushes the "Plot" button, the time scale factor is read, and then the Gaussian pulse should display on axes1 and the FFT magnitude should appear on axes2. To accomplish this, add the following lines at the end of the *pushbutton1_callback* function:

```
fs=10;      % sample frequency
N=128;      % number of samples, a power of 2
t=-10:1/fs:10;  % time vector
scale = get(handles.edit1,'String');  % get the string that is currently in the
edit1 text box
scale=str2num(scale);                   % convert this to a numeric value
x=exp(-pi*(scale*t).^2);                % create the Gaussian Pulse signal
axes(handles.axes1);                    % axes1 is now the current axis
plot(t,x)                               % plot the Gaussian Pulse
xlabel('time (sec)')
ylabel('amplitude')

d=abs(fft(x));                          % compute the magnitude of the FFT
d=d(1:N/2+1);                           % only keep the 1st N/2 +1 points of the
FFT
f=(0:N/2)*fs/N;                         % create the frequency vector
axes(handles.axes2);                    % axes2 is now the current axis
plot(f,d);                              % plot the FFT magnitude
xlabel('frequency (Hz)')
ylabel('magnitude')
```

11. When the user pushes the "Clear" button, both plots should go to all zeros. To accomplish this, add the following lines at the end of the *pushbutton2_callback* function:

```
N=128;
fs=10;    % sample frequency
t=-10:1/fs:10;
x=zeros(size(t));
axes(handles.axes1);
plot(t,x)

xlabel('time (sec)')
ylabel('amplitude')

f=(0:N/2)*fs/N;
d=zeros(size(f));
axes(handles.axes2);
plot(f,d);
xlabel('frequency (Hz)')
ylabel('magnitude')
```

12. After the m-file is saved, you can run the GUI at the command line with
    >> MatlabGUIdemo

13. When the GUI is up and running, see what the plots look like when you change the time scale factor to 0.2, and then change it to 2.

14. This concludes this tutorial. You can change the GUI *m*-file as desired, and as long as you save the *m*-file, the changes will be incorporated. You can add things to the .fig file (such as more push buttons, or slider bars, etc.), and when you save the .fig file, the *m*-file is updated to add functions for these. If you add additional buttons, etc., you need to add code to the *m*-file to give the additional blocks some functionality. You can even have real-time signals in the GUI, like the continuous display of the microphone's signal, or live video.

15. One thing you probably don't want to do is to rename the .fig file or the .m file after you initially save it; this will cause problems, as various properties of each object in the GUI refer to the original name of the GUI. For example, on MatlabGUIdemo.fig, double-click on the "Display original image" pushbutton. Notice that the call-back functions for this pushbutton is "MatlabGUIdemo (…". Depending on the complexity of the GUI, it can be challenging to rename a GUI and have it work as advertised, since a good number of the properties of each object may need to be changed, NOT ONLY in the .fig file, but ALSO in the *m*-file.

16. Finally, if you want to come back tomorrow to change the look of the GUI, you can edit the .fig file using the command: >> guide MatlabGUIdemo.fig.