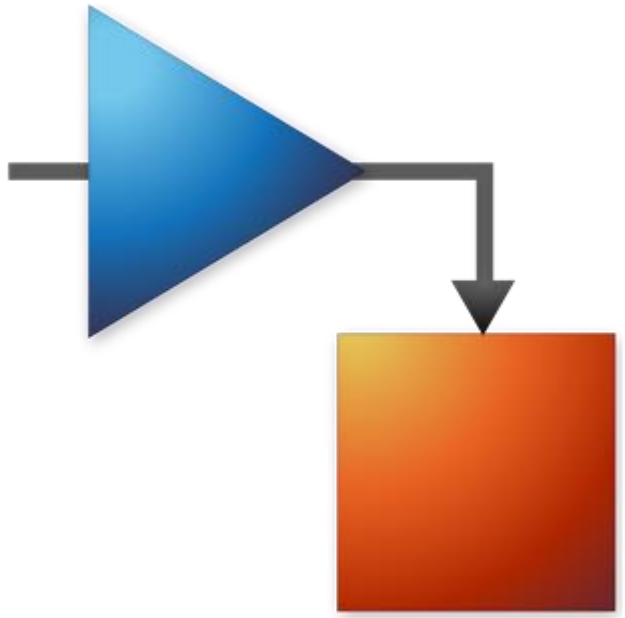
Introduction to Simulink

Mughees Asif
3rd Year Aerospace Engineering
QMUL MathWorks Student Ambassador



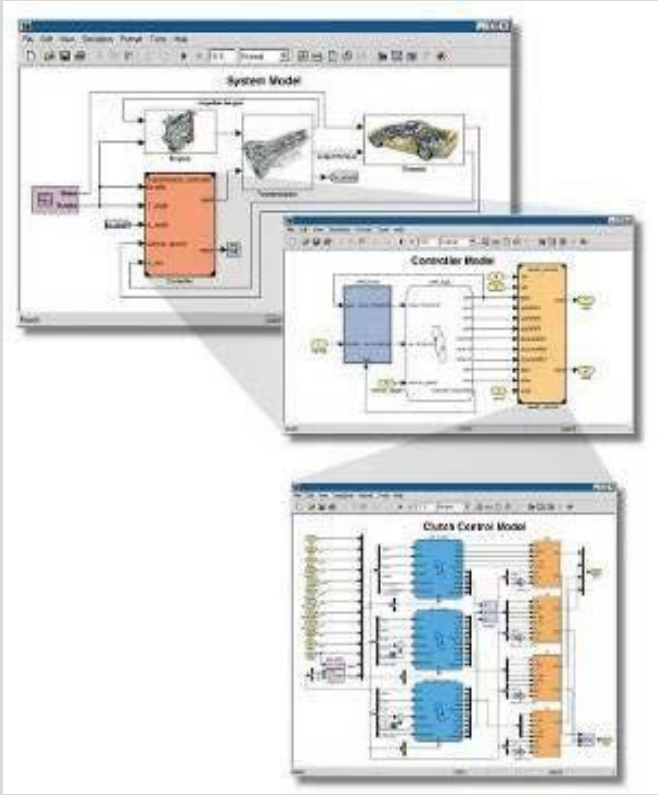
Outline

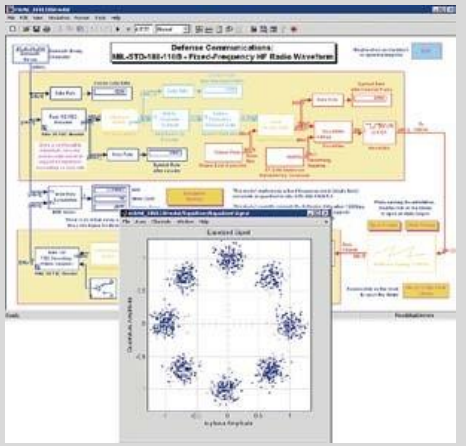
- **What is Simulink?**
- Working with Simulink
- How Simulink works
- Componentizing models
- Continuous and discrete models

Simulink

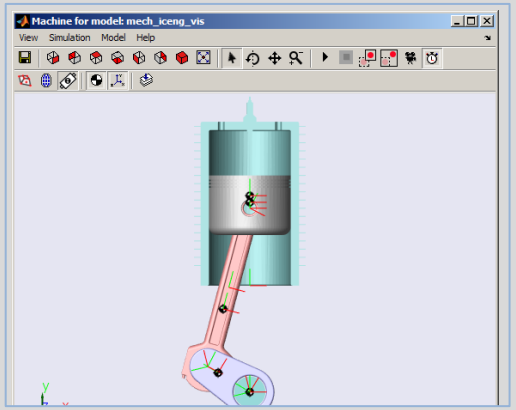
Simulink is a software package for modeling, simulating, and analyzing dynamic systems

- Block diagram editing
- Non-linear simulation
- Hybrid (continuous and discrete) models
- Asynchronous (non-uniform sampling) simulation
- Fully integrated with MATLAB, MATLAB toolboxes and blocksets.

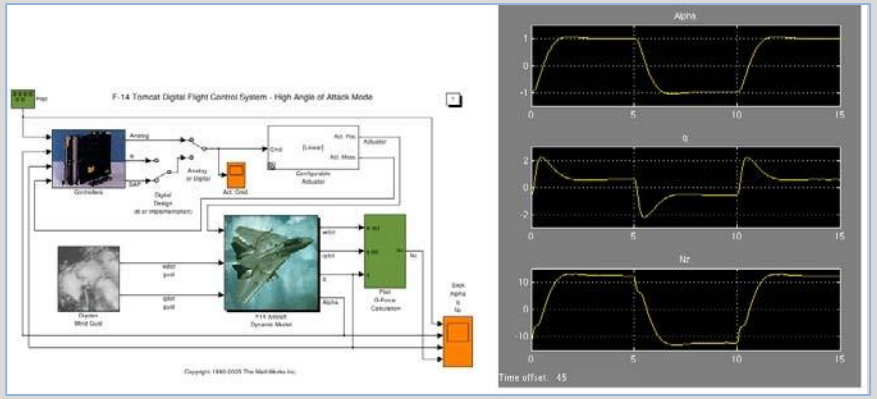


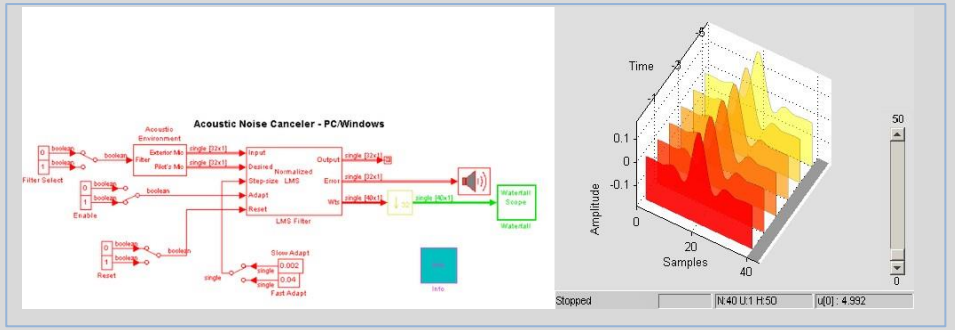


Simulink



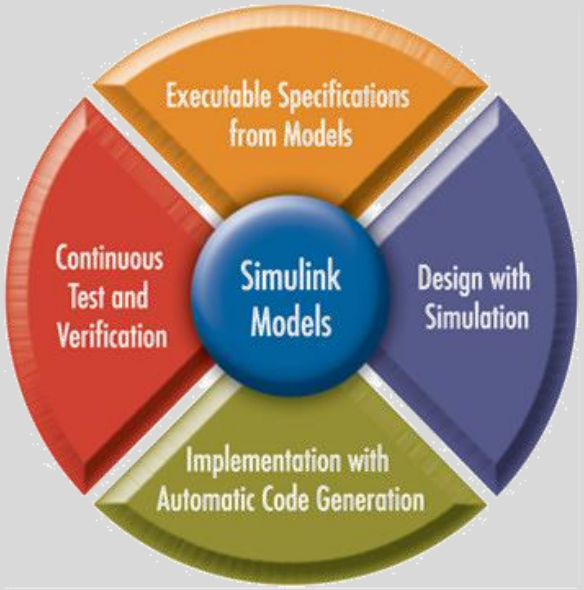
- Accurately design, implement, and test:
  - Control systems
  - Signal Processing systems
  - Communications systems
  - Embedded systems
  - Physical systems
  - other Dynamical systems



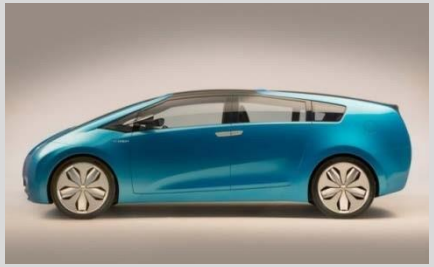


Model Based Design with Simulink

- Modeling and simulation
  - Multidomain Dynamic Systems
  - Nonlinear Systems
  - Continuous-time , Discrete-time, Multi-Rate systems
- Plant and Controller Design
  - Rapidly model what-if scenarios
  - Communicate design ideas
  - Embody performance specifications
  - Select/Optimize control architecture and parameters
- 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

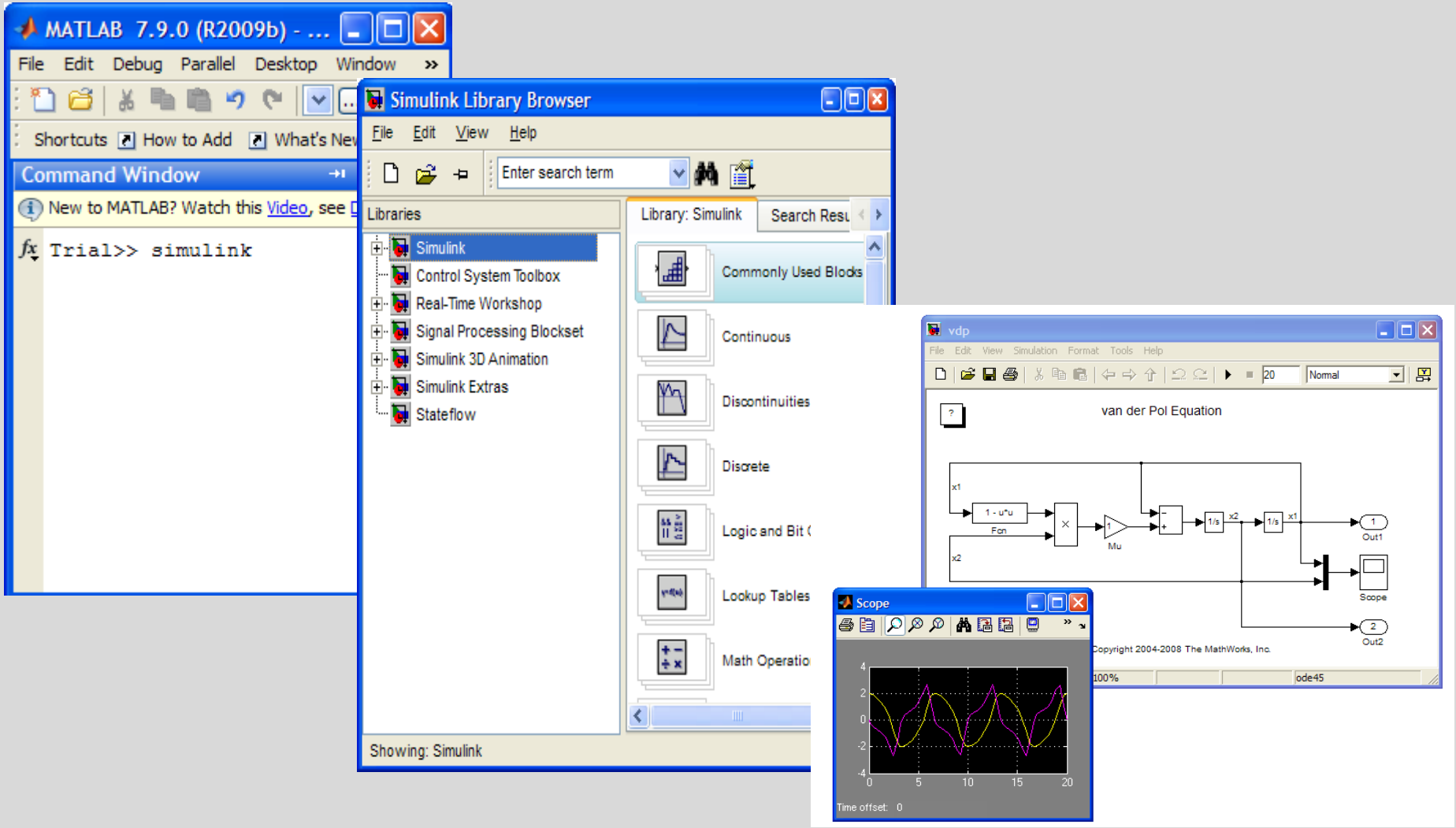
"Simulations and a rapid, iterative approach enabled us to minimize the unknowns and ensure that we had established enough margin that when we ran into a surprise we could continue to have a safe flight test program—and run it with unprecedented efficiency."

David King
Bell Helicopter

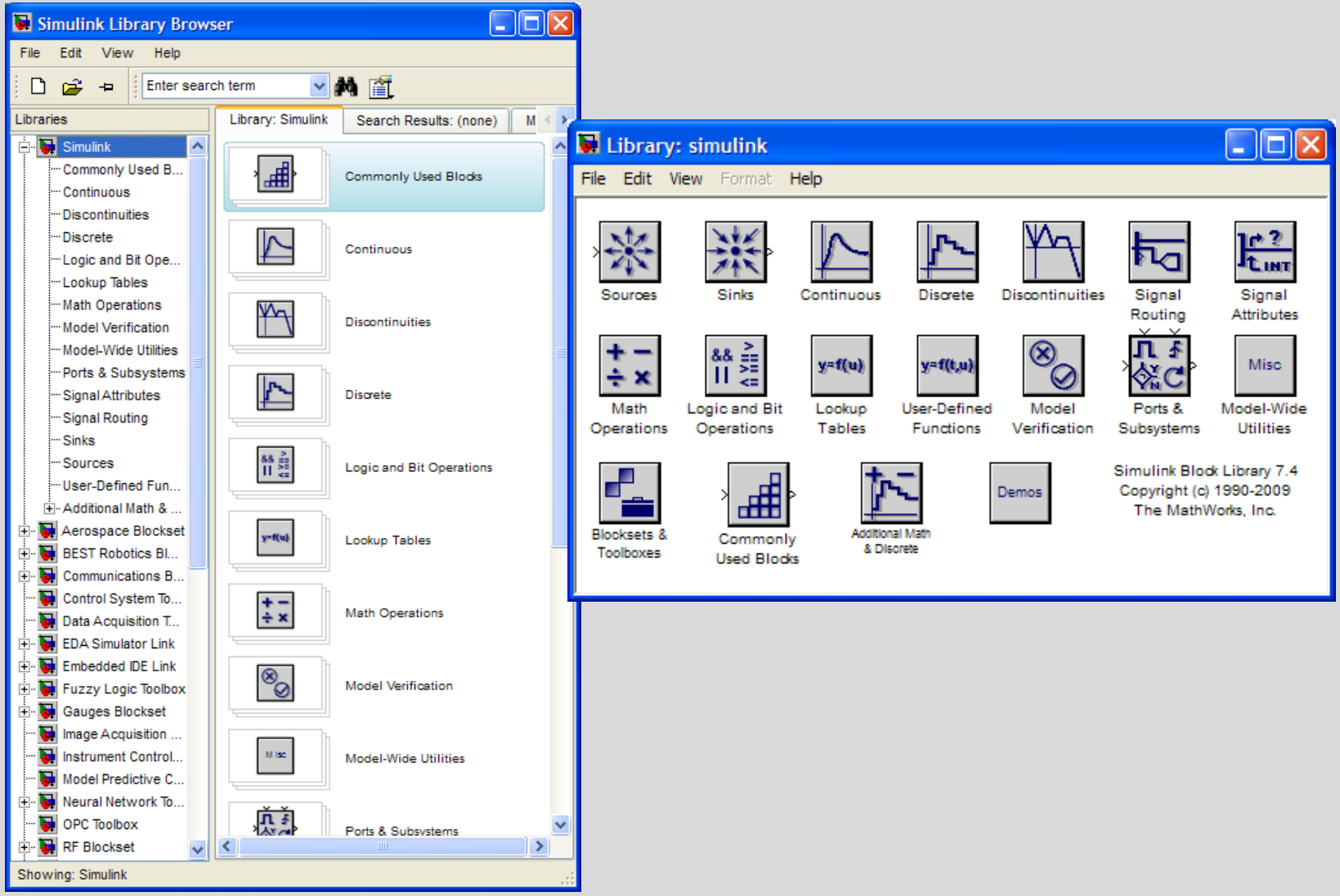
Outline

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

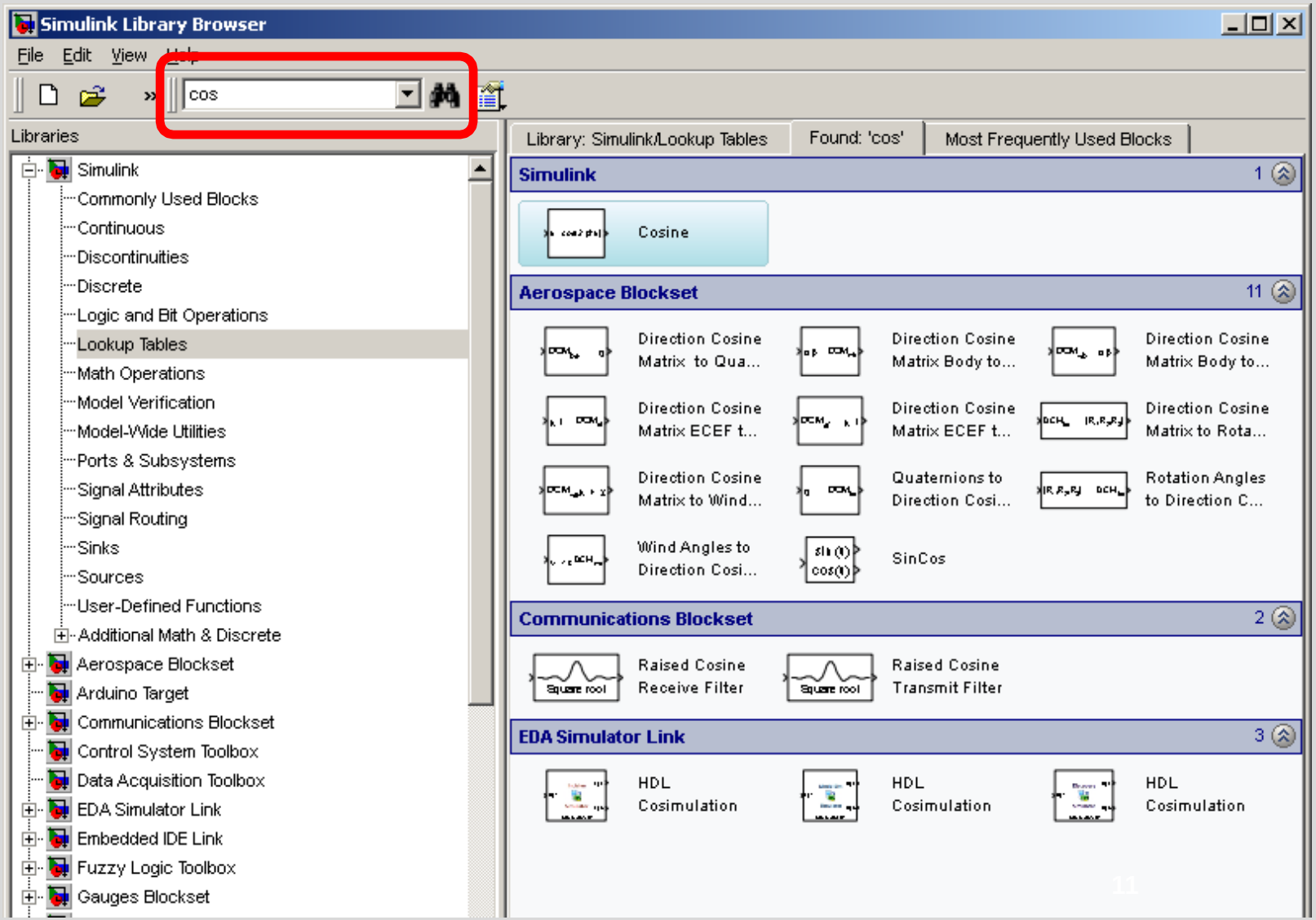
Launching Simulink



Simulink Library Browser

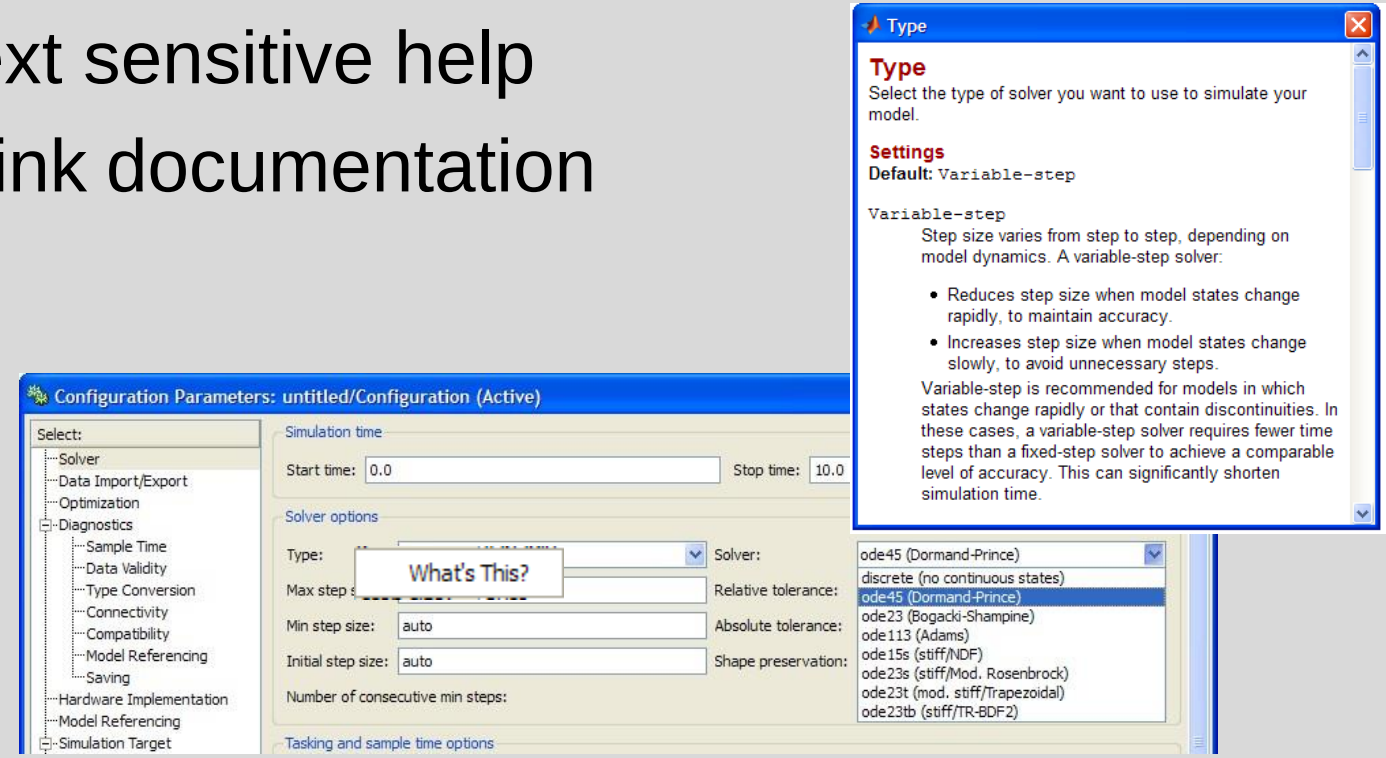


Finding Blocks



Getting Help

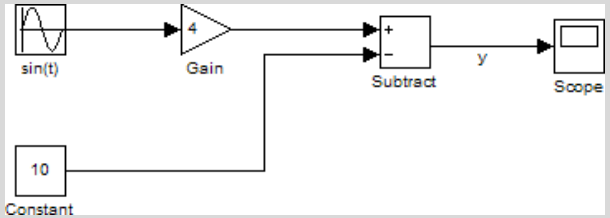
- Context sensitive help
- Simulink documentation

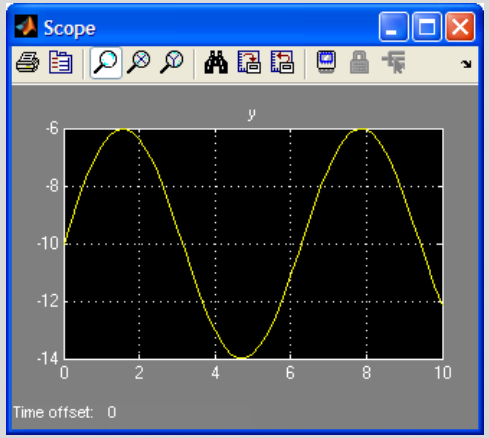


Demo

- Working with a simple model
- Changing block parameters
- Labeling blocks and signals
- Running a simulation
- Defining parameters with MATLAB variables
- Saving/opening a model

$$y = 4 * \sin(t) - 10$$



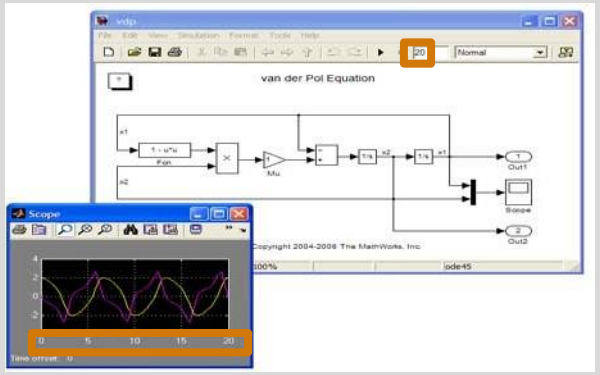


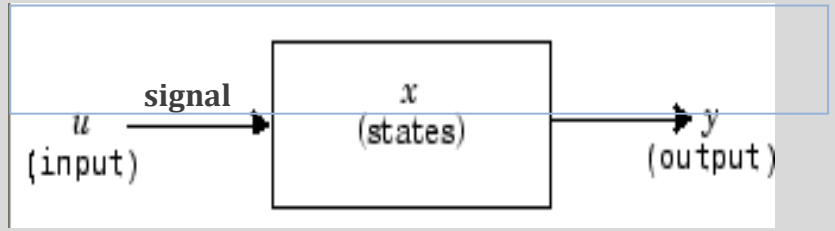
Outline

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

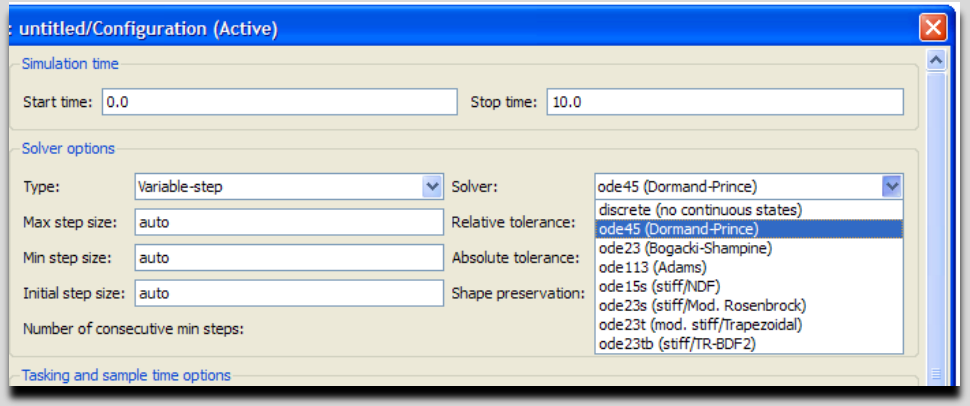
How Simulink Works

- 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
- Solver options:

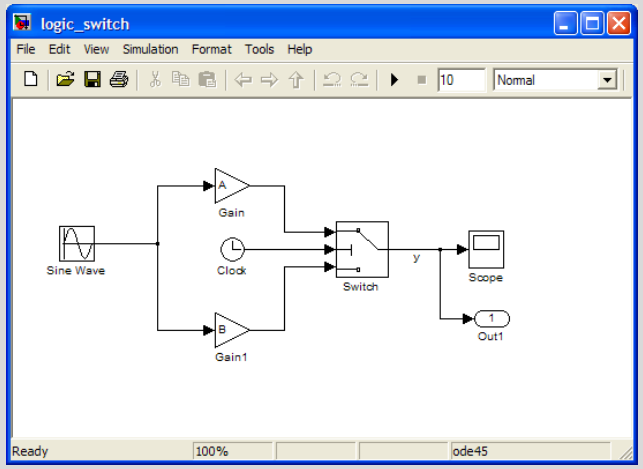
Fixed-Step	Variable-Step
✓Ode1	✓Ode45
✓Ode2	✓Ode23
✓Ode3	✓Ode113
✓Ode4	✓Ode15s
✓Ode5	✓Ode23s
✓Ode8	✓Ode23t
	✓Ode23tb

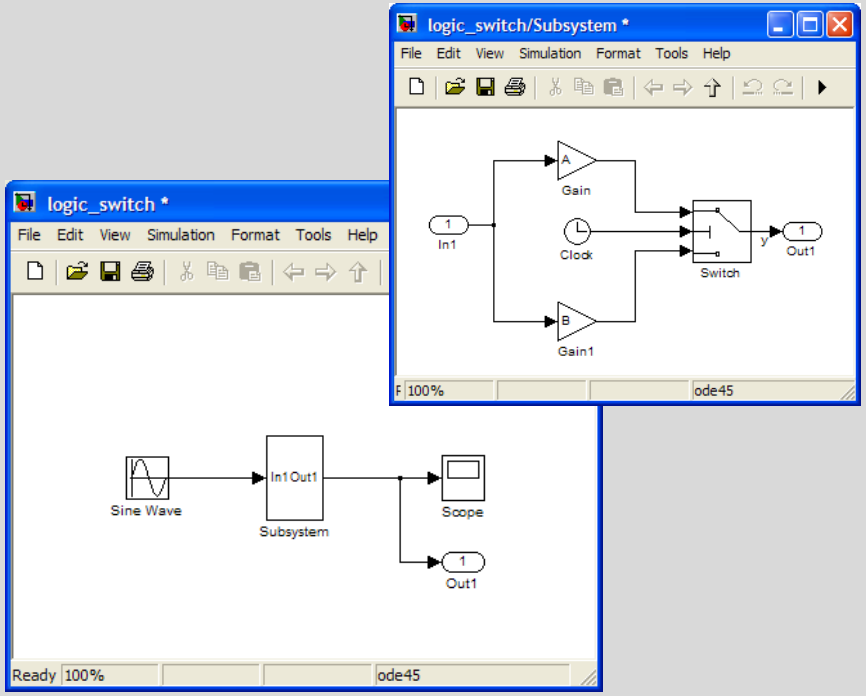
Outline

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

Creating Subsystem

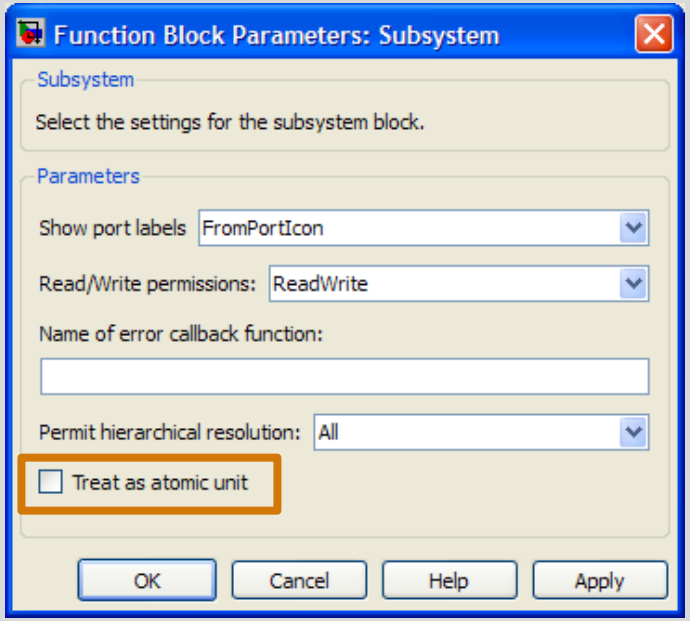
- Context menu → Create Subsystem
- Subsystem ports
- Inside a subsystem





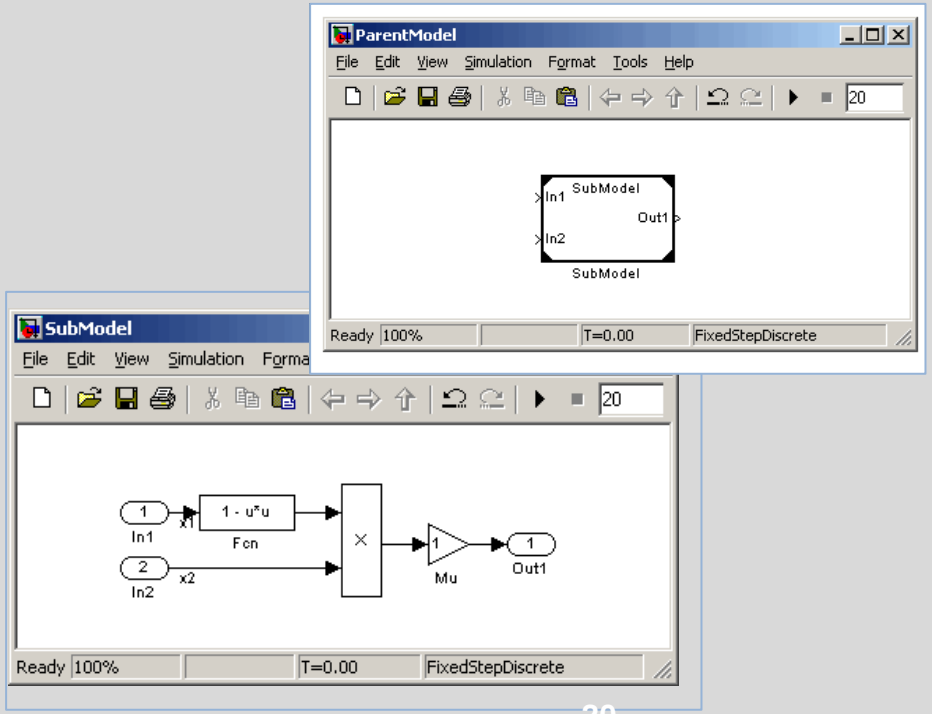
Subsystems

- Why?
  - Reduce blocks displayed in a model window
  - Keep functionally related block together
  - Establish hierarchical block diagram



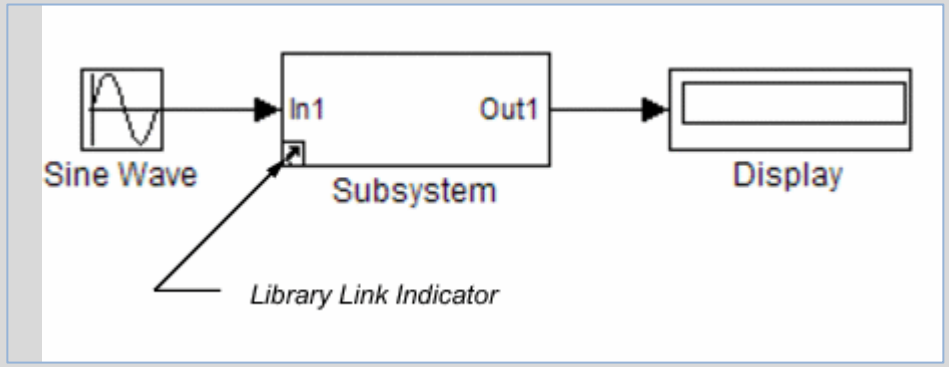
Model Referencing

- One model in another- *parent and referenced model*
- Advantages:
  - Componentization/Modularization
  - IP protection
  - Multiple referencing
  - Acceleration



Block Library

- Collection of blocks
- Prototype block vs Reference block
- Library Links
  - Disable link
  - Restore link
  - Break link

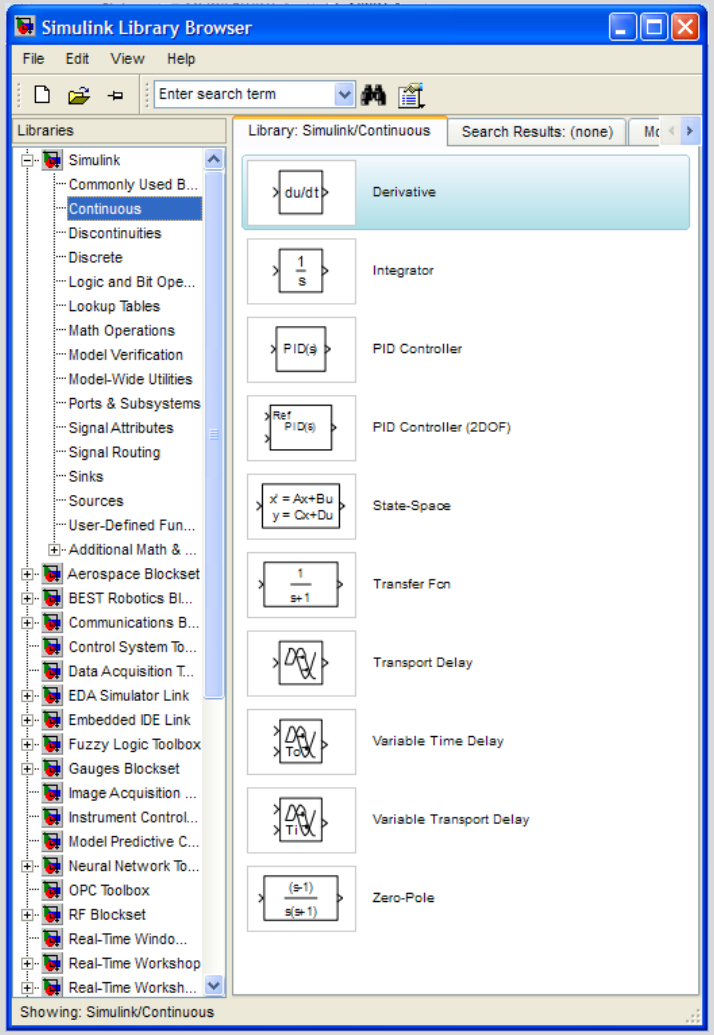


- Other features
  - Display in Simulink Library Browser
  - Add documentation

Outline

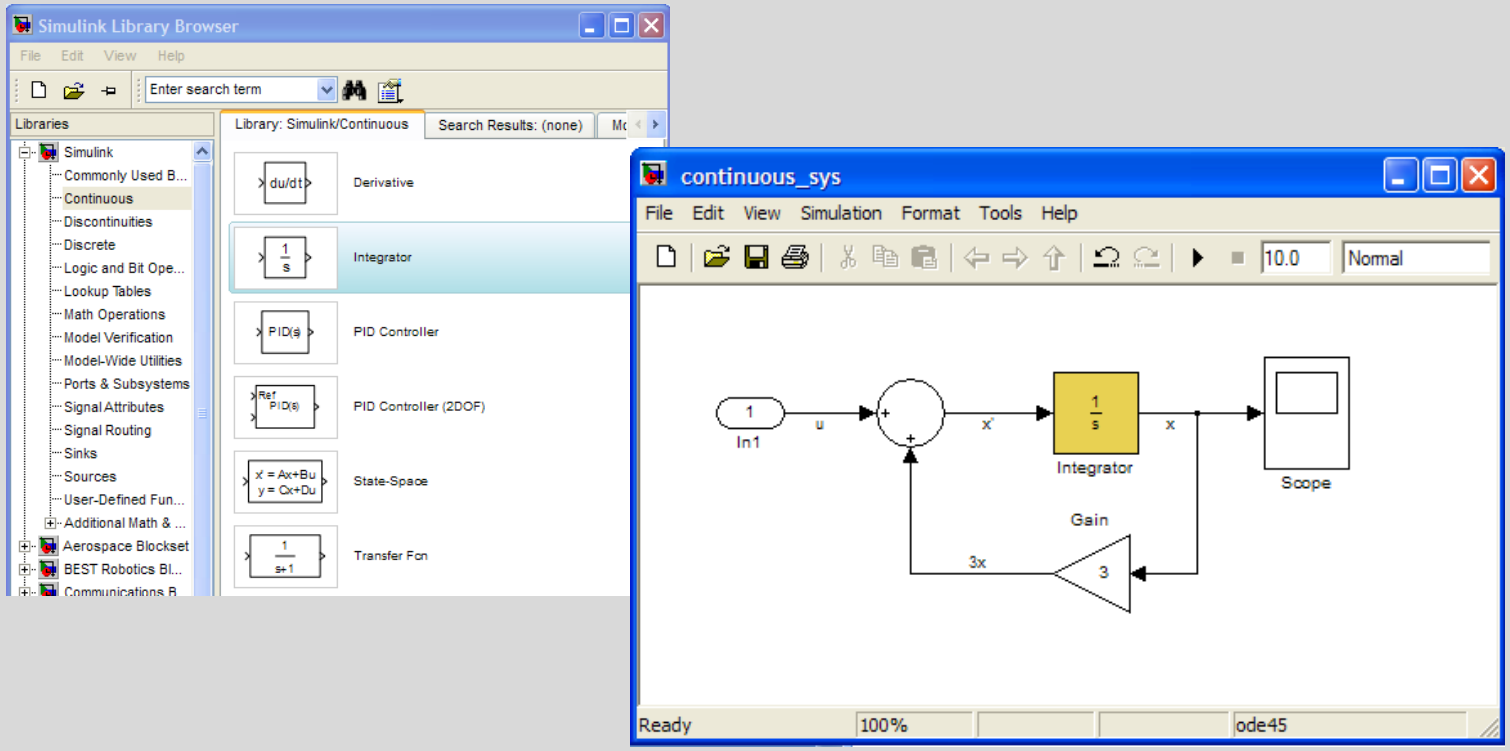
- Why Simulink?
- Working with Simulink
- How Simulink works
- Componentizing models
- **Continuous and discrete models**

‘Continuous’ Library



Continuous systems: Time-Domain Representation using Integrator Block

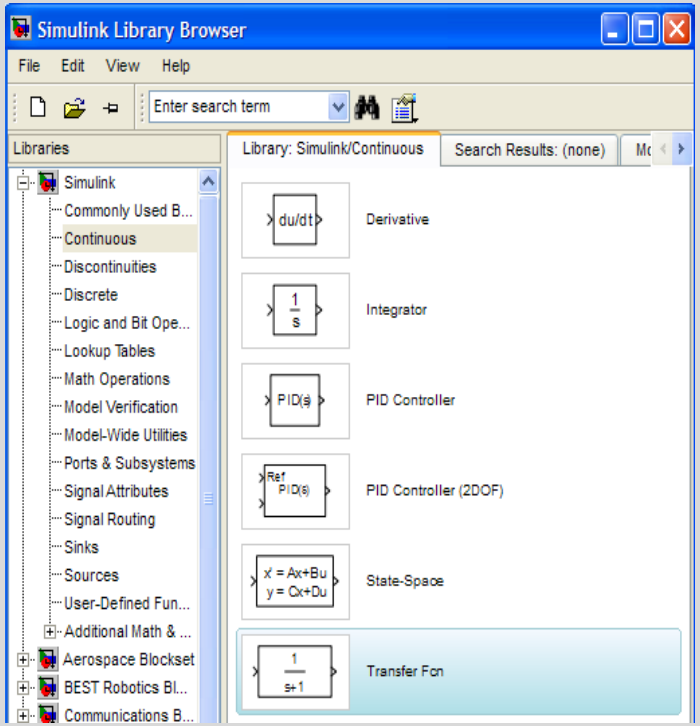
$$x'(t) = 3x(t) + u(t)$$

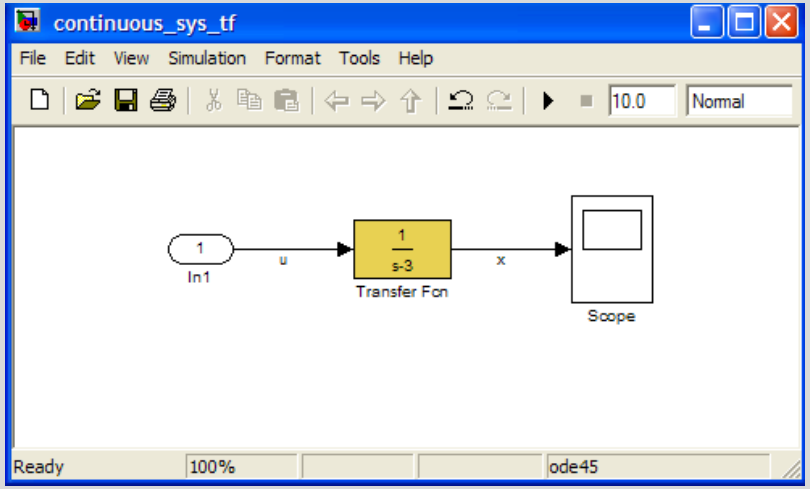


Continuous systems: Frequency-Domain Representation using Transfer Function Block

$$x'(t) = 3x(t) + u(t) \Leftrightarrow \frac{X(s)}{U(s)} = \frac{1}{s-3}$$

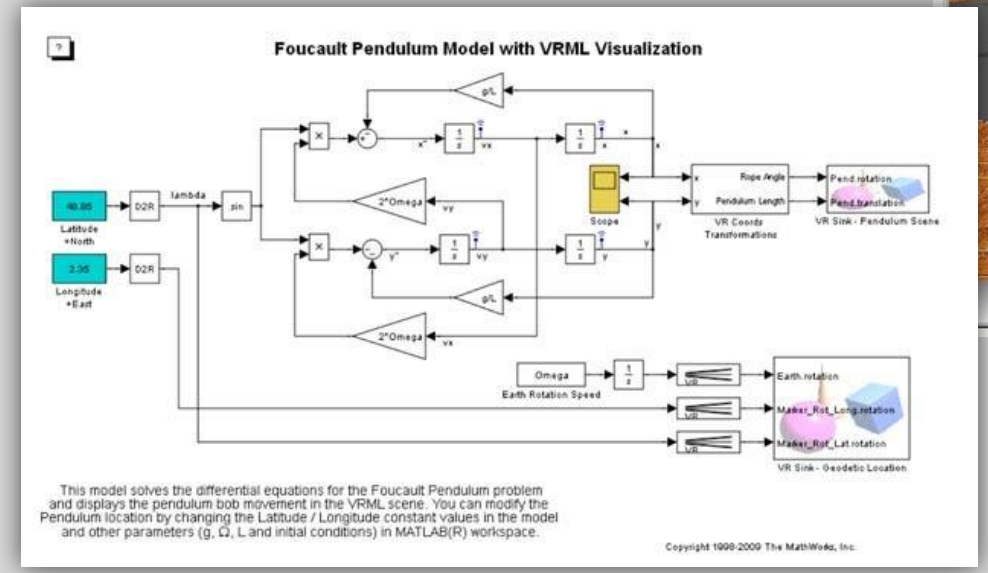
```
Transfer function:
sx = 3x + u
x = u/(s-3)
x/u = 1/(s-3)
```



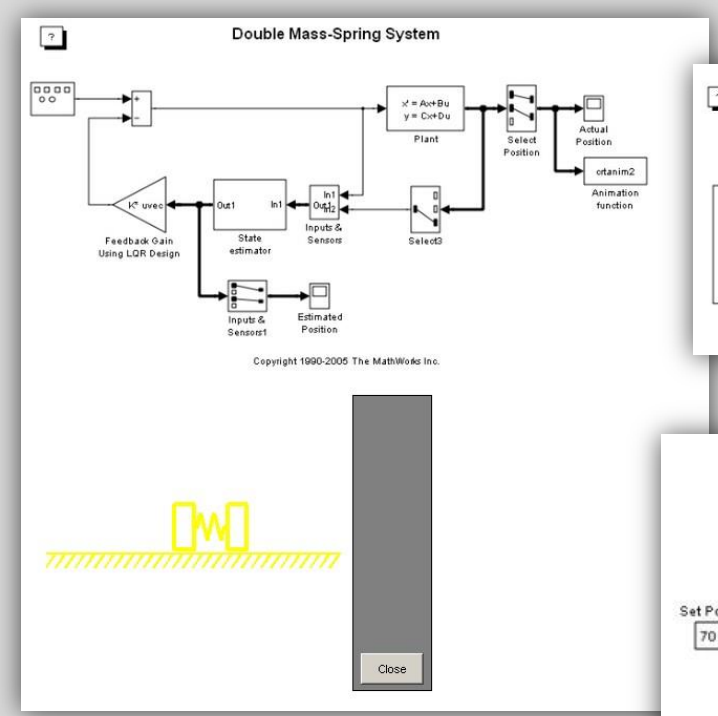


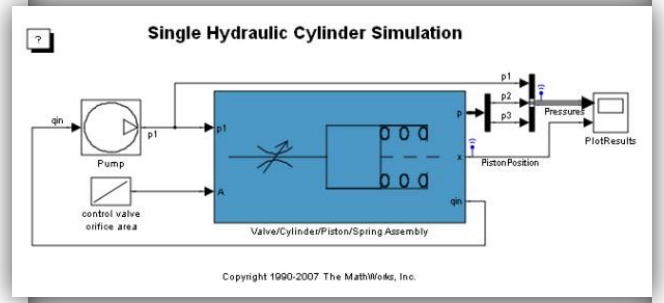
Simulink Demo: Foucault Pendulum Model with VRML Visualization

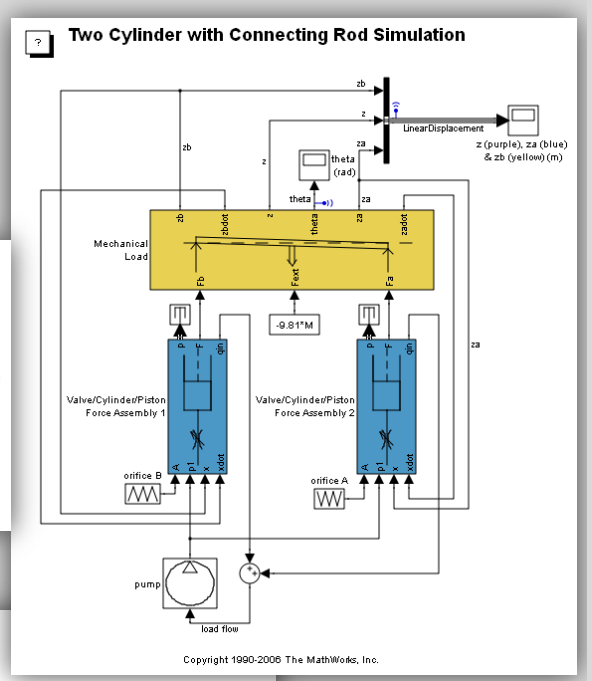
`sldemo_foucault_vr.mdl`

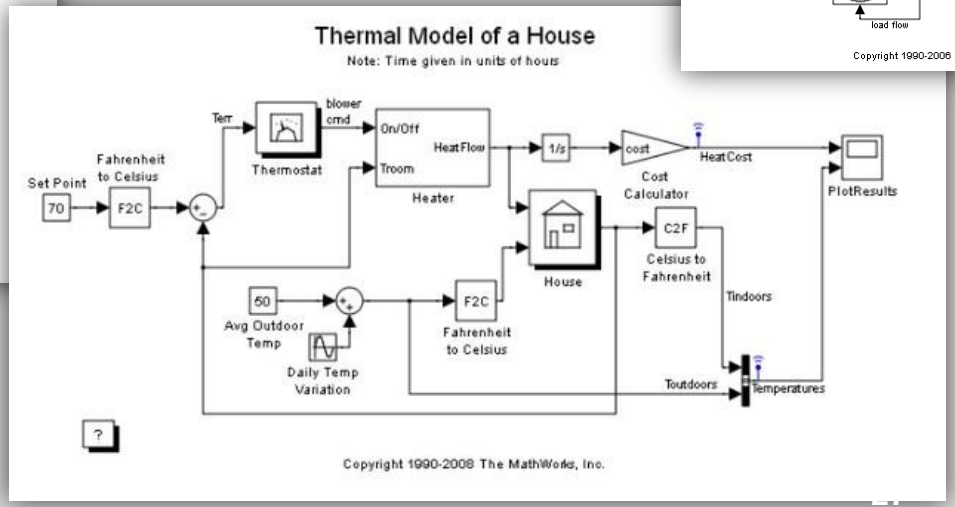


Other Demos for Continuous Systems





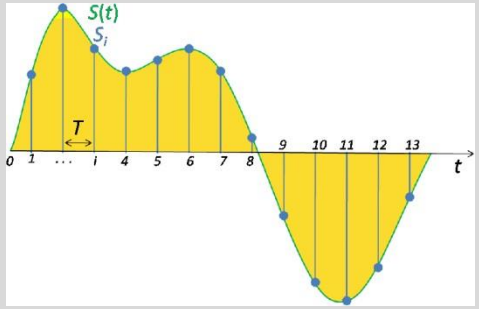


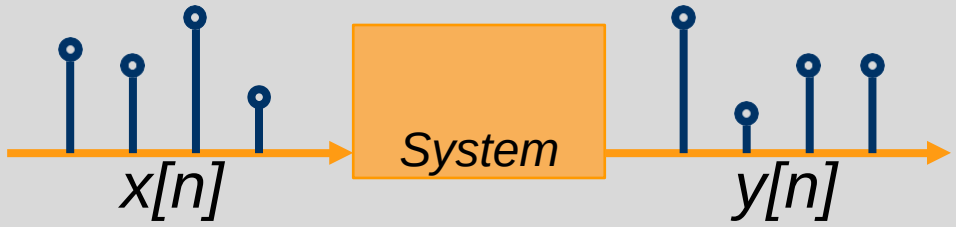


Discrete Systems

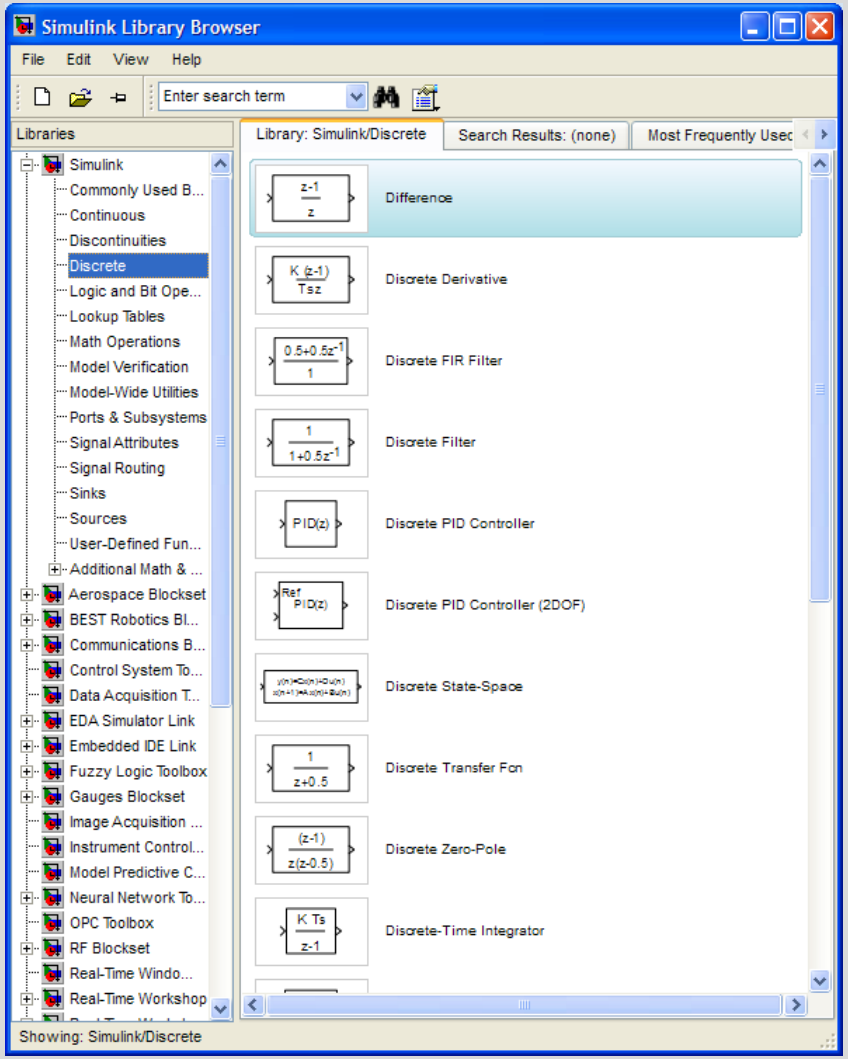
- System that takes an input sequence of samples and outputs a sequence of samples
- Sampling

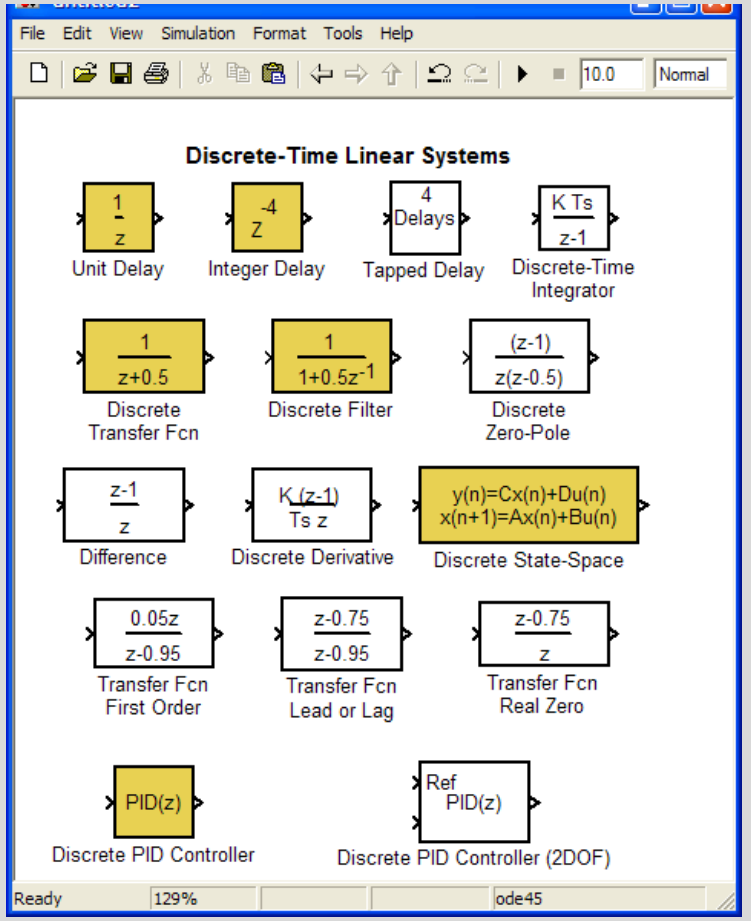
$$y[k] = 0.1x[k] + x[k-1]$$
$$x[k+1] = -0.5x[k] + u[k]$$





‘Discrete’ Library

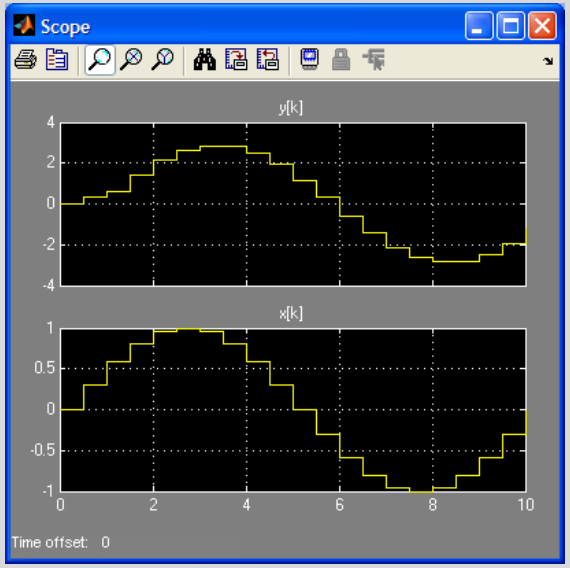




Discrete system example

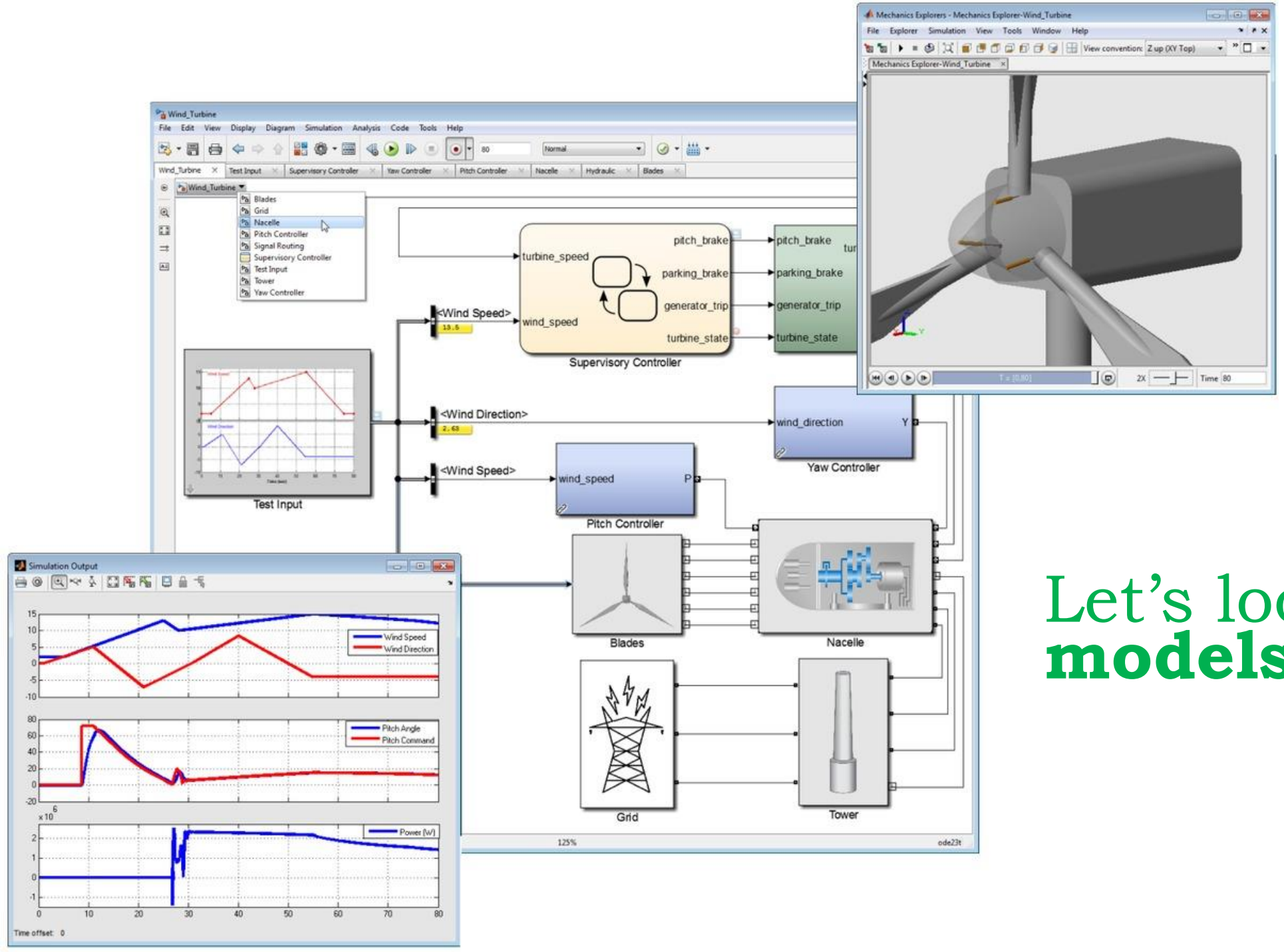
- Second order FIR filter

$$y[k] = \frac{x[k] + ax[k-2]}{2}$$



Summary

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



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

Cisco

- **Campus Ambassador**
- Do join the **LinkedIn page** for graduate/internship opportunities: https://www.linkedin.com/groups/12470630/
- Join the **email newsletter** to stay up-to-date: https://forms.gle/QZamk91qoJm3pt4s6
- To **access** the job opportunities, use the link: https://rb.gy/y8u16a

Developer Student Clubs

Queen Mary, University of London

- Work in a team of 17 students, in collaboration with Google
- Leveraging Google products to solve local community challenges
- Do join the **LinkedIn page** to get involved: https://www.linkedin.com/groups/12467711/
- Check out the team and for future updates, do **join the chapter**: https://dsc.community.dev/queen-mary-university-of-london/

KAHOOT! Challenge

Will post all prizes by next day

All sizes available

First	1 x ***t-shirt*** 1 x ***baseball cap*** 1 x ***sunglasses*** 10 x ***pens***
Second	1 x ***drawstring bag*** 1 x ***baseball cap*** 10 x ***pens*** 20 x ***stickers***
Third	10 x ***pens*** 10 x ***stickers***

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#queen-mary-matlab-tutorials