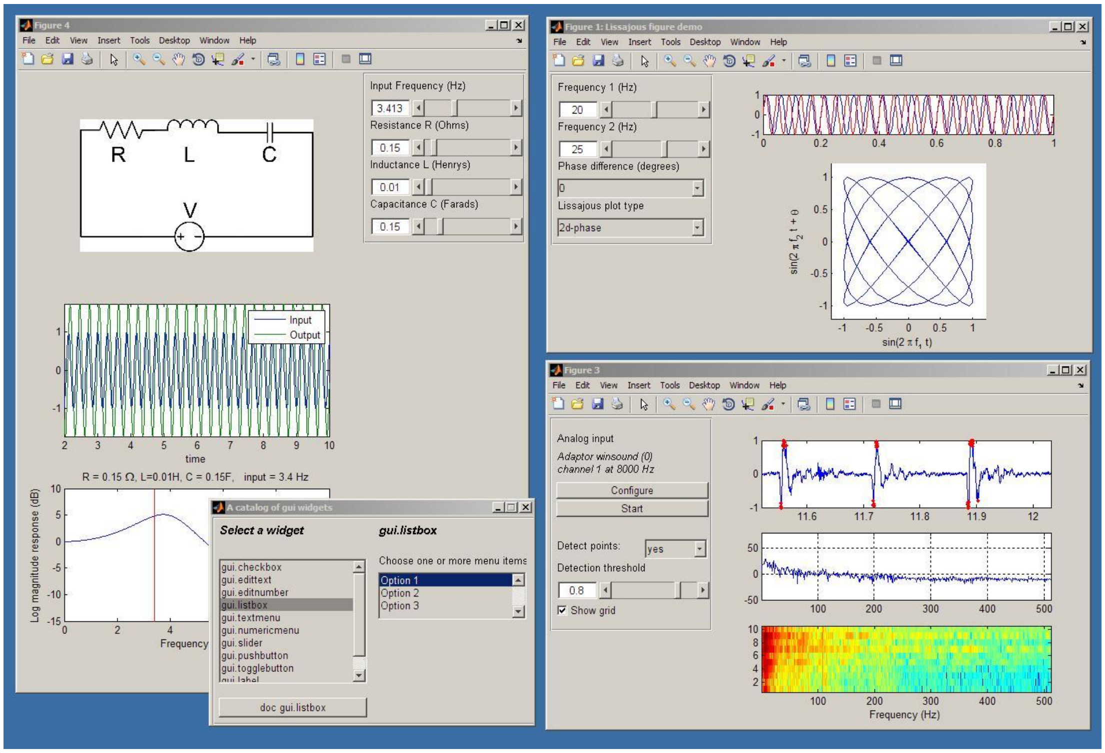
CONTENTS



Input Frequency (Hz) 3.413

Resistance R (Ohms) 0.15

Inductance L (Henrys) 0.01

Capacitance C (Farads) 0.15

R = 0.15 Ω, L=0.01H, C = 0.15F, input = 3.4 Hz

Frequency 1 (Hz) 20

Frequency 2 (Hz) 25

Phase difference (degrees) 0

Lissajous plot type 2d-phase

Select a widget

gui.listbox

Choose one or more menu items

Option 1
Option 2
Option 3

Analog input

Adaptor winsound (0)
channel 1 at 8000 Hz

Detect points: yes

Detection threshold 0.8

☑ Show grid

WHAT IS EASYGUI?

EasyGUI is a set of tools to make GUIs quickly and easily in MATLAB. It provides:

- A set of "widgets" (UI controls) with a simple programming interface
- Automatic layout of the widgets
- An intuitive way to combine a GUI with MATLAB plotting and visualization

REQUIREMENTS

EasyGUI uses MATLAB Object-Oriented Programming, so it requires at least R2008a. It does not need any specialized toolboxes. I have tested it on Windows XP with R2008a, R2008b and R2009a.

ADDING EASYGUI TO YOUR PATH

When you unzip EasyGUI_v1_0.zip, it will create a folder named `EasyGUI` with subfolders `+gui` and `examples`. You need to add the EasyGUI folder to your path. Here is one way to do it:

1) In the MATLAB window, select File -> Set Path …
2) Click on "Add Folder" and find the EasyGUI folder
3) Click on "Save" in the "Set Path" window

PACKAGES AND CLASSES

All the EasyGUI classes are in a package named "gui" (see here for an introduction to packages in MATLAB). Consequently, all the EasyGUI class names have to be prefixed by `gui.`, like this:

```
>> myMenu = gui.textmenu; % create a textmenu widget
>> help gui.version
   GUI.VERSION
     Returns a struct containing the version of EASYGUI.
```

The dot-notation is also used in MATLAB to indicate the property of an object. For example:

```
>> mySlider = gui.slider;  % 'gui' is a package, 'slider' is a class in the package
>> mySlider.Value          % 'mySlider' is an object', 'Value' is its property
ans =
    0.5000
>> mySlider.Value = 0.1;
```

It is important not to confuse these different uses. Also, do not create a variable named `gui`, as the variable name will take precedence over the package name and can lead to confusing errors.

CREATING GUIS

HOW DO I GET STARTED?

The best way to get started is to create a GUI from the command line.

First, create the figure object (also called a *container*) that will automatically position the widgets.

```
myGui = gui.autogui;
```

Now create a widget (it is added to `myGui` by default) .

```
num = gui.slider('Number of points:', [5 50]);
myGui.waitForInput()
num.Value
```

Subsequent widgets are automatically positioned:

```
color = gui.textmenu('Choose a color', {'red', 'green', 'blue'});
showgrid = gui.checkbox('Show grid?');
```

You can also set the values of the widgets

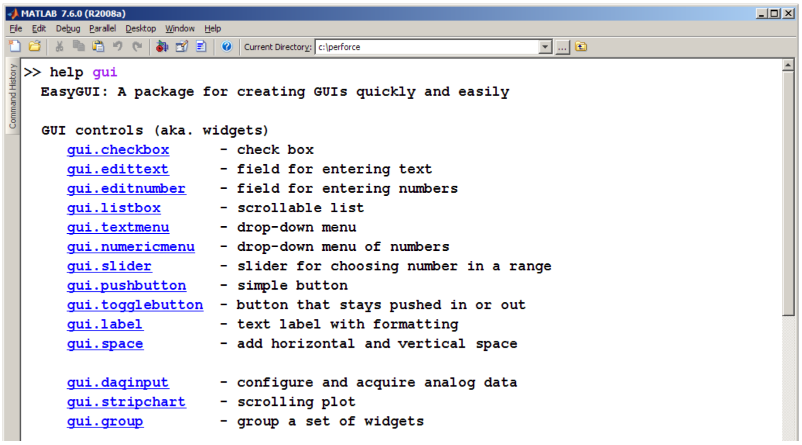
```
num.Value = 20;
color.Value = 'green';
showgrid.Value = true;
```

Get user input and use the widget values to do some plotting:

```
  myGui.waitForInput();
  y = rand(1, round(num.Value));
  plot(y, 'color', color.Value(1)); % use first character of color name
  if showgrid.Value, grid on; end
```

WHAT KINDS OF WIDGETS ARE AVAILABLE?

To see a list of available widgets, type `help gui`. The widget names (e.g., gui.checkbox, gui.edittext) are hyperlinked to help text.

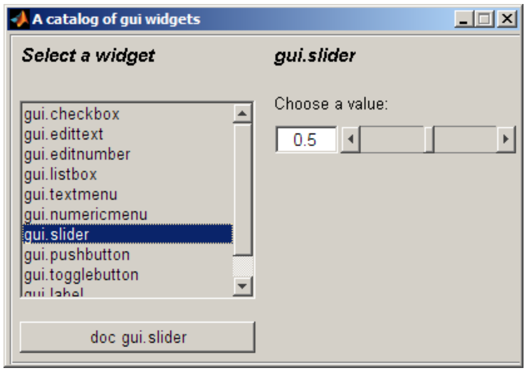

MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
Current Directory: c:\perforce
Command History

```
>> help gui
  EasyGUI: A package for creating GUIs quickly and easily

  GUI controls (aka. widgets)
     gui.checkbox      - check box
     gui.edittext      - field for entering text
     gui.editnumber    - field for entering numbers
     gui.listbox       - scrollable list
     gui.textmenu      - drop-down menu
     gui.numericmenu   - drop-down menu of numbers
     gui.slider        - slider for choosing number in a range
     gui.pushbutton    - simple button
     gui.togglebutton  - button that stays pushed in or out
     gui.label         - text label with formatting
     gui.space         - add horizontal and vertical space

     gui.daqinput      - configure and acquire analog data
     gui.stripchart    - scrolling plot
     gui.group         - group a set of widgets
```

Another way to explore the widgets is to run `catalog` in EasyGUI/examples. This opens up the following GUI. When you select one of the widgets, that widget is created and shown on the right.



gui.checkbox
gui.edittext
gui.editnumber
gui.listbox
gui.textmenu
gui.numericmenu
gui.slider
gui.pushbutton
gui.togglebutton
gui.label

Choose a value: 0.5

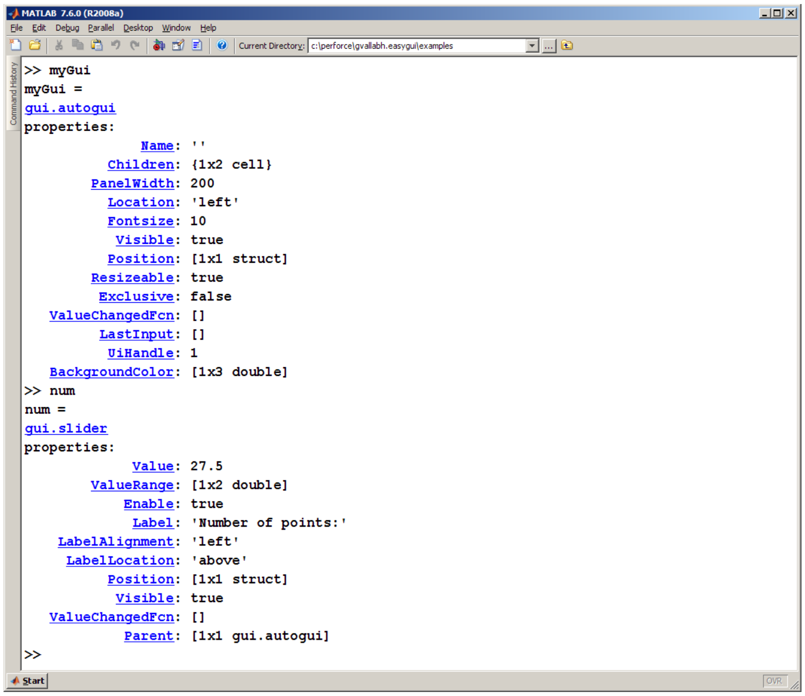
A widget can be created from the command line simply by typing its class name (its gets created with some reasonable defaults).
For example:

```
>> myGui = gui.autogui;
>> w1 = gui.slider;
>> w2 = gui.textmenu;
>> w3 = gui.togglebutton;
```

WHERE DO I GET MORE HELP WITH USING EASYGUI?

There are several ways to get more information.

1) Type the name of the EasyGUI object (container or widget) at the command line. You will get a list of properties with hyperlinks to help text, as shown below. Explore what happens when you change the various properties.


MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
Current Directory: c:\perforce\gvallabh.easygui\examples
Command History
Start
OVR

```
>> myGui
myGui =
gui.autogui
properties:
               Name: ''
           Children: {1x2 cell}
         PanelWidth: 200
           Location: 'left'
           Fontsize: 10
            Visible: true
           Position: [1x1 struct]
         Resizeable: true
          Exclusive: false
    ValueChangedFcn: []
          LastInput: []
           UiHandle: 1
    BackgroundColor: [1x3 double]
>> num
num =
gui.slider
properties:
              Value: 27.5
         ValueRange: [1x2 double]
             Enable: true
              Label: 'Number of points:'
     LabelAlignment: 'left'
      LabelLocation: 'above'
           Position: [1x1 struct]
            Visible: true
    ValueChangedFcn: []
             Parent: [1x1 gui.autogui]
>>
```

2) Type `help gui` at the command line, and click on the hyperlinks for help text. You can also get help on a single class, or on a method or property in the class:

```
help gui.slider                  % help on the gui.slider class
help gui.slider.Value            % help on the 'Value' property
help gui.autogui.waitForInput    % help on the 'waitForInput' method
```

3) Look through the examples in the folder EasyGUI/examples. Start with `autodemo` for a simple gui, then look at `autodemo2` and `autodemo3` for more sophisticated versions of that gui. For an index of examples, type `help EasyGUI/examples`

HOW DOES THE AUTOMATIC POSITIONING WORK?

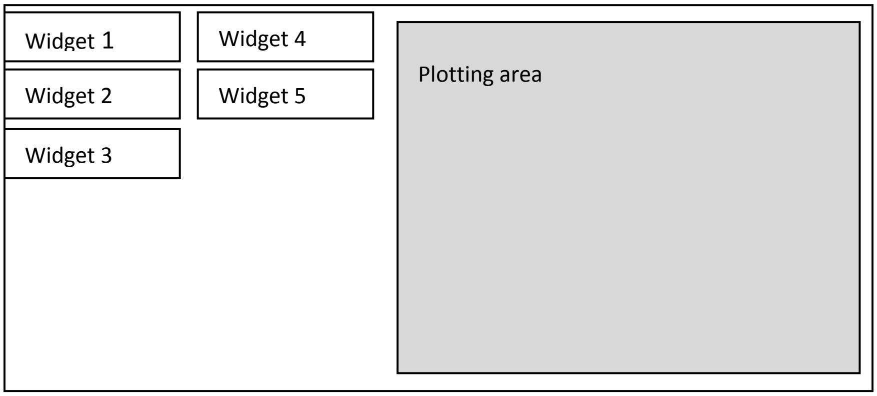
The `gui.autogui` class automatically positions all the widgets in it. This positioning follows two very simple rules.

1) A new widget is positioned *below* the most recent widget.
2) To start a new "column" of widgets, invoke the addPanel() method.

For example, consider the following sequence of commands:

```
myGui = gui.autogui;
widget1 = gui.slider;
widget2 = gui.textmenu;
widget3 = gui.pushbutton;
myGui.addPanel;     % start a new column of widgets
widget4 = gui.slider;
widget5 = gui.numericmenu;
```

The above sequence will produce the following layout:



Sometimes you may not be satisfied with the default vertical and horizontal spacing. The **vertical spacing** may be adjusted by using the `gui.space` widget – this simply inserts blank space.

```
myGui = gui.autogui;
widget1 = gui.slider;
spacer1 = gui.space; spacer1.Position.height = 30;
widget2 = gui.textmenu;
```

The **horizontal spacing** (the space between adjacent "columns") can be tweaked with the `PanelWidth` property, which sets the width of the current panel. For example:

```
myGui = gui.autogui;
myGui.PanelWidth = 150;
widget1 = gui.slider; % will have a width of 150
widget2 = gui.textmenu; % will have a width of 150
myGui.PanelWidth = 200; % widens the current panel but not the widgets in it
myGui.addPanel;
widget3 = gui.slider;
```

WHAT IF I WANT TO POSITION WIDGETS MANUALLY?

To position widgets manually, use the `gui.manualgui` class (see `examples\manualdemo.m`).

QUESTIONS, COMMENTS, OR SUGGESTIONS?

Send an email to Gautam Vallabha (gautam.vallabha@mathworks.com).