



Computer Application (MATLAB)

تطبيقات الحاسبة (ماتلاب)
2025-2024

Lecture 12

by

Dr Murtada Dohan

murtada.dohan@uomus.edu.iq





Introduction to MATLAB GUI

- What is a GUI?
 - A Graphical User Interface (GUI) allows users to interact with a program through graphical elements.
 - Replaces command-line execution with buttons, sliders, and input fields.
- Why Use MATLAB for GUI Development?
 - Easy integration with MATLAB functions.
 - Built-in tools for rapid development.
 - Suitable for data visualization and control applications.
- Examples of MATLAB GUIs
 - Data plotting applications.
 - Image processing interfaces.
 - Control system simulations.





MATLAB GUI Development Tools

- GUIDE (Deprecated)
 - Older GUI development tool in MATLAB (removed in R2020a).
 - Code-heavy, **.fig** files.
 - Limited customization.
- App Designer (Recommended)
 - Modern, drag-and-drop environment.
 - Integrated code editor and layout management.
 - Supports better interactivity and event handling.
- Creating a Basic GUI in App Designer
 - Open MATLAB → appdesigner command.
 - Select "Blank App" template.
 - Drag components like buttons, labels, and axes.





App Designer Interface Overview

- Launch via appdesigner or MATLAB toolbar.
- Two Views:
 - Design View: Drag-and-drop components (axes, buttons).
 - Code View: Class-based structure (app.mlapp).
- Key Tools: Component Library, Properties Inspector.
- Visual: Annotated App Designer screenshot.



New to App Designer? Try a 3-minute tutorial.

[Start Tutorial](#)

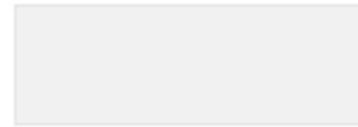
 Open...

Recent Files

No Recent Files

[Getting Started](#) | [How-to Videos](#) | [GUIDE Migration Strategies](#) | [Display Graphics in App Designer](#) | [Find and Create UI Components](#) | [Release Notes](#)

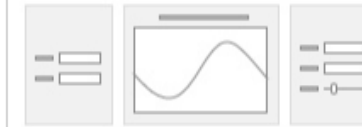
Apps



Blank App



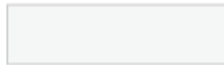
2-Panel App with Auto-Reflow



3-Panel App with Auto-Reflow

[Show examples](#)

Custom UI Components



Blank Component



Blank Component with Grid Layout

[Show examples](#)

COMMON

Axis

Button

Check Box

Date Picker

Drop Down

Edit Field (Numeric)

Edit Field (Text)

HTML

Hyperlink

Image

Label

List Box

Radio Button Group

Slider

Spinner

State Button

Table

Text Area

Toggle Button Group

Tree

Tree (Check Box)

CONTAINERS

Grid Layout

Panel

Tab Group

FIGURE TOOLS

Context Menu

Menu Bar

Toolbar

INSTRUMENTATION

90 Degree Gauge

Discrete Knob

Gauge

Knob

Lamp

Linear Gauge

Rocker Switch

Semicircular Gauge

Switch

app1

app.UIFigure

app.Button

app.EditField2

app.EditField2

app.EditField2

App

Callbacks

SHARING DETAILS

Name

app1

Version

1.0

Author

Summary

Description

CODE OPTIONS

Single Running Instance

Input Arguments

Edit Field

Edit Field2

Button



Layout and Design Principles

- Organizing Components
 - Use panels and tabs for structured layout.
 - Group related controls together.
- Component Alignment and Sizes
 - Maintain uniform spacing.
 - Resize components for different screen sizes.
- Best Practices
 - Use intuitive labels and tooltips.
 - Keep the interface simple and functional.





UI Components

- Common Components:
 - Buttons, sliders, axes, edit fields, labels.
- Customization:
 - Use Properties Inspector to adjust size, labels, limits.
 - Visual: Grid of component icons with labels.





Event Handling & Callbacks

- What Are Callbacks?
 - Functions triggered by user actions (e.g., button click).
- Adding Interactivity
 - Assign functions to UI components.
 - Example: Button click to update a plot.
- Common Callback Functions
 - ButtonPushed for buttons.
 - ValueChanged for sliders and dropdowns.

```
function ButtonPushed(app, event)
    x = linspace(0,10,100);
    y = sin(x);
    plot(app.UIAxes, x, y);
end
```





Working with Data in MATLAB GUI

- Reading User Input
 - Use `EditField` and `TextArea` to collect input.
- Displaying Results
 - Use `Label`, `Table`, and `Axes` components.
- Real-time Data Updates
 - Use `drawnow` to refresh UI components dynamically.





Interfacing MATLAB GUI with External Files & Databases

- Importing/Exporting Data
 - Load files using `uigetfile`.
 - Save data using `xlswrite` or `save`.
- Database Connection
 - Connect to SQL databases using `database` function.
- Dynamic Data Plotting
 - Read sensor data and update plots in real-time.





Advanced GUI Features

- Using Multiple Windows
 - Create dialogs with uifigure.
 - Example: Confirmation dialogs before actions.
- Adding Animations
 - Use pause and loops to create dynamic plots.
- MATLAB Code Integration
 - Call external scripts/functions from the GUI.





Debugging and Optimization

- Common MATLAB GUI Issues
 - Unresponsive UI elements.
 - Callback function errors.
- Performance Optimization Tips
 - Use vectorized computations.
 - Avoid excessive redraw operations.
- Debugging Tools
 - Use breakpoints and `disp()` statements.
 - MATLAB's Debugger for step-by-step execution.





Conclusion

- MATLAB GUI development is user-friendly and powerful.
- App Designer provides a modern approach with intuitive UI design tools.
- Mastering event-driven programming is key to creating interactive applications.
- Practice by building small GUI projects and gradually adding complexity.

