Introduction to Simulink

Todd Atkins
tatkins@mathworks.com
Outline

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

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

• Block diagram editing
• Nonlinear 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
Bell Helicopter Develops the First Civilian Tiltrotor, Using Model-Based Design

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

Link to technical article
Outline

Why Simulink?

Working with Simulink

How Simulink works

Componentizing models

Continuous and discrete models
Launching Simulink

Trial >> simulink

Simulink Library Browser

Commonly Used Blocks

van der Pol Equation

Scope
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**
    - Ode1
    - Ode2
    - Ode3
    - Ode4
    - Ode5
    - Ode8
  - **Variable-Step**
    - Ode45
    - Ode23
    - Ode113
    - Ode15s
    - Ode23s
    - 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

- How to undo ‘Create Subsystem’?
Subsystems

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

- 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

\[ x'(t) = 3x(t) + u(t) \iff \frac{X(s)}{U(s)} = \frac{1}{s - 3} \]

Transfer function:

\[ sx = 3x + u \]

\[ x = \frac{u}{s-3} \]

\[ \frac{x}{u} = \frac{1}{s-3} \]
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
More on Simulink

- Simulink Tutorials
- Demos and Webinars
- Documentation