Introduction to Simulink

Mughees Asif

MSc. Artificial Intelligence

MathWorks Campus Ambassador

Outline

What is Simulink?

Working with Simulink

How Simulink works

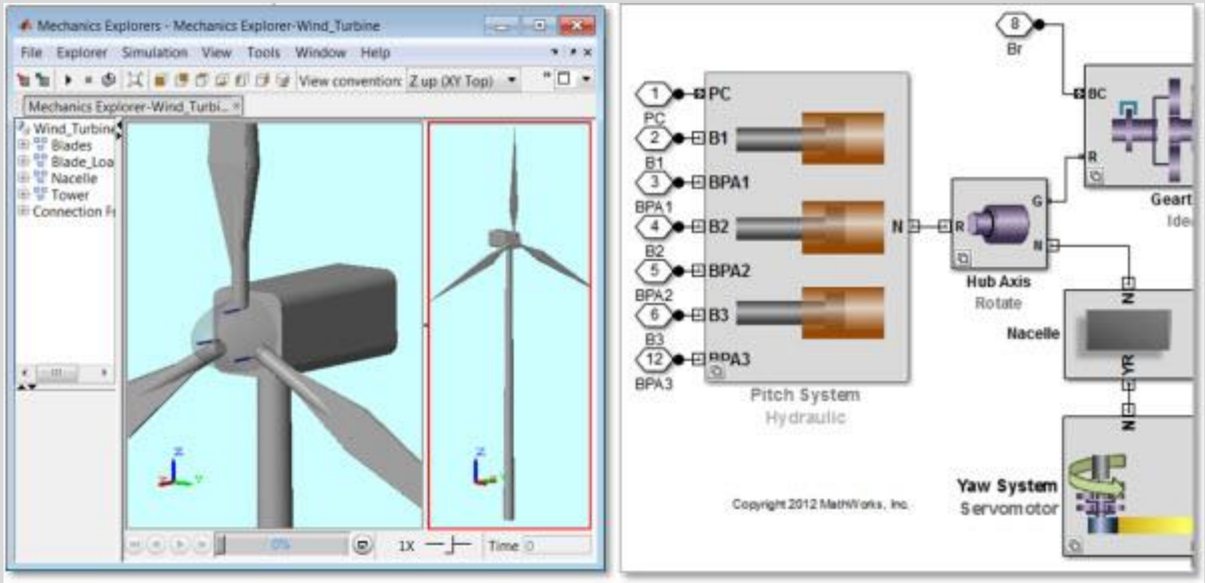
Decoupling models

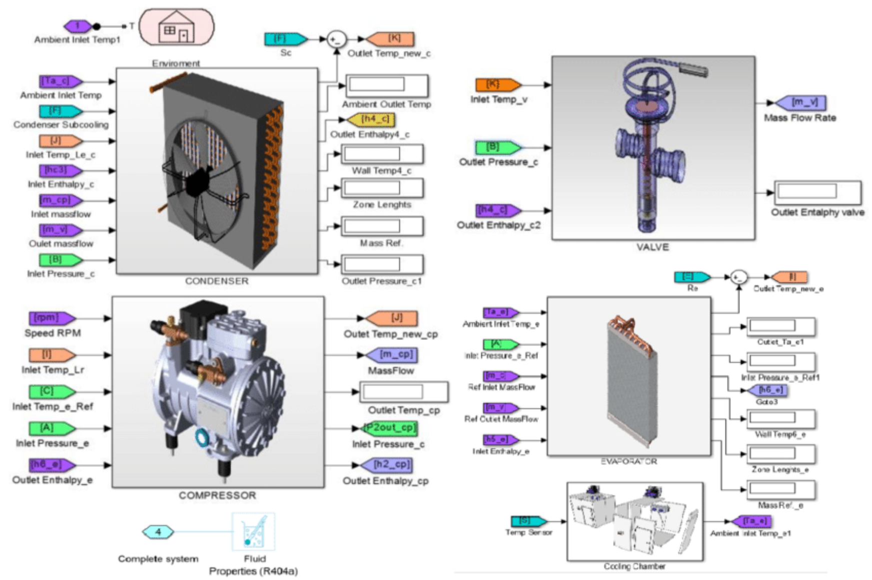
Continuous and discrete models

Simulink

Simulink is a software package for modeling, simulating, and analyzing dynamic systems:

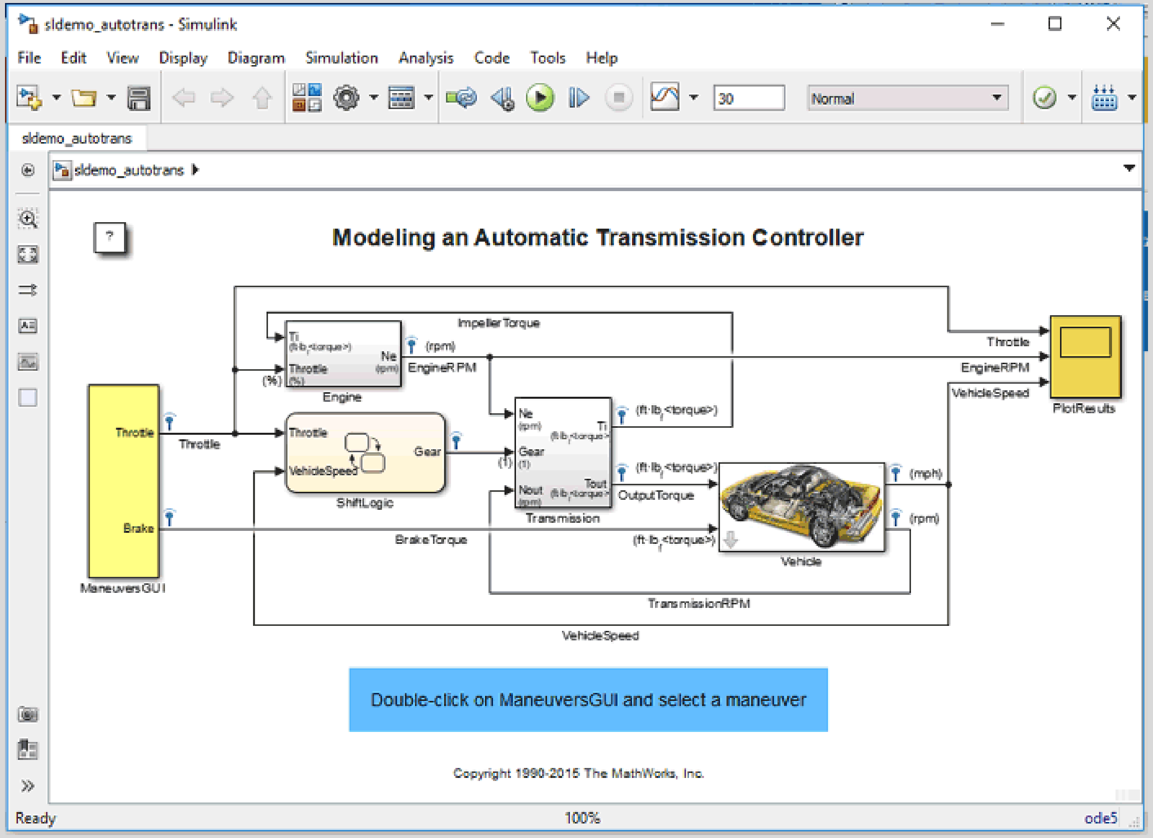
- Block diagram editing
- Non-linear simulation
- Hybrid (continuous and discrete) models
- Asynchronous (non-uniform sampling) simulation
- Fully integrated with MATLAB → MATLAB toolboxes and block sets


Mechanics Explorers - Mechanics Explorer-Wind_Turbine
File Explorer Simulation View Tools Window Help
View convention: Z up (XY Top)
Mechanics Explorer-Wind_Turbi...
Wind_Turbine
Blades
Blade_Loa
Nacelle
Tower
Connection Fr
1X
Time
Br
PC
B1
BPA1
B2
BPA2
B3
BPA3
Pitch System
Hydraulic
Hub Axis
Rotate
Nacelle
Geart
Yaw System
Servomotor
Copyright 2012 MathWorks, Inc.



Simulink

- **Accurately design, implement, and test**:
  - Control systems
  - Signal Processing systems
  - Communications systems
  - Embedded systems
  - Physical systems
  - other Dynamical systems



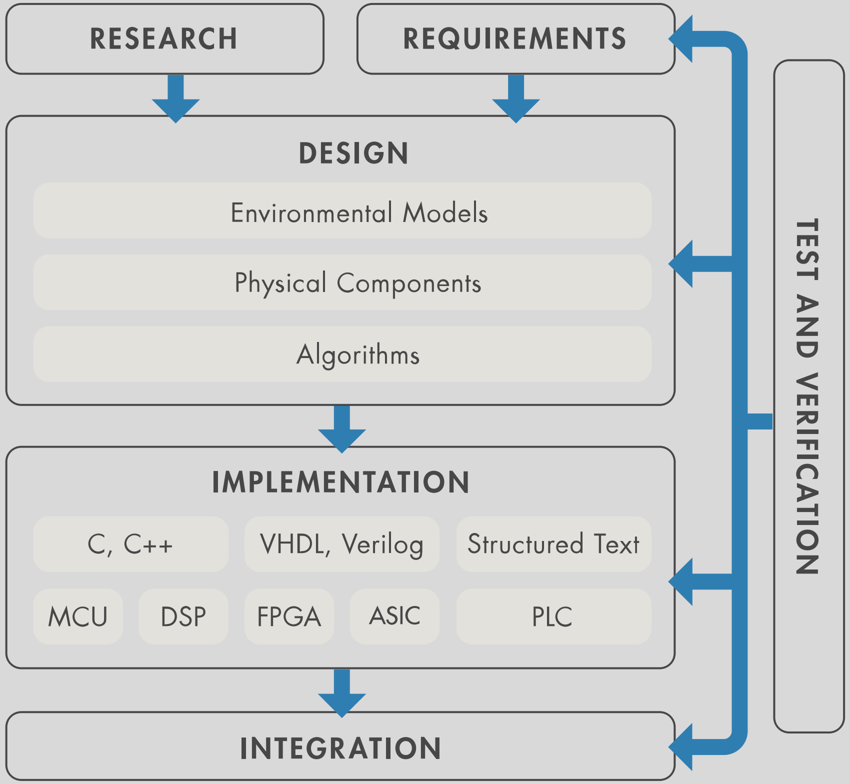
Modeling an Automatic Transmission Controller

Double-click on ManeuversGUI and select a maneuver

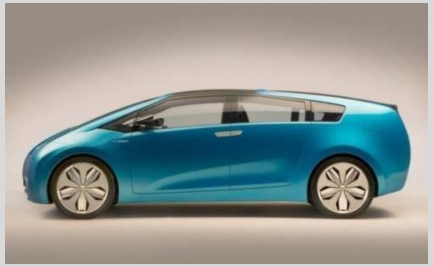
Copyright 1990-2015 The MathWorks, Inc.

Model Based Design with Simulink

- **Definition:**
  - A *model* is defined as a representation of a system for the purpose of studying the system.
- **Types:**
  - Static vs. dynamic
  - Deterministic vs. stochastic
  - Discrete vs. continuous
- **Implementation:**
  - Automatic code generation
  - Rapid prototyping for HIL, SIL
  - Verification and validation



Simulink Applications



Bell Helicopter Develops the First Civilian Tiltrotor, Using Model-Based Design

The BA609, flying in airplane mode.

Challenge

To design and build the BA609, the first and fastest commercially available tiltrotor aircraft in the world

Solution

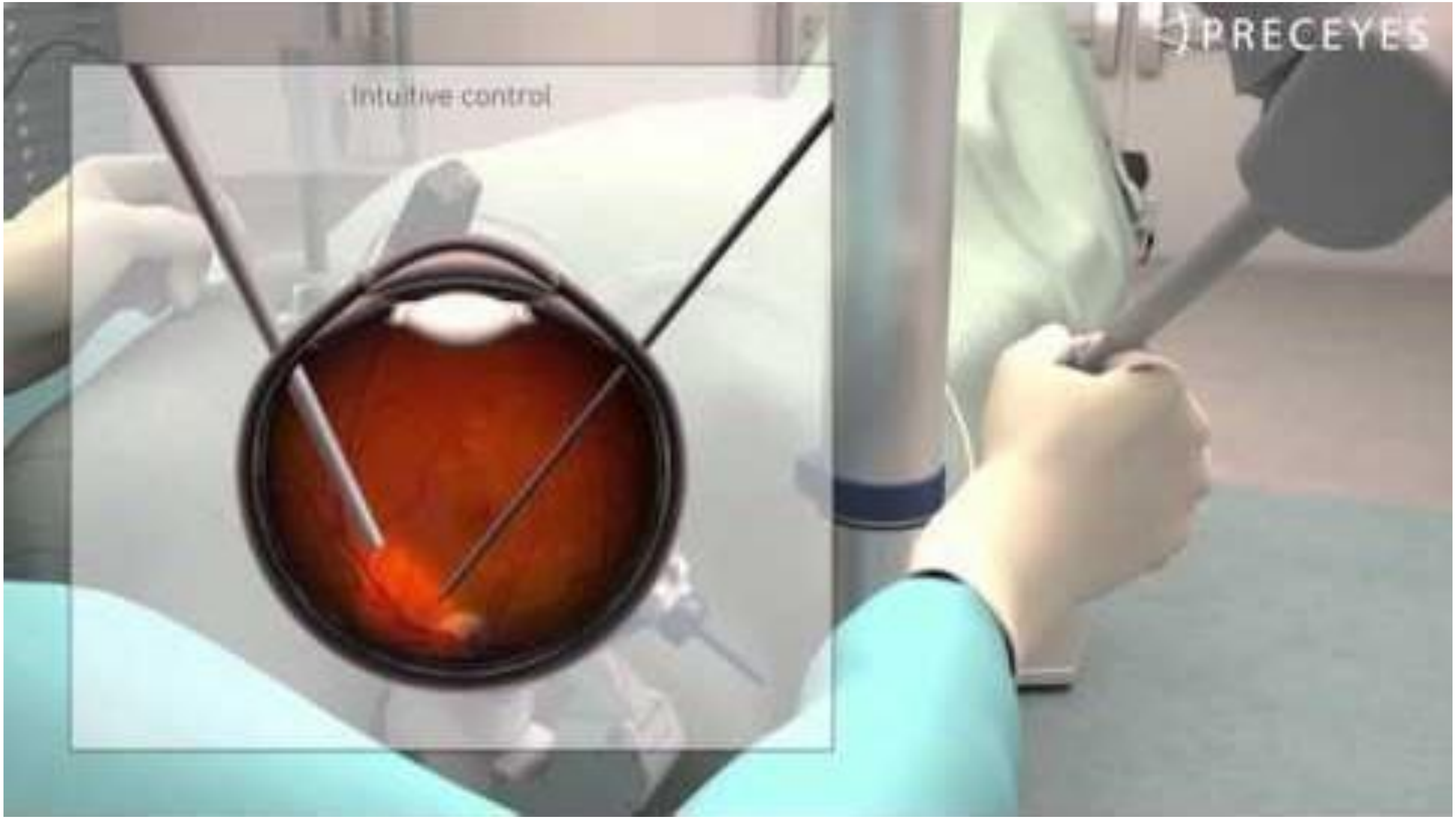
Use Model-Based Design with MATLAB, Simulink, and Real-Time Workshop software to model, simulate, test, and verify designs

Results

- Full collaboration with suppliers via Simulink models
- Flight control system code generated automatically from models
- 40% improvement in design and development time
- Flawless first flight, which went exactly like the simulation

"Simulations and a rapid, iterative approach enabled us to minimize the unknowns and ensure that we had established enough margin that when weran into a surprise we could continue to have a safe flight test program—and run it with unprecedented efficiency."

David King
Bell Helicopter


Intuitive control
PRECEYES

PRECEYES Accelerates Development of World’s First Eye-Surgery Robot Using Model-Based Design

The PRECEYES Surgical System

Challenge

Develop a real-time control system for robot-assisted surgical procedures performed within the human eye

Solution

Use Model-Based Design with MATLAB and Simulink to model and simulate the control system and use Simulink Coder and Simulink Real-Time to deploy it to a real-time target

Results

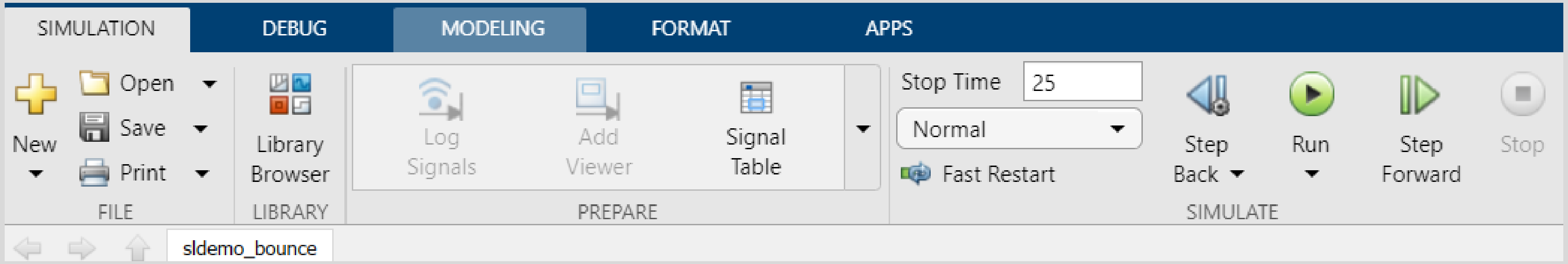
- Core controller developed by one engineer
- Patient safety assured
- Road map to industrialization set

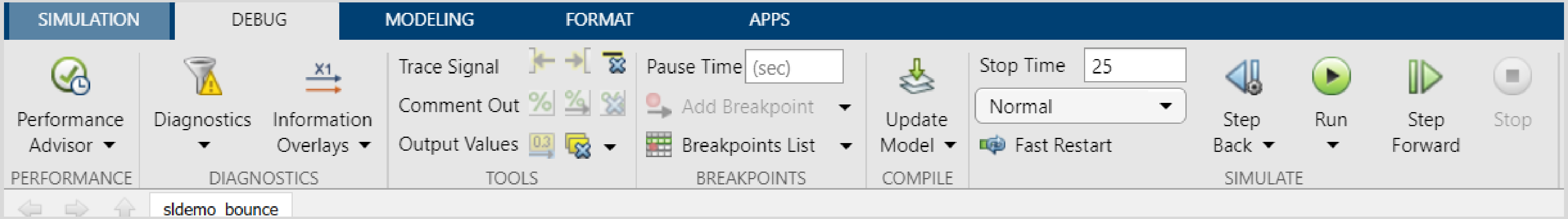
> **“MATLAB and Simulink provided a single platform that supported our complete workflow and all the components and protocols we needed for our robotic system. That enabled us to quickly develop a safe, real-time device, ready for clinical investigation.”**
>
> Maarten Beelen,
> PRECEYES

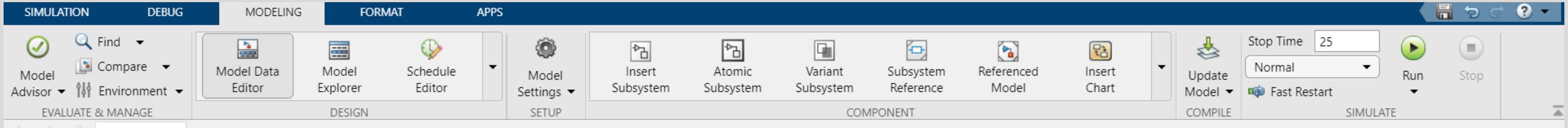
Outline

- Why Simulink?
- **Working with Simulink**
- How Simulink works
- Decoupling models
- Continuous and discrete models

Simulink Toolbar



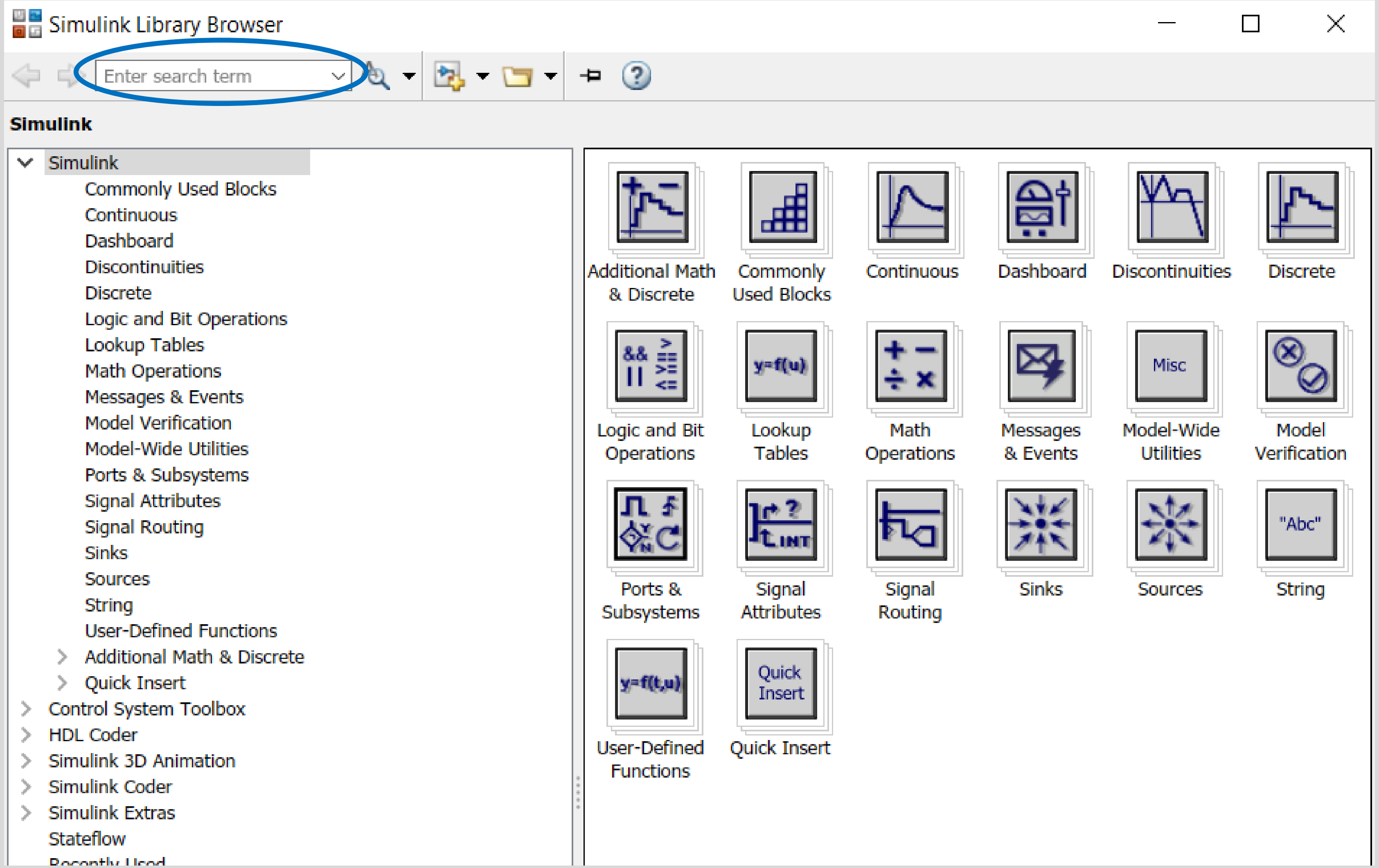




Overview of the library

Library Name	Examples
Sources	Constant, Sine Wave, Steps
Sinks	Scope, XY Graphs
Math operations	Add, divide, absolute
Ports & subsystems	Subsystem, Enable port, Inputs and Outputs: In1 and Out1
User defined functions	Fcn, MATLAB Fcn
Lookup tables	1D Lookup table
Signal Routing	Mux, BusCreator, Goto, Switch
Continuous	Integrator, Derivative
Discrete	Unit delay, Discrete Derivative
Logical and Bit operations	Compare to Zero, Logical operators

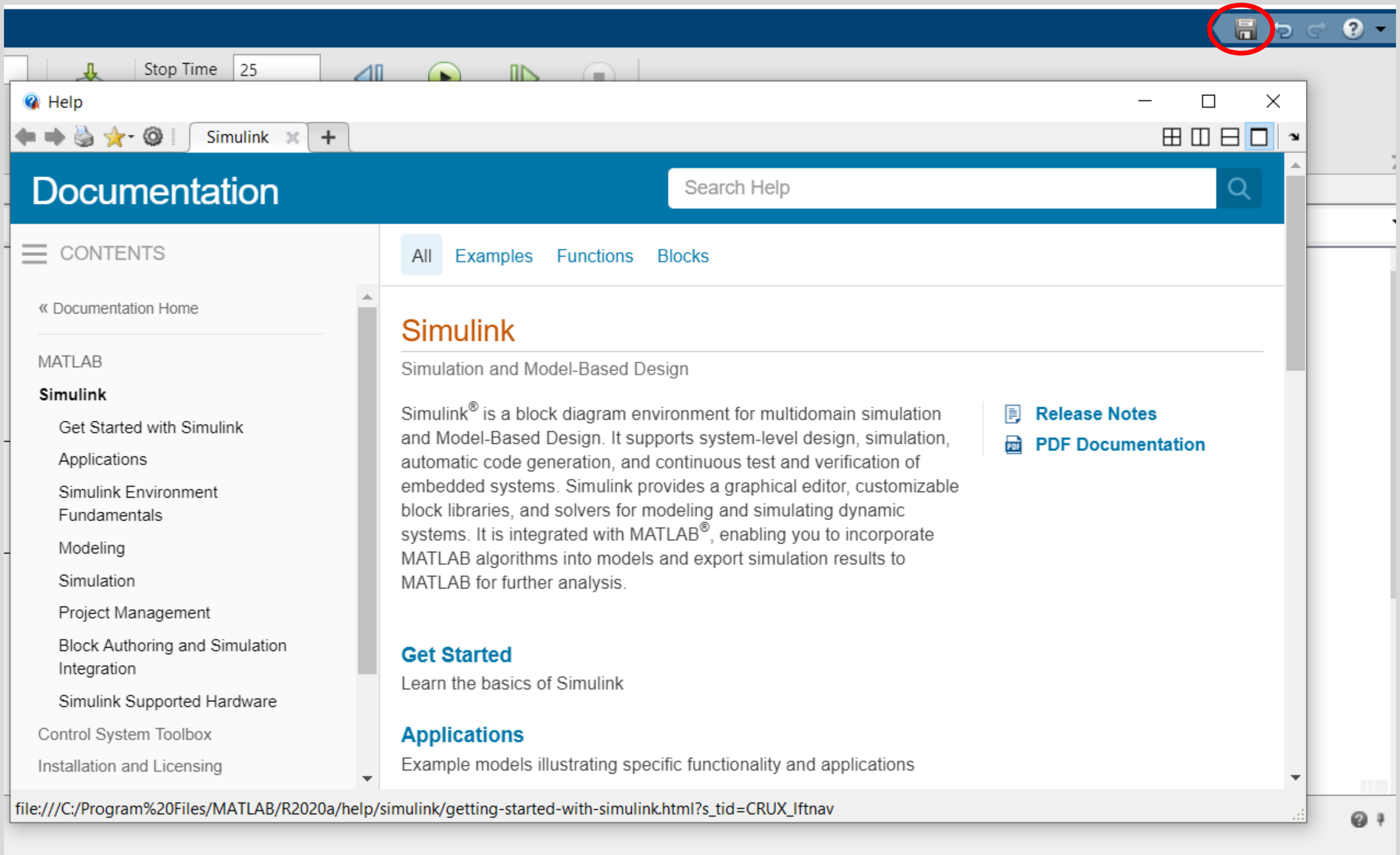
Finding Blocks



Simulink
- Commonly Used Blocks
- Continuous
- Dashboard
- Discontinuities
- Discrete
- Logic and Bit Operations
- Lookup Tables
- Math Operations
- Messages & Events
- Model Verification
- Model-Wide Utilities
- Ports & Subsystems
- Signal Attributes
- Signal Routing
- Sinks
- Sources
- String
- User-Defined Functions
- Additional Math & Discrete
- Quick Insert

Control System Toolbox
HDL Coder
Simulink 3D Animation
Simulink Coder
Simulink Extras
Stateflow
Recently Used

Getting Help



Simulink

Simulation and Model-Based Design

Simulink® is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB®, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

Get Started

Learn the basics of Simulink

Applications

Example models illustrating specific functionality and applications

file:///C:/Program%20Files/MATLAB/R2020a/help/simulink/getting-started-with-simulink.html?s_tid=CRUX_lftnav

Outline

Why Simulink?

Working with Simulink

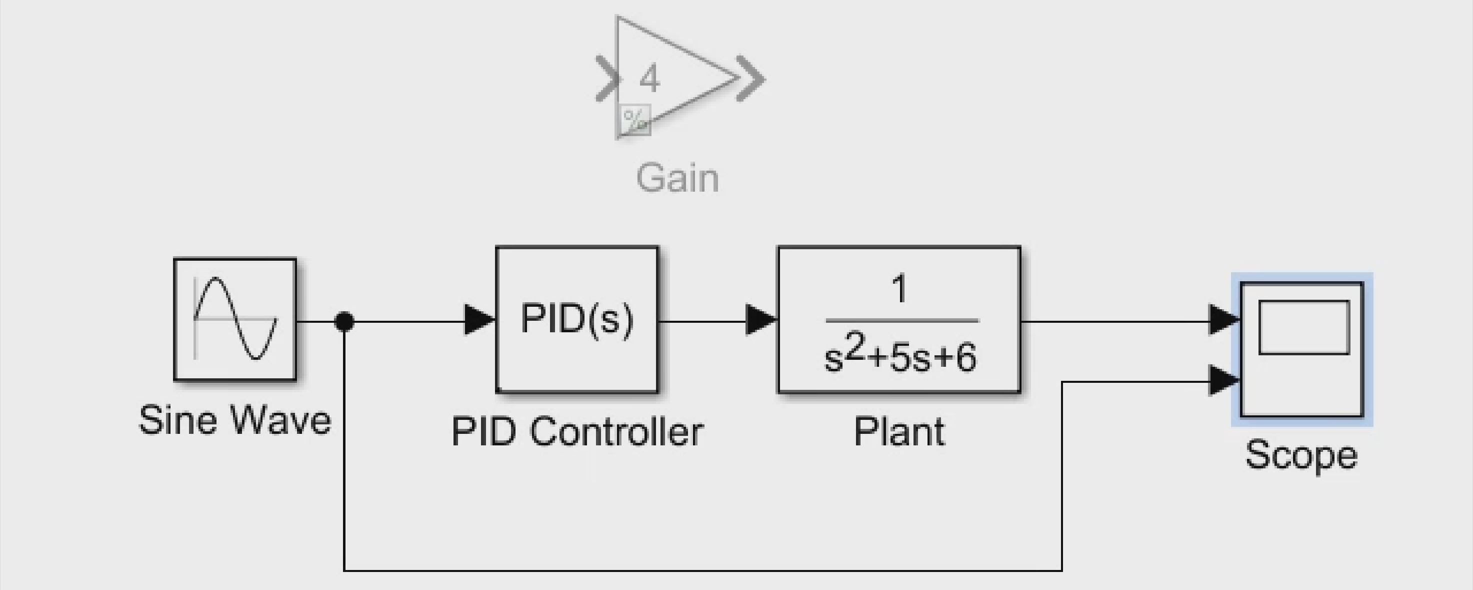
How Simulink works

Decoupling models

Continuous and discrete models

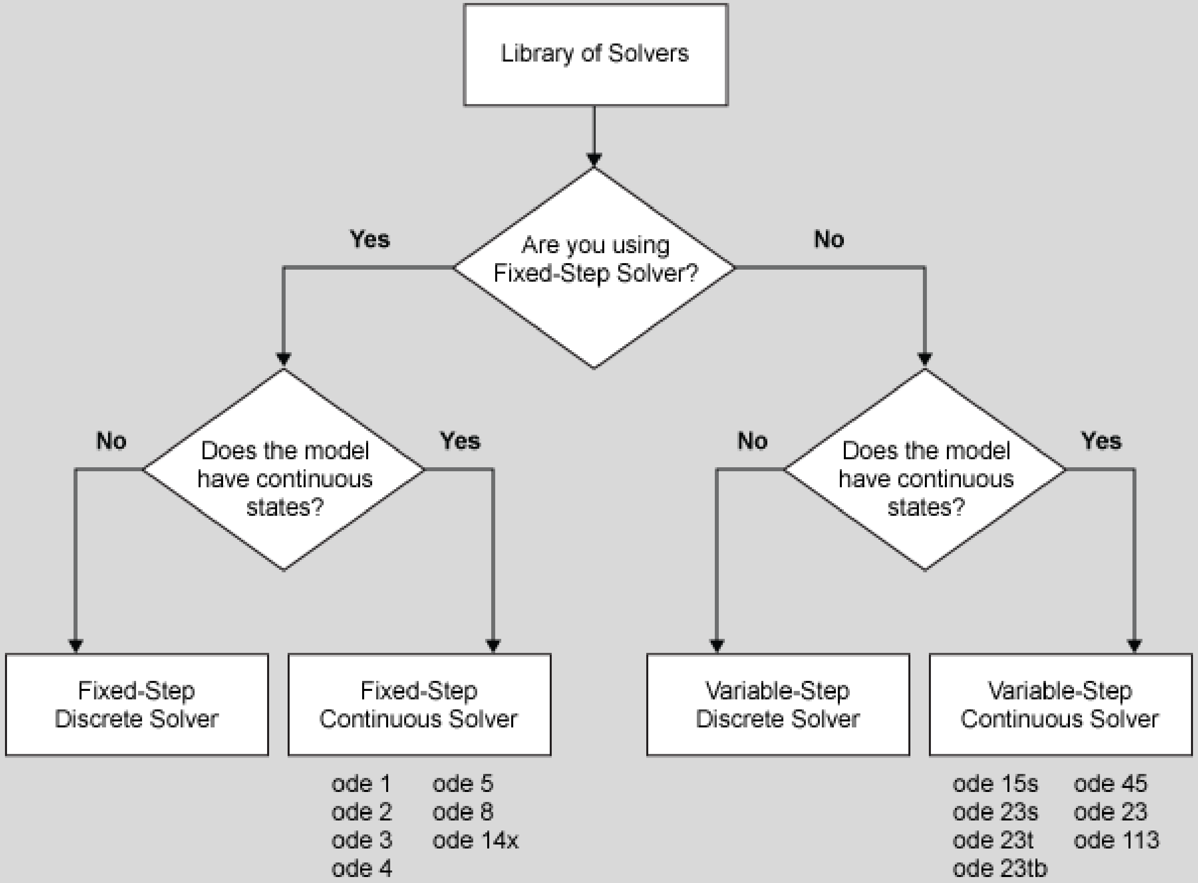
How Simulink Works

- Engine provides variable-step and fixed-step ODE solvers
- Block diagram representation of dynamic systems
- Blocks define governing equations
- Signals are propagated between blocks over time



Simulink Solvers

- **Solver**:
  - Determines solution at current time step
  - Determines the next simulation time step
  - MATLAB naming convention: *ode*
- **Selection Criteria**:
  - System dynamics
  - Solution stability
  - Computation speed
  - Solver robustness



Library of Solvers
Are you using Fixed-Step Solver?
Yes
No
Does the model have continuous states?
No
Yes
Does the model have continuous states?
No
Yes
Fixed-Step Discrete Solver
Fixed-Step Continuous Solver
Variable-Step Discrete Solver
Variable-Step Continuous Solver
ode 1
ode 2
ode 3
ode 4
ode 5
ode 8
ode 14x
ode 15s
ode 23s
ode 23t
ode 23tb
ode 45
ode 23
ode 113

Outline

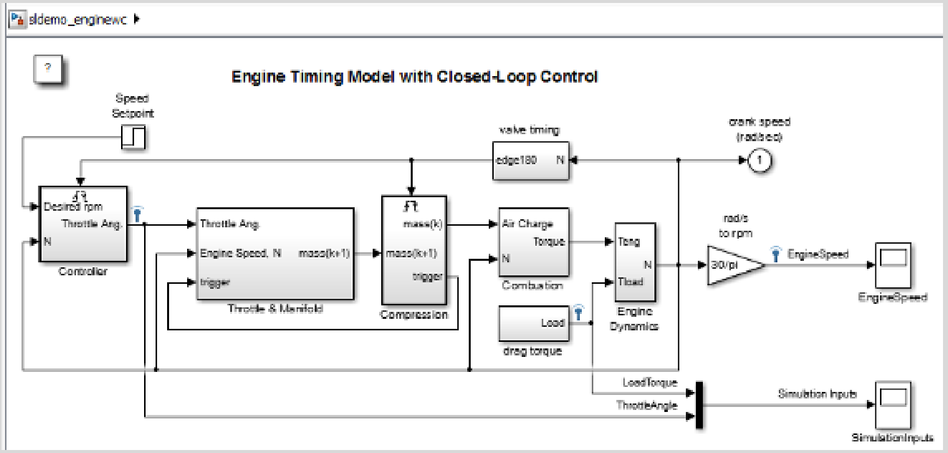
- Why Simulink?
- Working with Simulink
- How Simulink works
- **Decoupling models**
- Continuous and discrete models

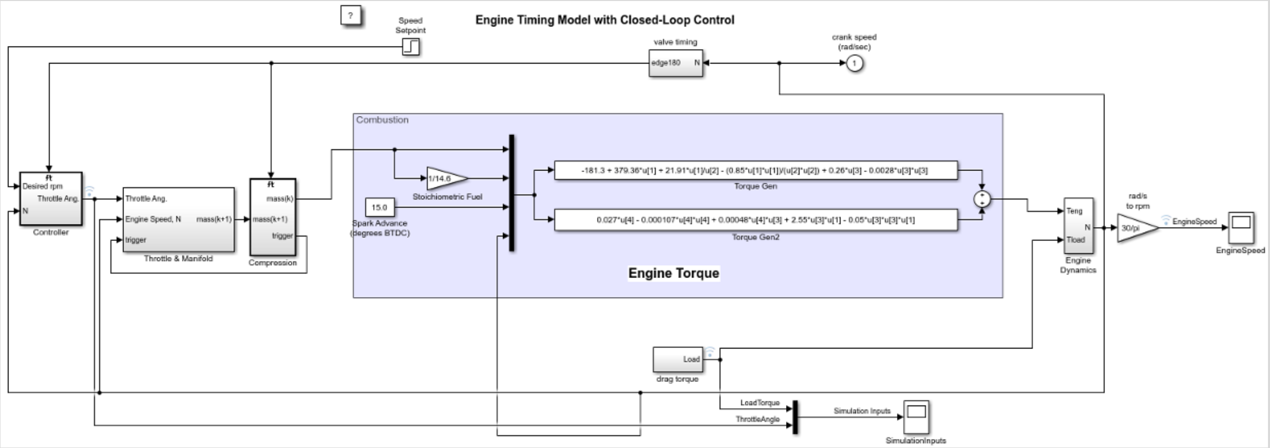
Subsystems

- Group blocks into functional subsystems, ***create model hierarchy***.
- As a model increases in size and complexity, you can simplify it by grouping blocks into ***subsystems***. A subsystem is a set of blocks that you group into an overall single block.
- **Using subsystems**:
  - Establishes a hierarchical block diagram, where a Subsystem block is on one layer and the blocks that make up the subsystem are on another.
  - Keeps functionally related blocks together.
  - Helps reduce the number of blocks displayed in your model window.
  - Establishes an interface with inputs and outputs.

Creating Subsystems

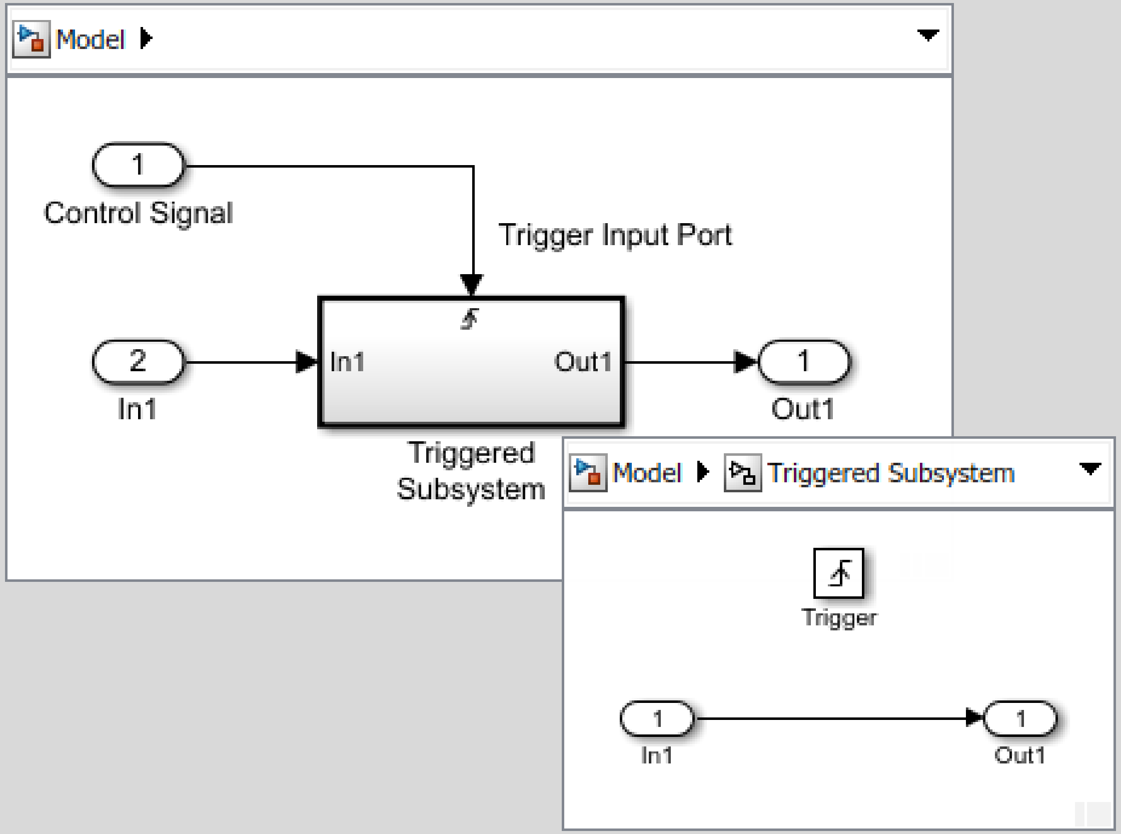
- **To create a subsystem, you can:**
  - In the Simulink Editor, double-click and start typing the subsystem type, then select the corresponding block from the menu.
  - In the Simulink Editor, drag a selection box to outline the subsystem that you want to create, then select the subsystem type.
  - Drag a Subsystem block from the Library Browser.
  - Copy and paste a Subsystem block from a model.





Model Referencing

- One model in another – ***parent and referenced model***
- **Advantages**:
  - Modular development
  - Model protection
  - Inclusion by reference
  - Incremental loading
  - Accelerated simulation
  - Incremental code generation
  - Independent configuration sets

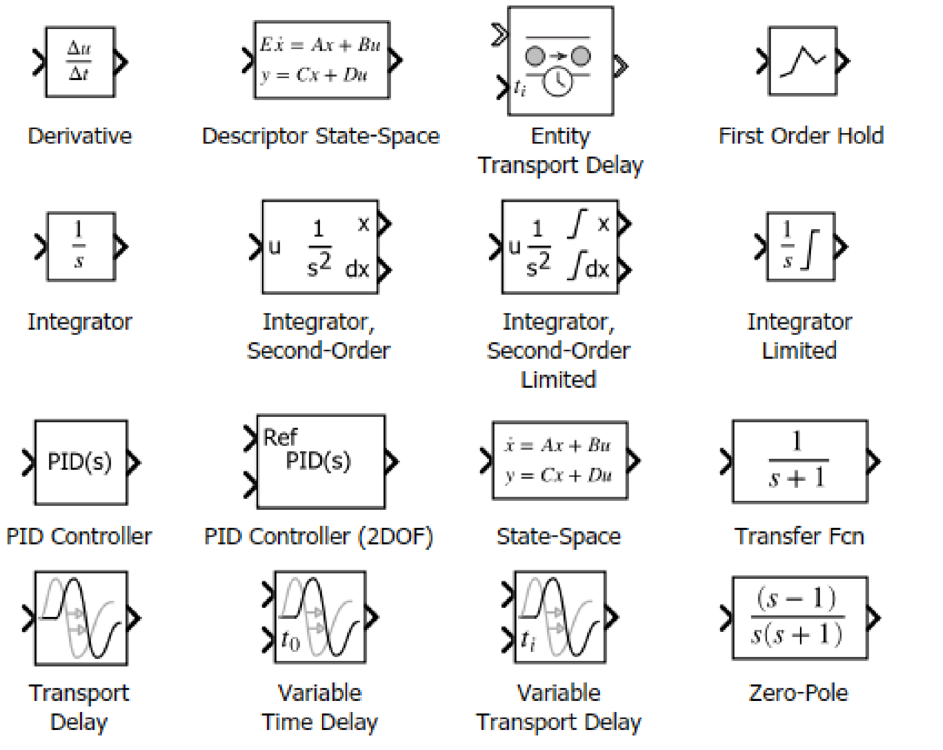


Outline

- Why Simulink?
- Working with Simulink
- How Simulink works
- Decoupling models
- **Continuous and discrete models**

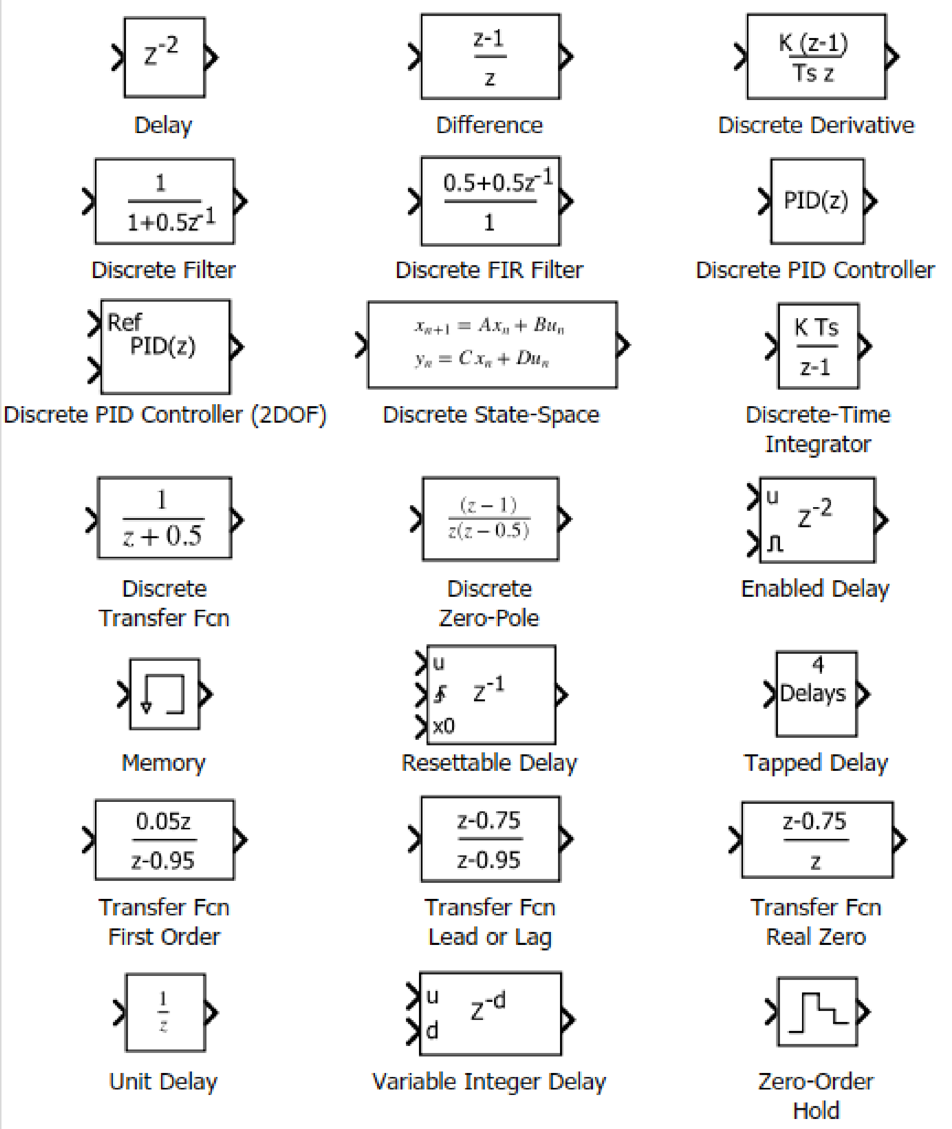
‘Continuous’ Library

- Use blocks from the Continuous library to model ***differential equations***.
    - Time derivative of a signal.
    - Integrate or delay a signal.
    - Model PID controllers and linear systems using transfer function or state-space representations.



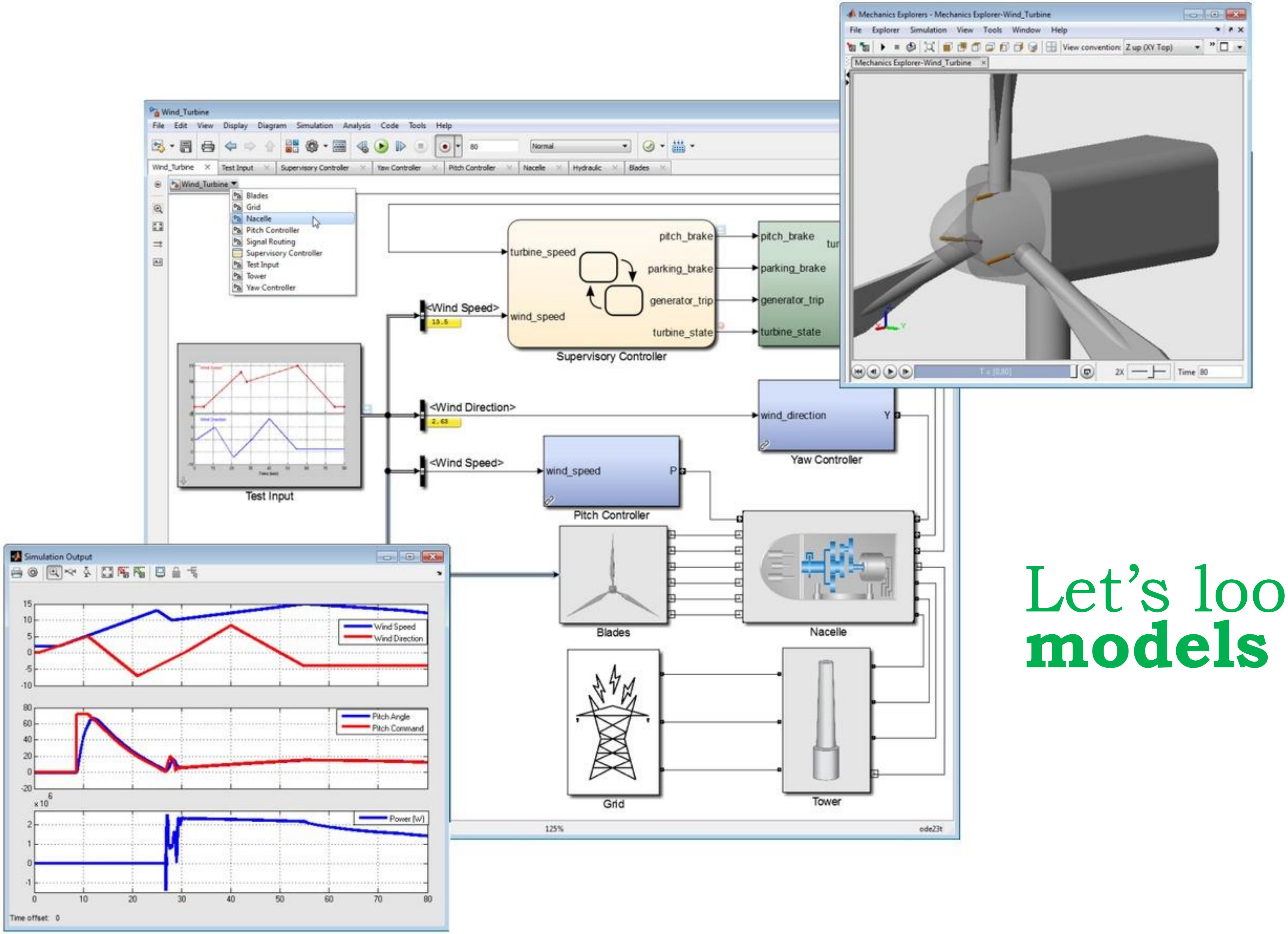
‘Discrete’ Library

- Use blocks from the Discrete library to model ***recurrence equations***.
- Discrete time function blocks such as Unit Delay.



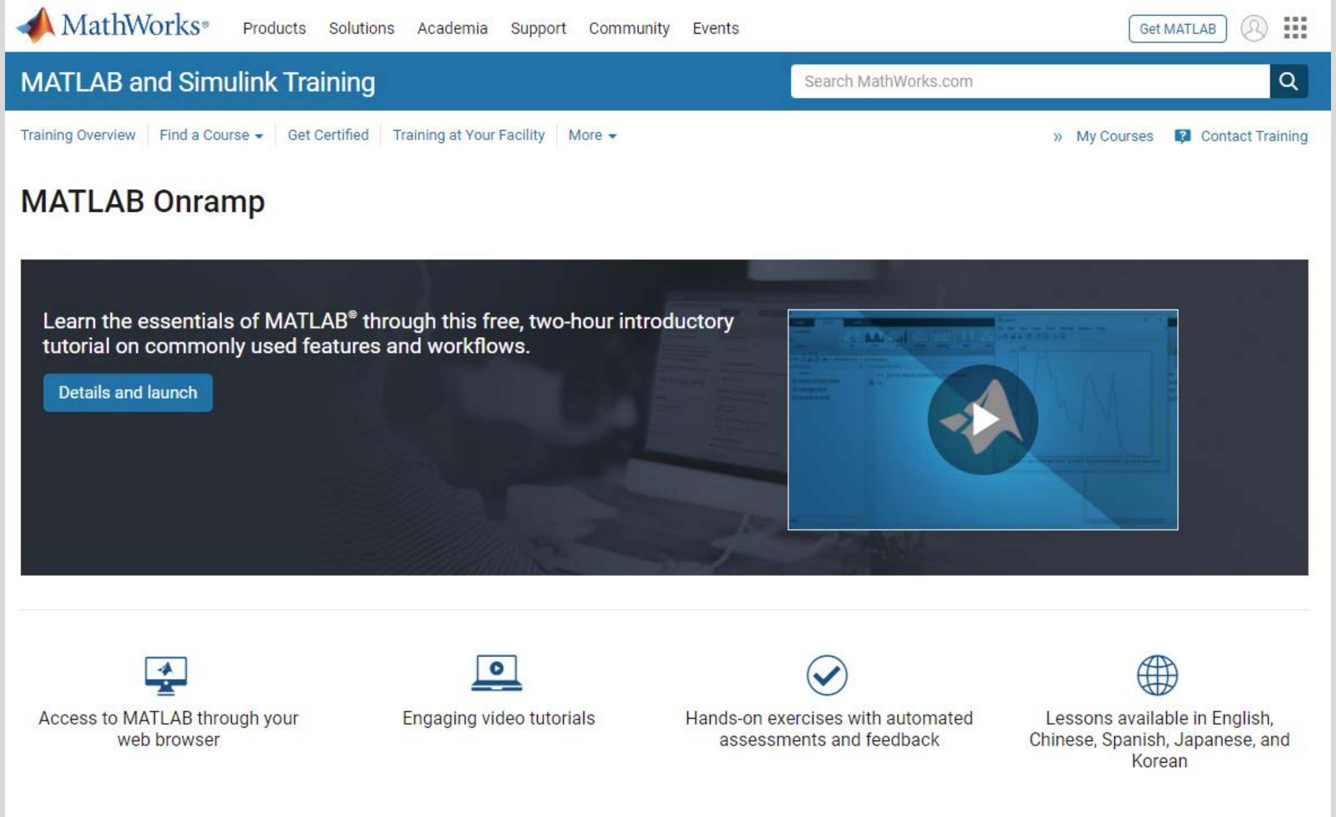
Summary

- Why Simulink?
- Working with Simulink
- How Simulink works
- Componentizing models
- Continuous and discrete models



Let's look at some **models** now!

https://matlabacademy.mathworks.com/#getting-started



MATLAB Onramp

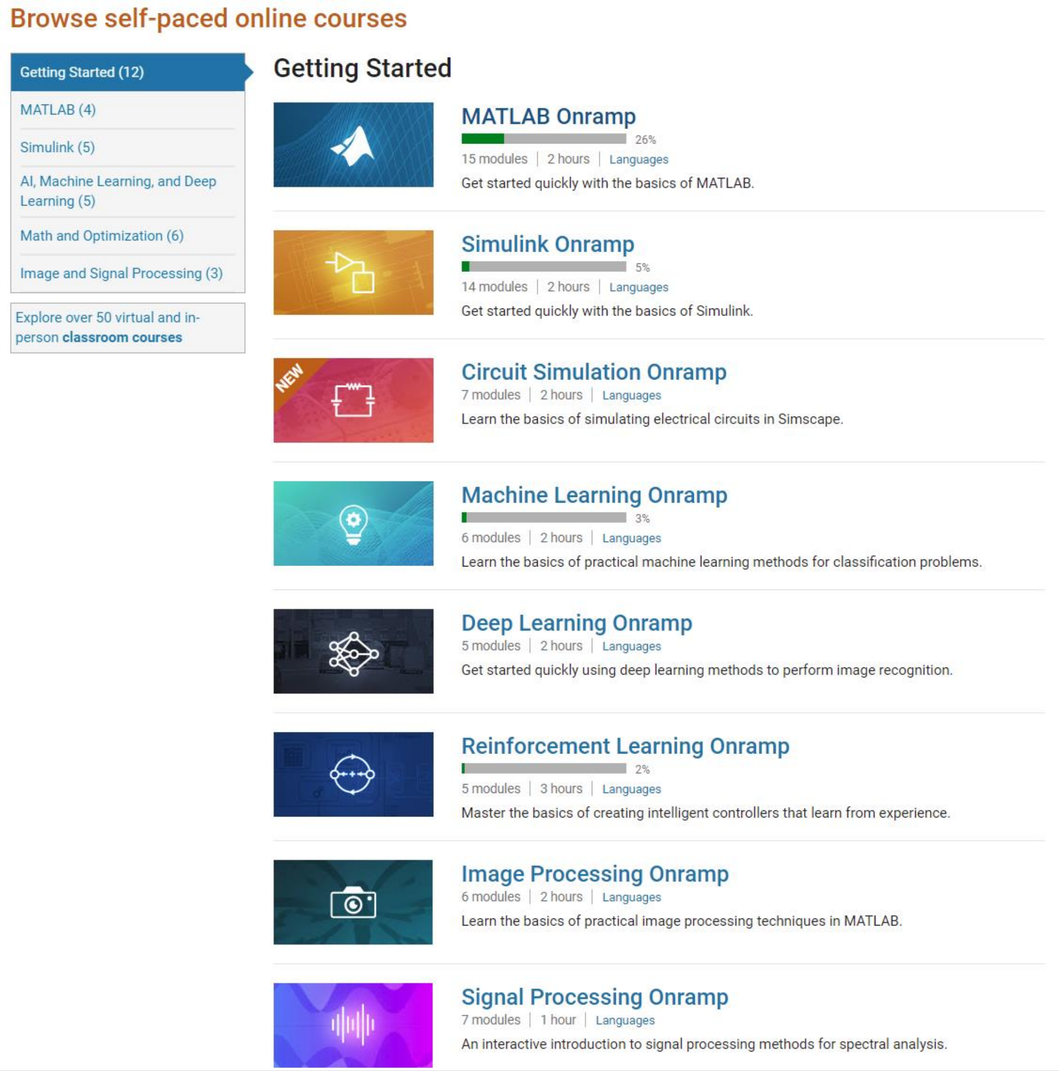
Learn the essentials of MATLAB® through this free, two-hour introductory tutorial on commonly used features and workflows.

Access to MATLAB through your web browser

Engaging video tutorials

Hands-on exercises with automated assessments and feedback

Lessons available in English, Chinese, Spanish, Japanese, and Korean


Getting Started (12)
MATLAB (4)
Simulink (5)
AI, Machine Learning, and Deep Learning (5)
Math and Optimization (6)
Image and Signal Processing (3)
NEW

Browse self-paced online courses

Explore over 50 virtual and in-person **classroom courses**

Getting Started

MATLAB Onramp
26%
15 modules | 2 hours | Languages
Get started quickly with the basics of MATLAB.

Simulink Onramp
5%
14 modules | 2 hours | Languages
Get started quickly with the basics of Simulink.

Circuit Simulation Onramp
7 modules | 2 hours | Languages
Learn the basics of simulating electrical circuits in Simscape.

Machine Learning Onramp
3%
6 modules | 2 hours | Languages
Learn the basics of practical machine learning methods for classification problems.

Deep Learning Onramp
5 modules | 2 hours | Languages
Get started quickly using deep learning methods to perform image recognition.

Reinforcement Learning Onramp
2%
5 modules | 3 hours | Languages
Master the basics of creating intelligent controllers that learn from experience.

Image Processing Onramp
6 modules | 2 hours | Languages
Learn the basics of practical image processing techniques in MATLAB.

Signal Processing Onramp
7 modules | 1 hour | Languages
An interactive introduction to signal processing methods for spectral analysis.

Thank you!

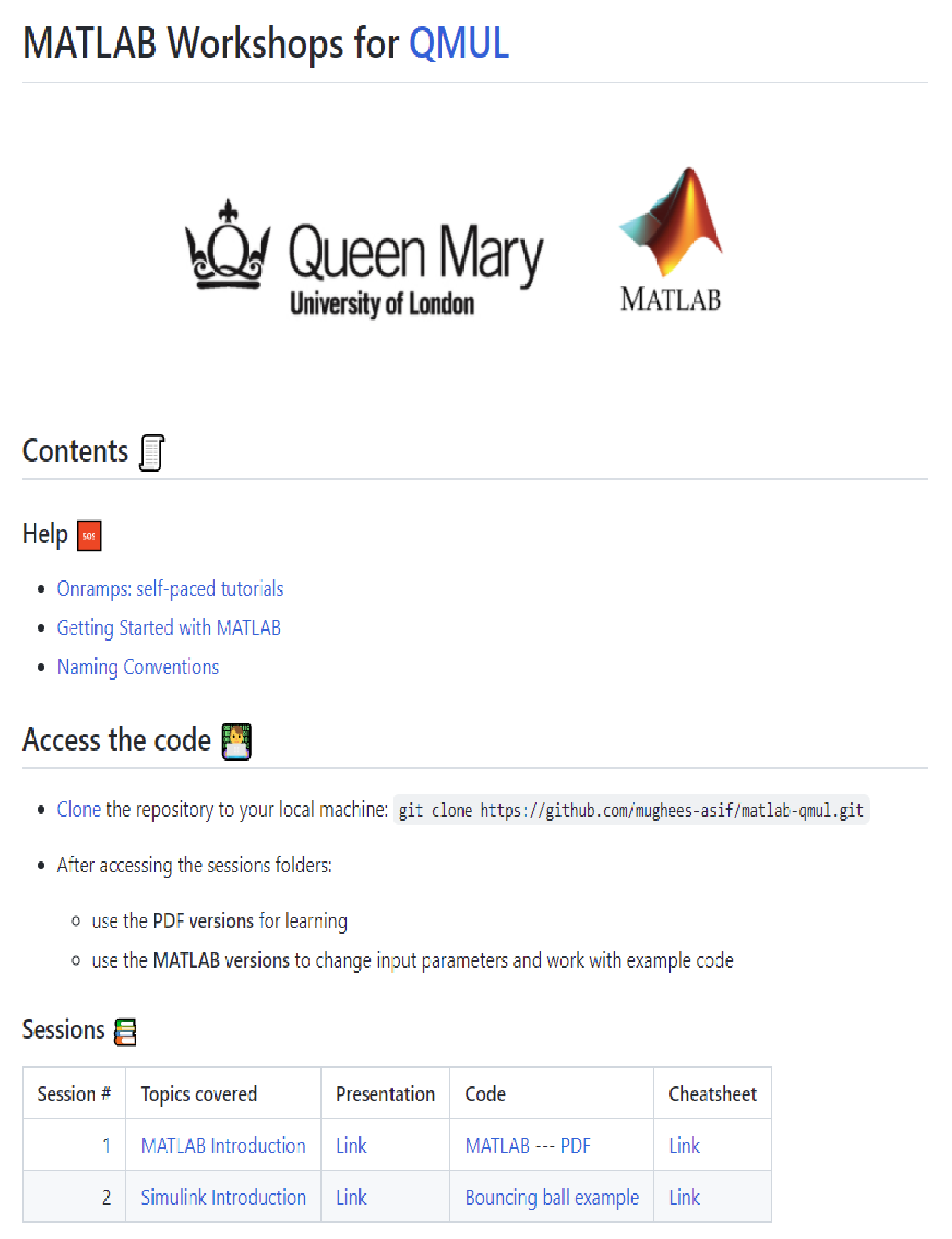
MATLAB® & SIMULINK®

Join the FB group to stay up to date with future events:

https://www.facebook.com/groups/196042678284982

The code and presentation can be downloaded from:

https://github.com/mughees-asif/matlab-qmul



MATLAB Workshops for QMUL

Contents 📜

Help 🆘

- Onramps: self-paced tutorials
- Getting Started with MATLAB
- Naming Conventions

Access the code 🧑‍💻

- Clone the repository to your local machine: `git clone https://github.com/mughees-asif/matlab-qmul.git`
- After accessing the sessions folders:
  - use the **PDF versions** for learning
  - use the **MATLAB versions** to change input parameters and work with example code

Sessions 📚

Session #	Topics covered	Presentation	Code	Cheatsheet
1	MATLAB Introduction	Link	MATLAB --- PDF	Link
2	Simulink Introduction	Link	Bouncing ball example	Link