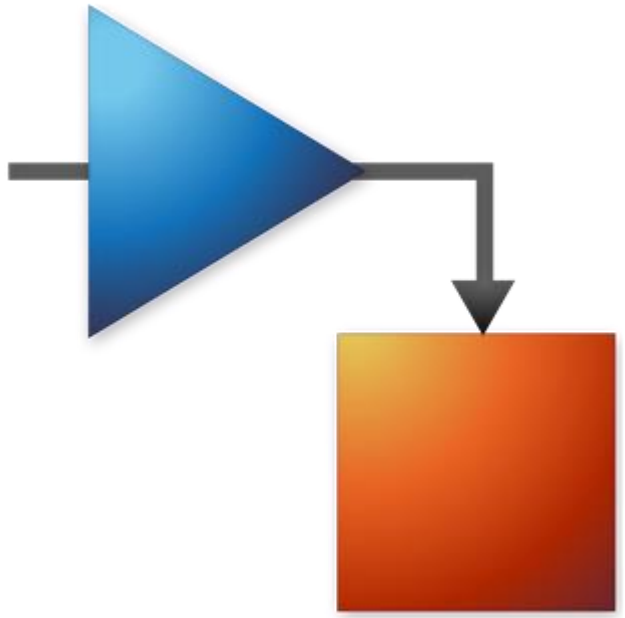
# Introduction to Simulink

**Mughees Asif**
3rd Year Aerospace Engineering
QMUL MathWorks Student Ambassador



# Outline

**What is Simulink?**

Working with Simulink

How Simulink works

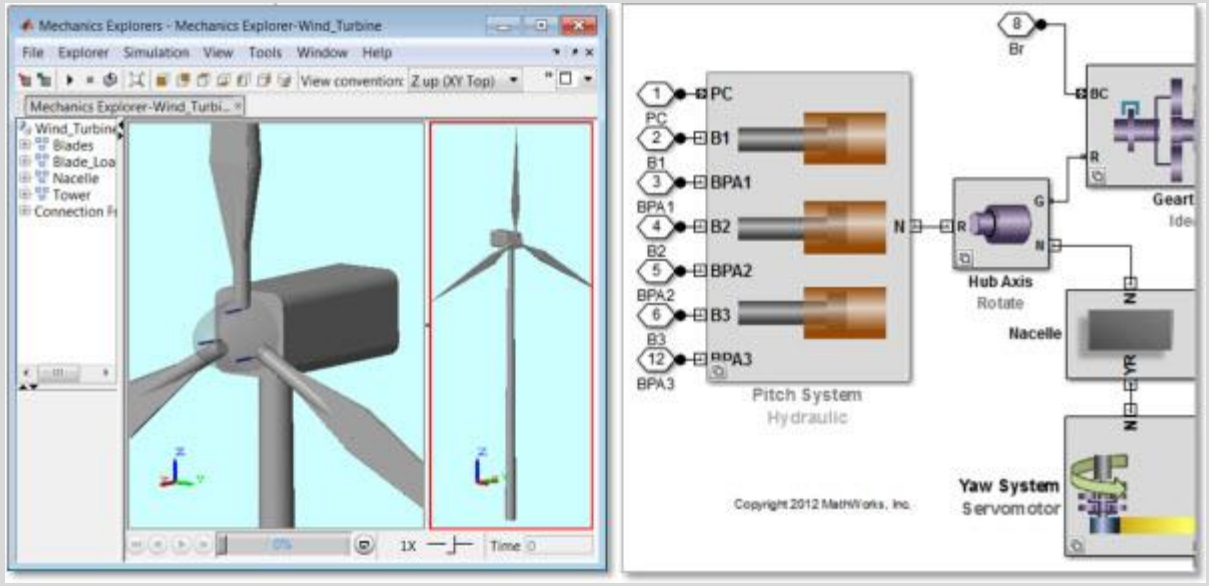
Decoupling models

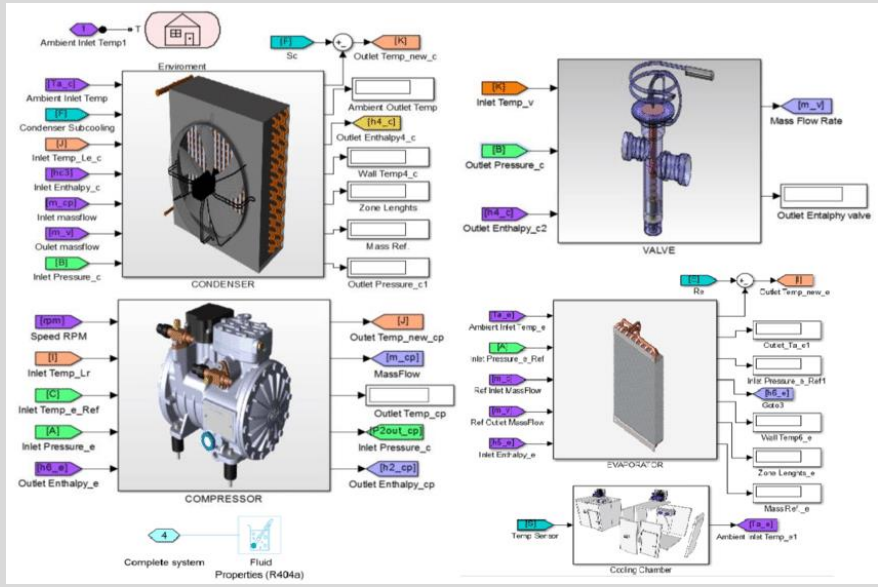
Continuous and discrete models

# Simulink

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

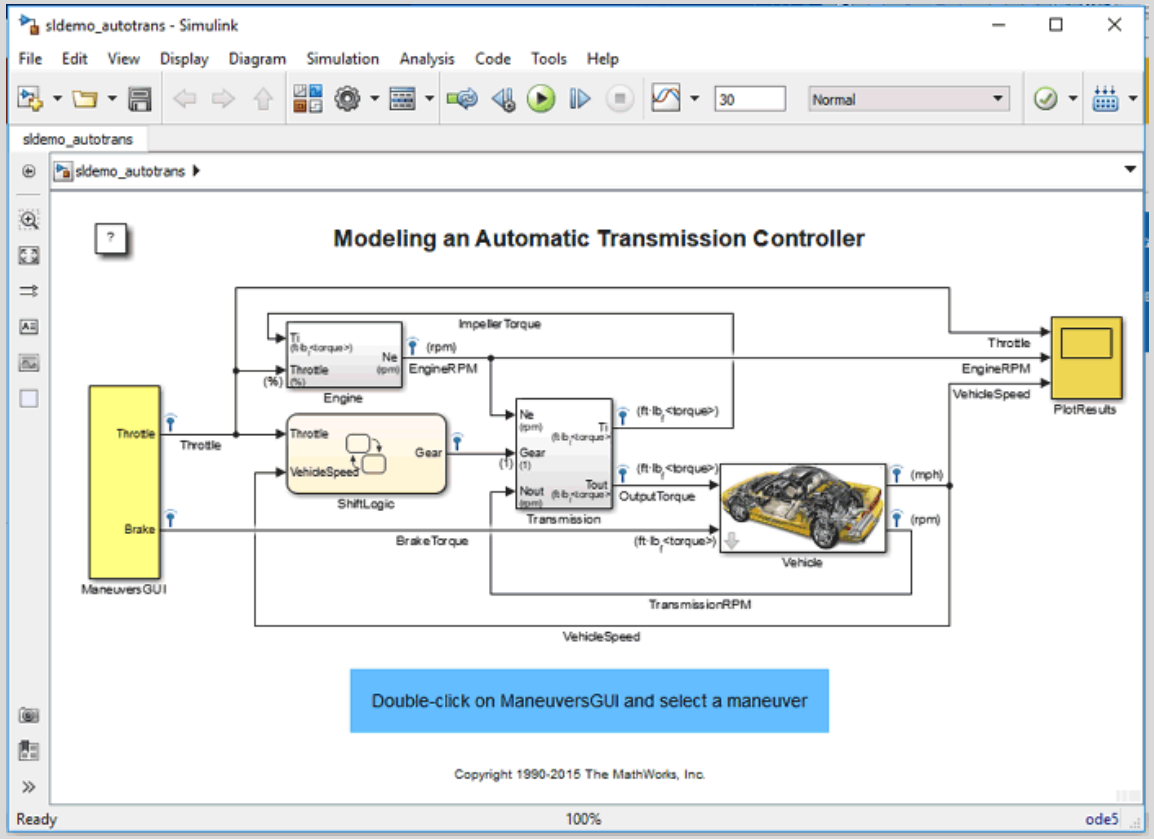
- Block diagram editing
- Non-linear simulation
- Hybrid (continuous and discrete) models
- Asynchronous (non-uniform sampling) simulation
- Fully integrated with MATLAB → MATLAB  toolboxes and block sets



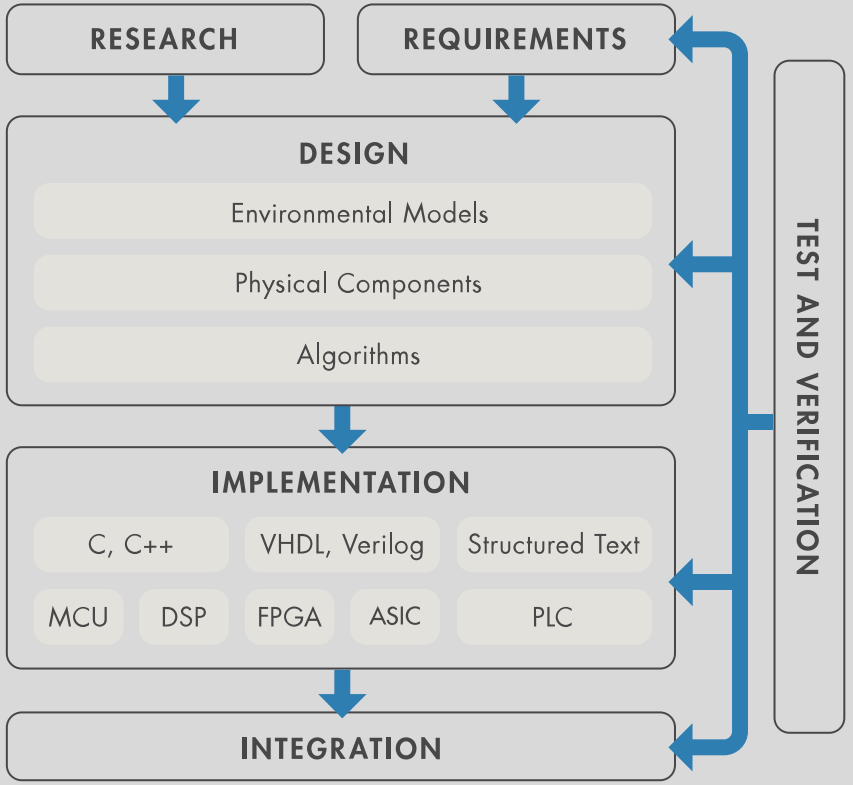
# Simulink

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

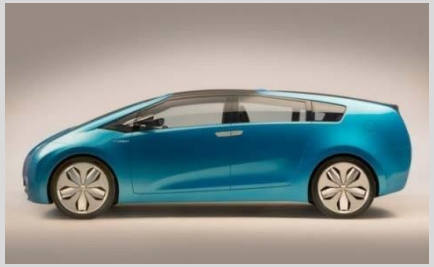


# Model Based Design with Simulink

- **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

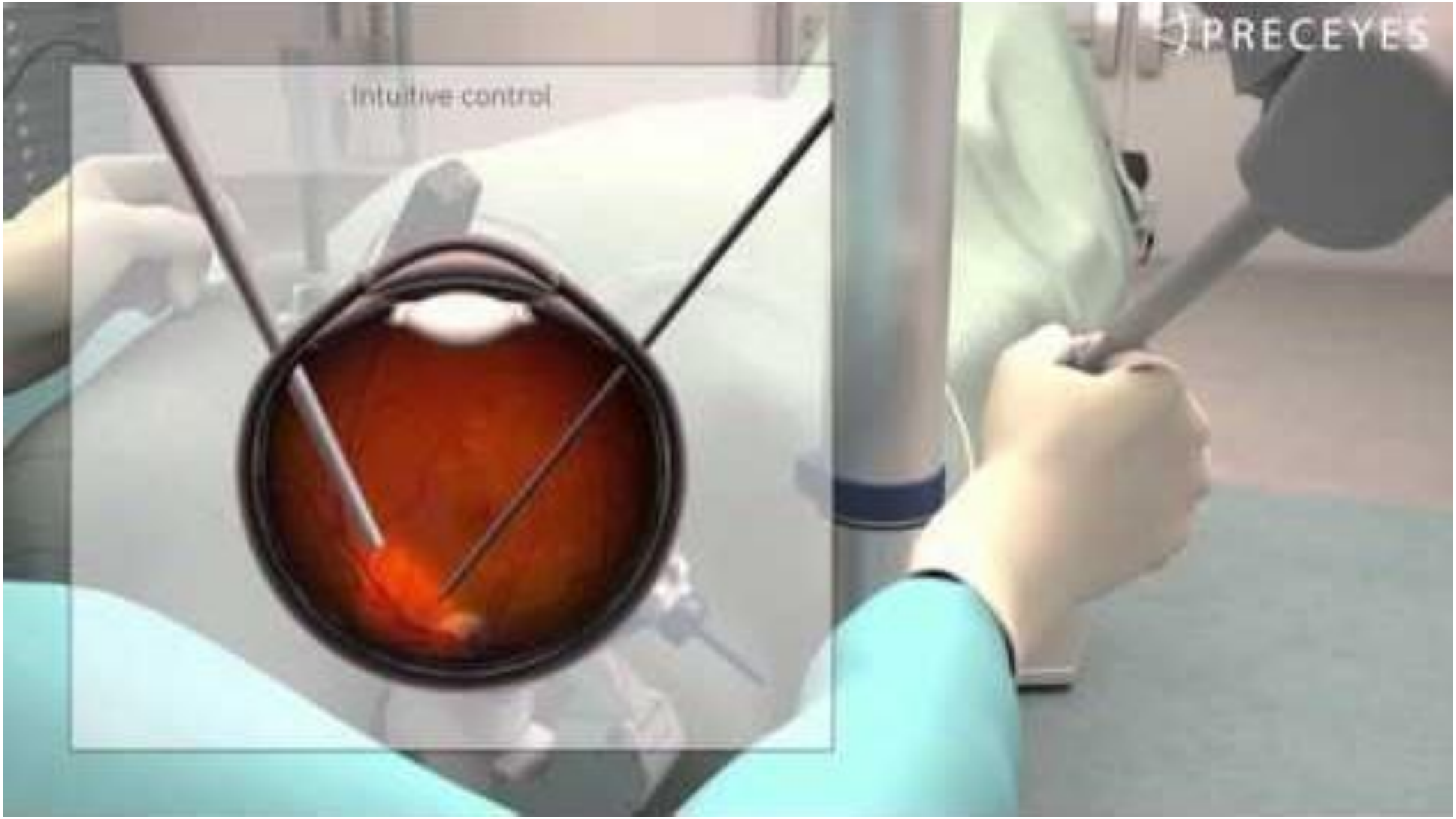
Use Model-Based Design with MATLAB, Simulink, and Real-Time Workshop software to model, simulate, test, and verify designs

## 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 we ran 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

## Challenge

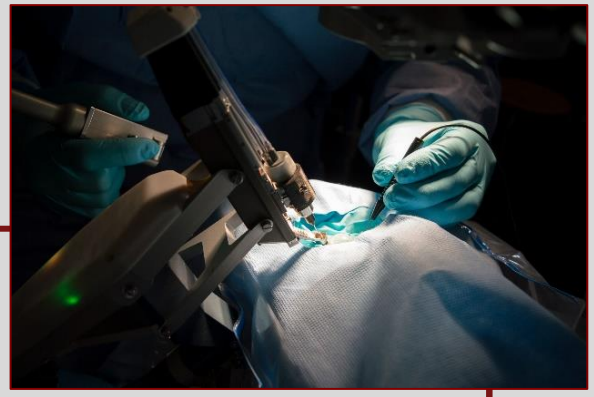
Develop a real-time control system for robot-assisted surgical procedures performed within the human eye

## 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



The PRECEYES Surgical System

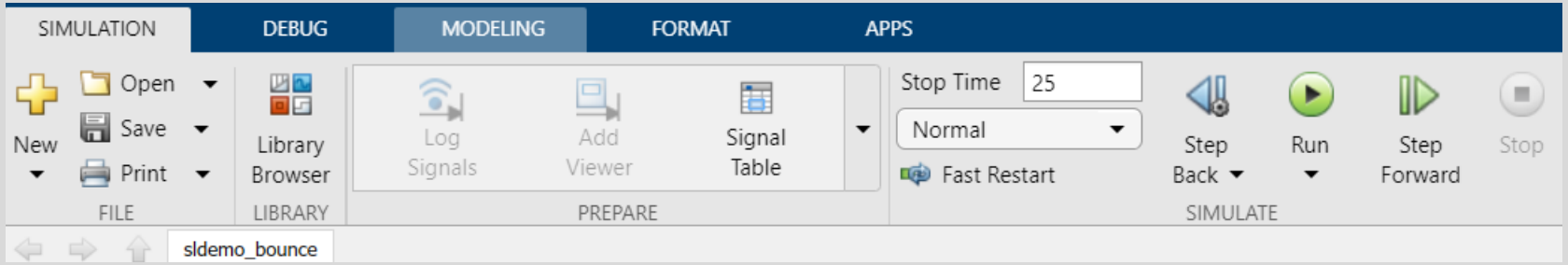
**"MATLAB and Simulink provided a single platform that supported our complete workflow and all the components and protocols we needed for our robotic system. That enabled us to quickly develop a safe, real-time device, ready for clinical investigation."**

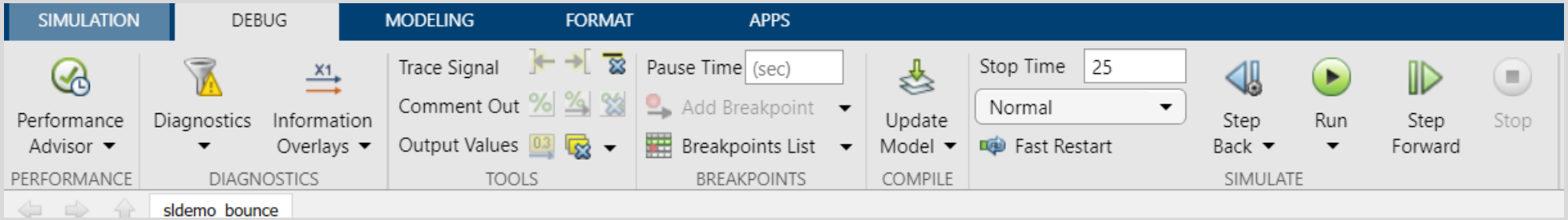
**Maarten Beelen, PRECEYES**

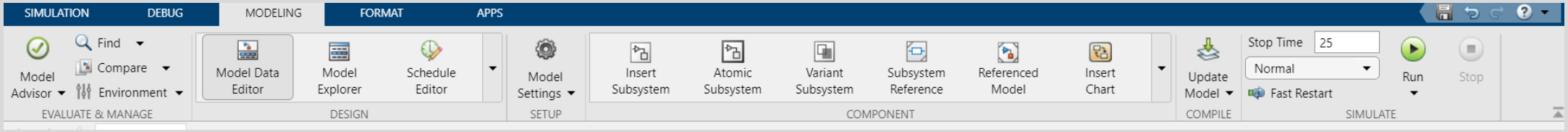
# Outline

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

# Simulink Toolbar



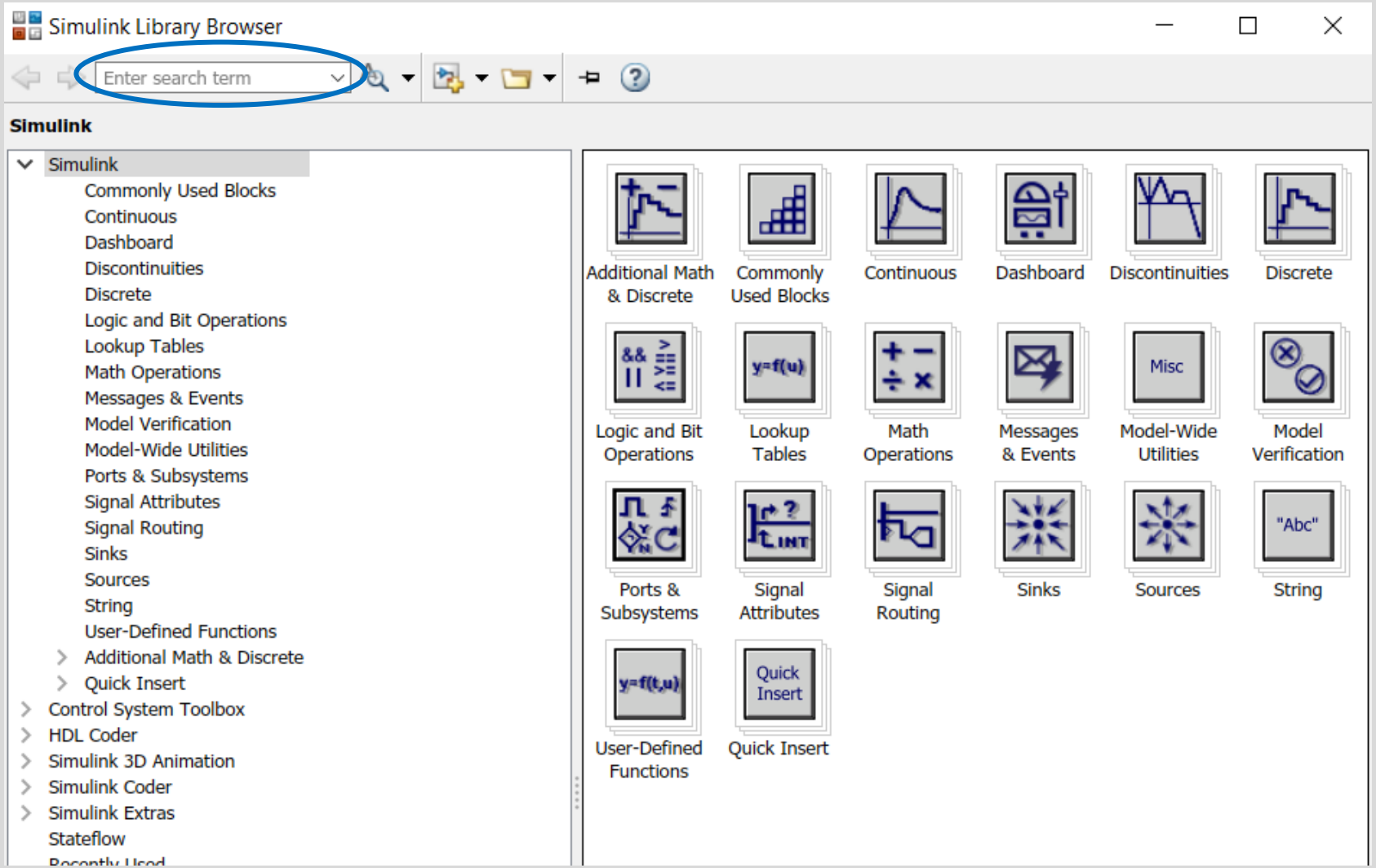



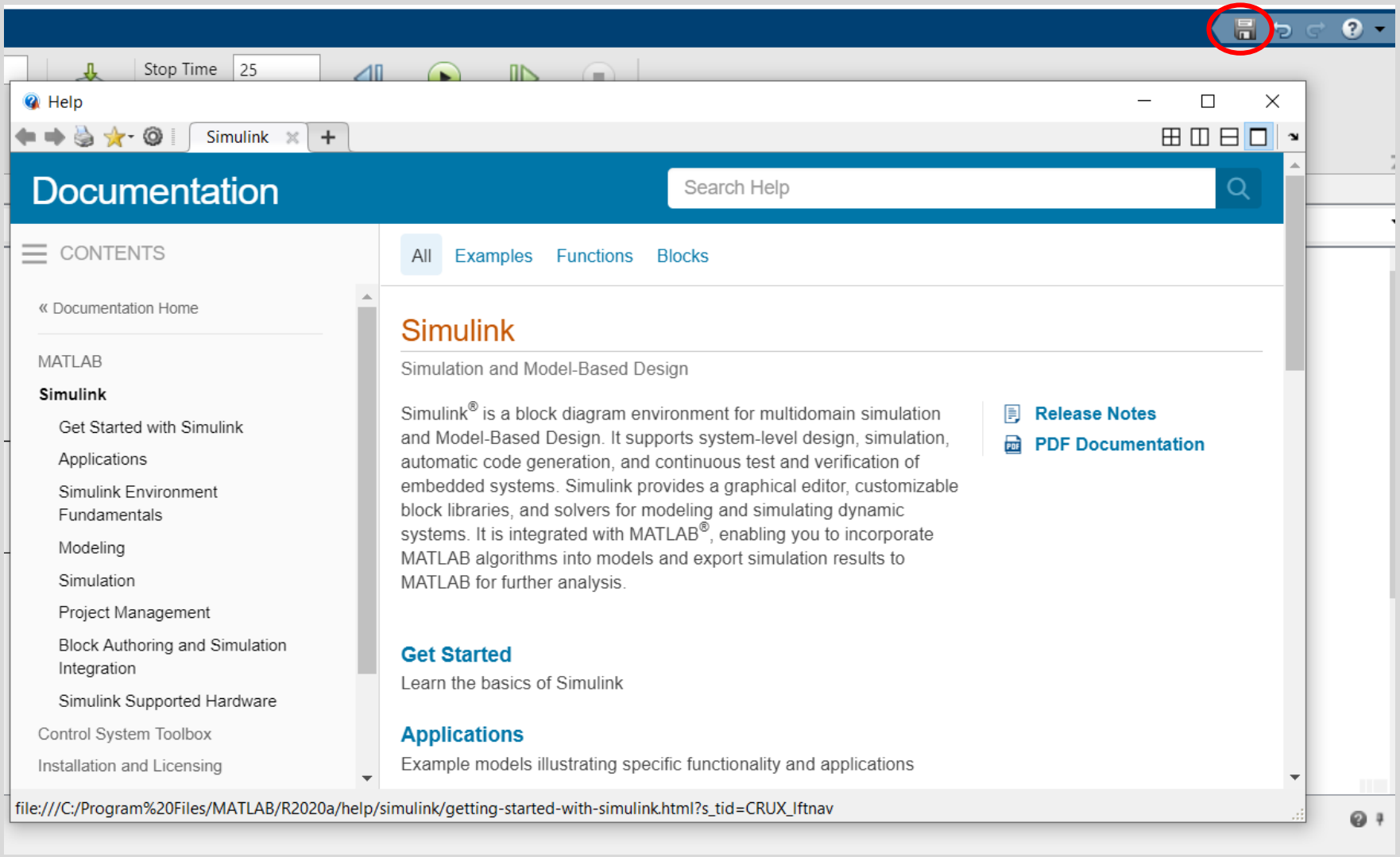
# Overview of the library

| 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 |
| Continouse | Integrator, Derivative |
| Discrete | Unit delay, Discrete Derivative |
| Logical and Bit operations | Compare to Zero, Logical operators |

# Finding Blocks



# Getting Help

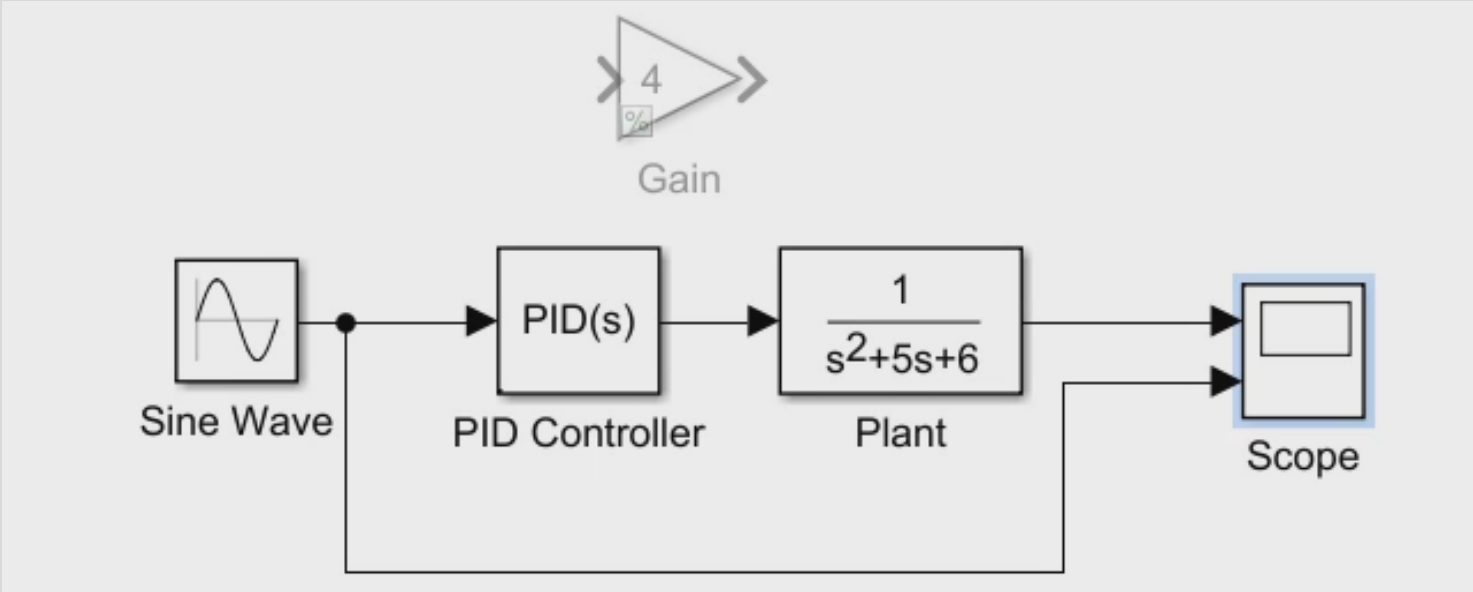


# Outline

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

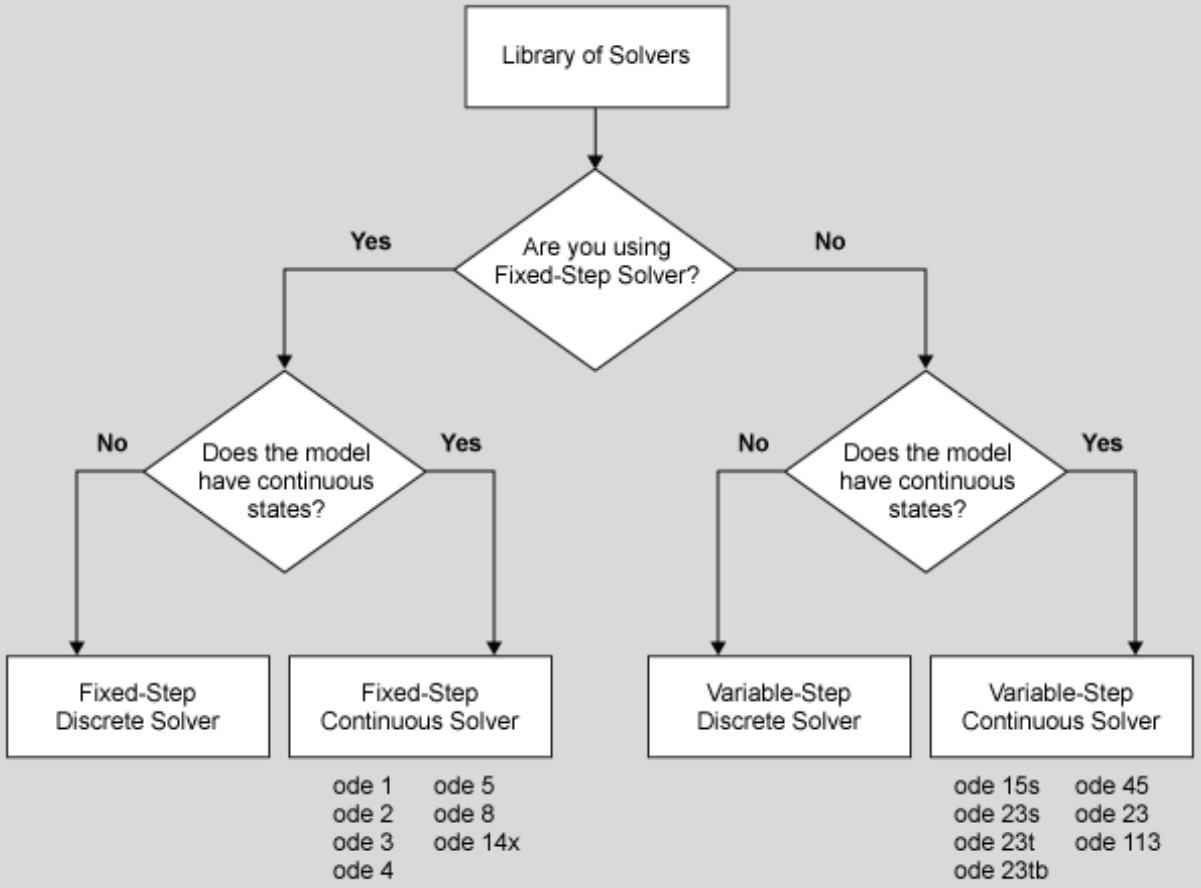
# How Simulink Works

- 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



# Outline

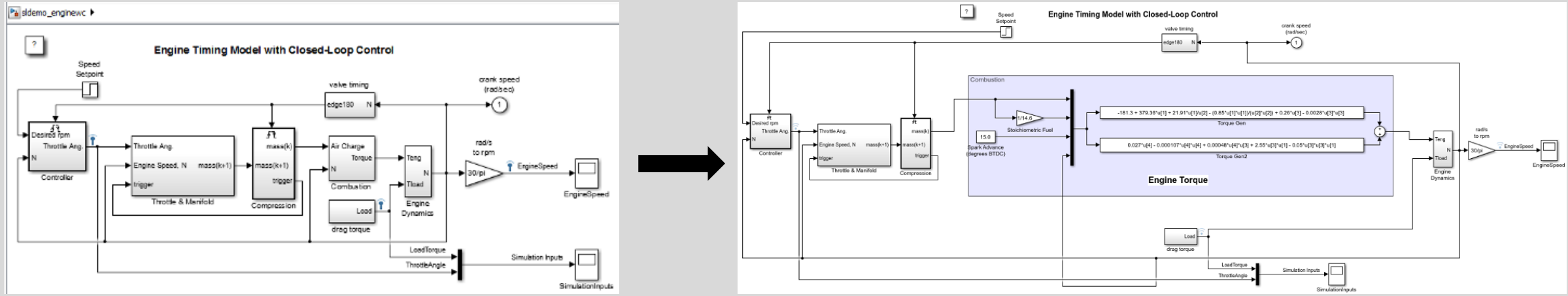
- Why Simulink?
- Working with Simulink
- How Simulink works
- **Decoupling models**
- Continuous and discrete models

# 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 a single Subsystem 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.

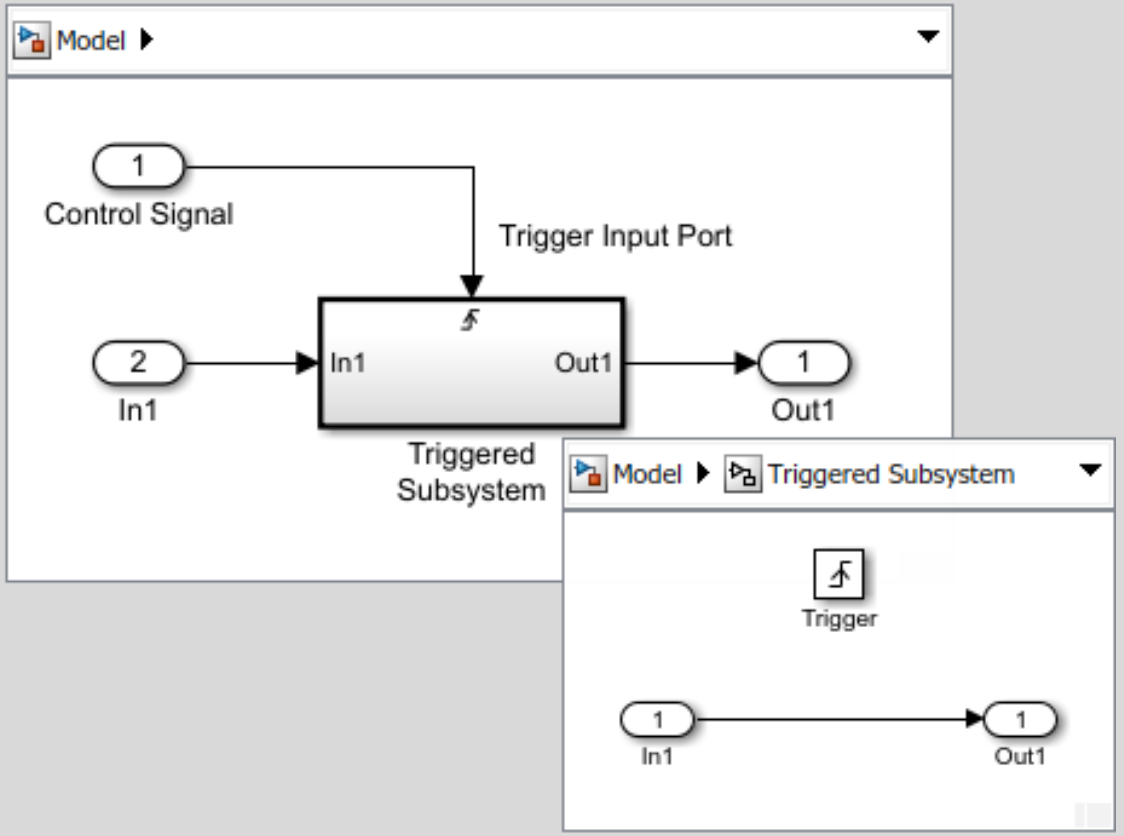
# Creating Subsystems

- **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

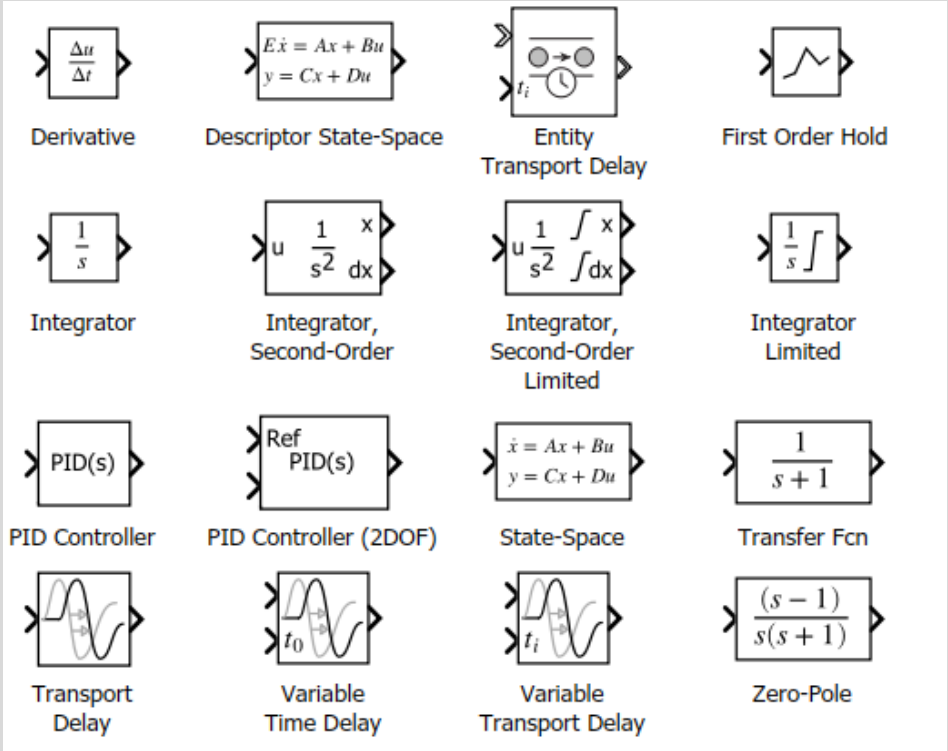


# Outline

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

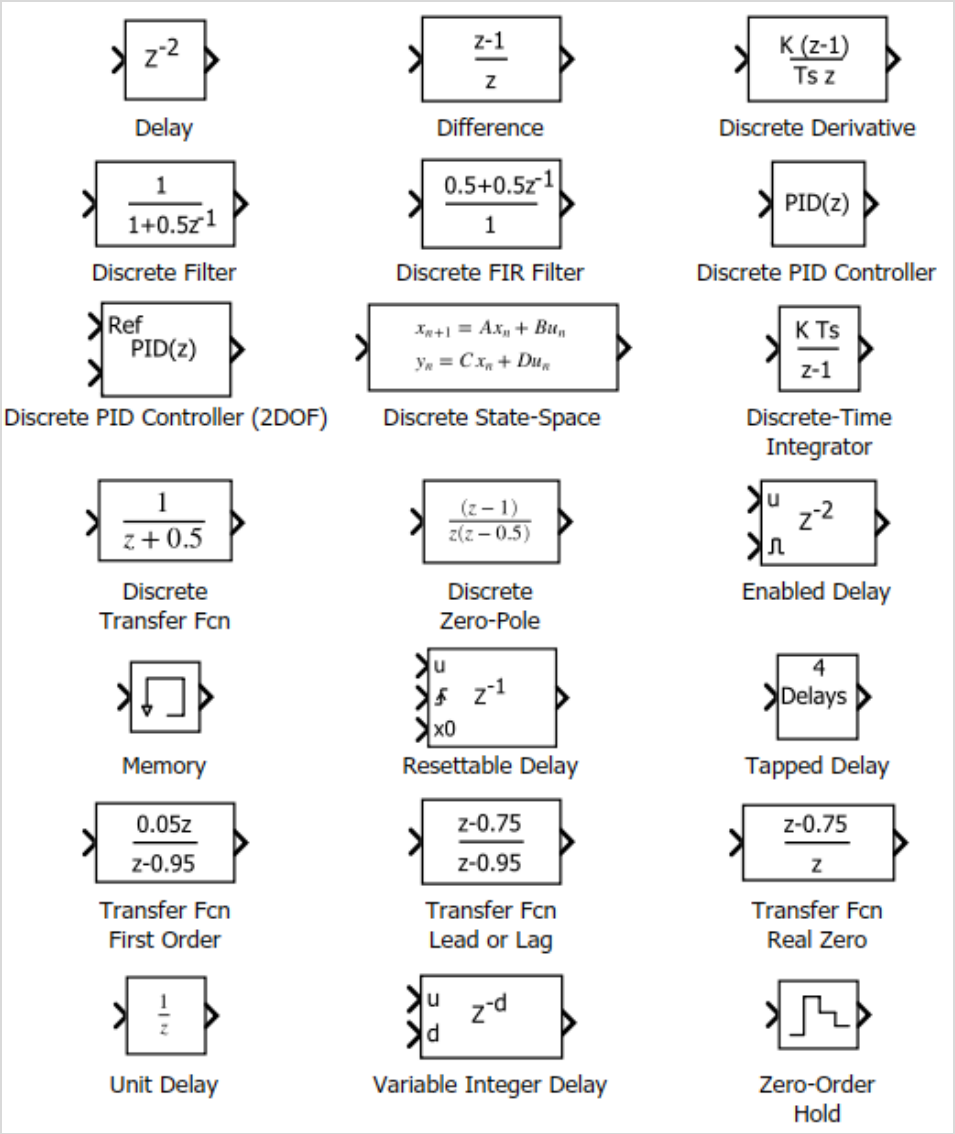
# ‘Continuous’ Library

- 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.



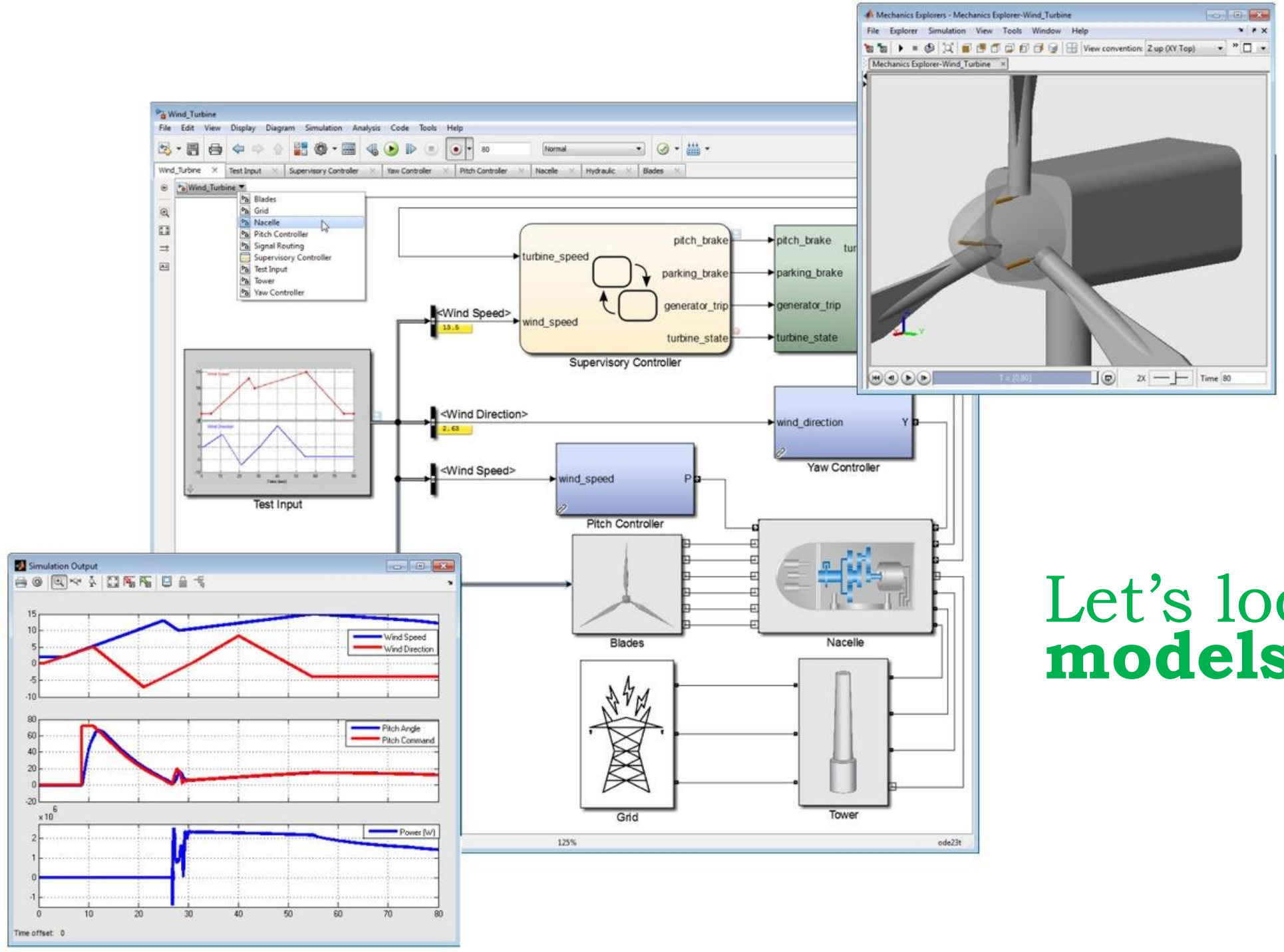
# ‘Discrete’ Library

- 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!