APPLICATIONS OF MATLAB IN ENGINEERING

Yan-Fu Kuo Dept. of Bio-industrial Mechatronics Engineering National Taiwan University

Today:

• Graphical user interface



Fall 2015

- User interface (UI) is a method of interaction between a person and a computer
- 1970s





- User interface (UI) is a method of interaction between a person and a computer
- 1980s



Current date is Tue 1-01-1980									
Enter neu	Enter new date:								
Current 1	Current time is 7:48:27.13								
Enter new time:									
The IBM Personal Computer DOS									
Version 1.10 (C)Copyright IBM Corp 1981, 1982									
A>dir∕w									
Command	COM	FORMAT	COM	CHKDSK	COM	SYS	COM	DISKCOPY	COM
DISKCOMP	COM	COMP	COM	EXE2B IN	EXE	MODE	COM	EDLIN	COM
DEBUG	COM	LINK	EXE	BASIC	COM	BASICA	COM	ART	BAS
SAMPLES	BAS	MORTGAGE	BAS	COLORBAR	BAS	CALENDAR	BAS	MUSIC	BAS
DONKEY	BAS	CIRCLE	BAS	PIECHART	BAS	SPACE	BAS	BALL	BAS
COMM	BAS								
26 File(s)									
A>dir command.com									
Command	COM	4959	5-07-82	12:00p					
	1 File(s	3)		-					
A>									

- User interface (UI) is a method of interaction between a person and a computer
- 1990s





 User interface (UI) is a method of interaction between a person and a computer

• 2000s





- User interface (UI) is a method of interaction between a person and a computer
- Future!





MATLAB GUI Programs



Start A GUI Program

- Set your "current folder" where you want to store the GUI program
- Use GUIDE (graphical user interface design environment) to create a MATLAB GUI interactively by typing in guide command window

GUIDE Quick Start

• Double click "Blank GUI (Default)"

J GUIDE Quick Start						
Create New GUI Open Existing GUI						
GUIDE templates	Preview					
 Blank GUI (Default) GUI with Uicontrols GUI with Axes and Menu Modal Question Dialog 	BLANK					
Save new figure as: \\home-00-ah\juser\Documents\MATLAB\u Browse						
	OK Cancel Help					

GUI Figure

- Display the names of the GUI components
 - Select File > Preferences > GUIDE
 - Check "Show names in component palette"
- Set GUI figure size



Add Components

- 3 push buttons
- A static text area

- A pop-up menu
- An axes



Available Objects/Components

	Component	Description			
	Push Button	Push buttons generate an action when clicked.			
	<u>Slider</u>	Sliders accept numeric input within a specified range by enabling			
Select		the user to move a sliding bar.			
Push Button	Radio Button	Radio buttons are similar to check boxes, but radio buttons are			
Slider		typically mutually exclusive within a group of related radio			
Radio Button		buttons.			
Check Box	Check Box	Check boxes can generate an action when checked and indicate			
Edit Text		their state as checked or not checked.			
TXT Static Text	Edit Text	Edit text components are fields that enable users to enter or			
📼 Pop-up Menu		modify text strings. Use edit text when you want text as input.			
E Listbox	Static Text	Static text controls display lines of text.			
Toggle Button	Pop-Up Menu	Pop-up menus open to display a list of choices when users click			
Table		the arrow.			
Axes	List Box	List boxes display a list of items and enable users to select one			
Panel		or more items.			
Button Group	Togale Button	Toggle buttons generate an action and indicate whether they are			
X ActiveX Control	<u> </u>	turned on or off.			
	<u>Table</u>	Use the table button to create a table component.			
	<u>Axes</u>	Axes enable your GUI to display graphics such as graphs and images.			

Align the Components

Select Tools > Align Objects



Label the Push Buttons

Select View > Property Inspector





GUI Script Structure

```
function varargout = untitled(varargin)
% UNTITLED MATLAB code for untitled.fig
% Begin initialization code - DO NOT EDIT
% --- Executes just before untitled is made visible.
function untitled OpeningFcn(hObject, eventdata, handles, varargin)
% --- Outputs from this function are returned to the command line.
function varargout = untitled OutputFcn(hObject, eventdata, handles)
% --- Executes on button press in pushbutton1.
function pushbutton1 Callback(hObject, eventdata, handles)
% --- Executes on button press in pushbutton2.
function pushbutton2 Callback(hObject, eventdata, handles)
% --- Executes on button press in pushbutton3.
function pushbutton3 Callback(hObject, eventdata, handles)
% --- Executes on selection change in popupmenul.
function popupmenul Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function popupmenul CreateFcn(hObject, eventdata, handles)
•••
```

function untitled_OpeningFcn



Callbck of An Object

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)

```
handles.peaks=peaks(35);
handles.membrane=membrane:
[x, y] = meshgrid(-8:.5:8);
r = sqrt(x.^{2}+y.^{2}) + eps;
sinc = sin(r)./r;
handles.sinc = sinc:
handles.current data = handles.peaks;
surf(handles.current data)
```

What If We Have Two axes?

• Which axes the figure will show on?



handles – information of the GUI objects



Set the axes for Plotting

Modify the script to

```
handles.peaks=peaks(35);
handles.membrane=membrane;
[x,y] = meshgrid(-8:.5:8); r = sqrt(x.^2+y.^2) + eps;
sinc = sin(r)./r; handles.sinc = sinc;
handles.current_data = handles.peaks;
surf(handles.axes1, handles.current_data);
```

or

```
handles.peaks=peaks(35);
handles.membrane=membrane;
[x,y] = meshgrid(-8:.5:8); r = sqrt(x.^2+y.^2) + eps;
sinc = sin(r)./r; handles.sinc = sinc;
handles.current_data = handles.peaks;
axes(handles.axes1); surf(handles.current_data);
```

- •get() acquires properties
- set() sets properties
- Example:

```
a = get(handles.slider1, 'Value');
```

set(handles.text2, 'String', 'TEST');

Practice

- Write a GUI program
- It contains 2 sliders each of which is associated with a variable with value ranged from 0 to 100
- Display the summation of the slider variables as an integer
- You may need these functions:
 - get()
 - set()
 - •int16()
 - num2str()

4	untitled	_ _ ×
A+B=	79	
A: 0		[,] 100
B: 0		100

Handles to Store Variables

- handles is also used to pass variables from one
 GUI object to another
- Store variable a into handles

handles.myData = a; guidata(hObject, handles);

Retrieve variable a from handles

```
a = handles.myData;
```

End of Class

