Algorithms in Everything
Using MATLAB & Simulink to Build Algorithms in Everything

Simplifying your work…

…by working at good levels of abstraction.
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs
Creating Your Own Data

Signal Builder
Create and generate interchangeable groups of signals whose waveforms are piecewise linear

Library: Simulink / Sources

Description

Note
The Signal Builder block is not recommended. Use the Signal Editor block instead.
Creating Your Own Data
Changing Block Parameters

Diagram showing a Simulink model with blocks named sensor.x1, sensor.x2, Gain, Operator Logic1, Goal Delta, and connections to goal.x_goal and goal.signal1.
Model Navigation
Adding New Blocks
This implements the electromechanical components of a Faulhaber Series 0615 DC-Micromotor permanent magnet electric motor. The containing system needs to provide voltage and mechanical load.
This implements the electromechanical components of a Faulhaber Series 0615 DC-Micromotor permanent magnet electric motor. The containing system needs to provide voltage and mechanical load.
Anti-Windup PID Control Demonstration

![Diagram of Anti-Windup PID Control System](image)
Componentisation
Use libraries to share rarely changing *utilities* across models and projects

Less duplication
More readable
More efficient
How do you edit the library block?
Editing design components in libraries

Edit library directly?

Edit link in-context?
Subsystem Reference

Select the settings for the subsystem block. To enable parameters for code generation, select 'Treat as atomic unit'.

Main  Code Generation  Subsystem Reference

Subsystem: ClientComputer  Browse...  Open Subsystem

Current Folder

<table>
<thead>
<tr>
<th>Name</th>
<th>Date Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL_MGR.slx</td>
<td>5/7/2019 4:23 PM</td>
</tr>
<tr>
<td>SL_Ball.slx</td>
<td>5/7/2019 4:25 PM</td>
</tr>
<tr>
<td>SL_Paddle.slx</td>
<td>5/8/2019 12:36 PM</td>
</tr>
<tr>
<td>PongUtilities.slx</td>
<td>5/8/2019 12:42 PM</td>
</tr>
<tr>
<td>PongIntegration.slx</td>
<td>5/8/2019 2:34 PM</td>
</tr>
<tr>
<td>PongIntegration_BEP.slx</td>
<td>5/9/2019 9:39 AM</td>
</tr>
<tr>
<td>ClientComputer.slx</td>
<td>5/14/2019 4:37 PM</td>
</tr>
</tbody>
</table>
Subsystem Reference
Subsystem Reference componentizes your model into separate files

- Reduce file contention
- Edit in-context without links
- Automatic synchronization
Summary of componentization techniques

**Libraries**
- Graphical reuse
- **Dynamic** interface
- Ideal for utilities & blocksets

**Subsystem Reference**
- Graphical reuse
- **Dynamic** interface
- Ideal for storing design model components

**Model Reference**
- Behaviour reuse
- **Defined** interface
- Ideal for code generation components

R2019b
Simulink has grown in capability
Introducing the Simulink Toolstrip
Improved Commands Discoverability & Workflow

Basic Simulation workflow

Toolbar
Controls have no labels. What do they do?

What is the order of steps In workflow?

Toolstrip
Selection-based Filtering

Format tab automatically adds/removes sections based on current selection on canvas
Can't find what you're looking for? Try Apps in Simulink or view Menus to Toolstrip Mapping. Do not show again.
Controlling the Execution of Model Components

Schedulable Rate-Based Model

Export Function Model
Schedule Editor

Partition = Entry Point

Execution order

Connection = Data transfer
Change the schedule easily
Using the Schedule Editor simplifies your model
Using MATLAB & Simulink to Build Algorithms in Everything
XY Visualization
Parallel Simulations in Simulink

Simulation Manager

![Simulation Manager screenshot]

MATLAB Desktop

Simulation Jobs

Simulation Results

batchsim

Scaling
Graphical analysis helps identify failures quickly

Expected Plot
Viewing Generated Code Alongside the Model
Include Custom Code in Test & Verification

Simulink

Simulink Design Verifier

Stateflow

Simulink Coverage

C/C++
Using MATLAB & Simulink to Build Algorithms in Everything
<table>
<thead>
<tr>
<th>Time</th>
<th>Session Title</th>
<th>Presenter/Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:45</td>
<td>Deep Learning and Reinforcement Learning Workflows in AI</td>
<td>Jon Cherry, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Controlling Complexity at McLaren Automotive Using the Latest MATLAB Features</td>
<td>Matthew Gwilliam, McLaren Automotive Ltd</td>
</tr>
<tr>
<td>12:15</td>
<td>Extreme Quantum Mechanics in MATLAB</td>
<td>Illya Kuprov, University of Southampton</td>
</tr>
<tr>
<td></td>
<td>Systems Engineering: Requirements to Architecture to Simulation</td>
<td>Mark Walker, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Pixels to Features to Models: Object Detection and Image Segmentation</td>
<td>Matthew Elliott, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Deploying Deep Neural Networks to Embedded GPUs and GPUs</td>
<td>Steven Thomsett, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Developing a User Community to Drive Sharing, Self-Learning, and Personal Development</td>
<td>Matthew Offredi and Rayner Saggia, BAE Systems</td>
</tr>
<tr>
<td>12:45</td>
<td>Lunch</td>
<td></td>
</tr>
<tr>
<td>14:00</td>
<td>Big Data, Big Transformation: Big Benefits for Large-Scale Engineering Products</td>
<td>Martin McDonald and Andrew Gorin, Leonards</td>
</tr>
<tr>
<td></td>
<td>Simulating Passenger Comfort and Motion Sickness in Autonomous Vehicles</td>
<td>Michael Wheelon, Ricardo</td>
</tr>
<tr>
<td></td>
<td>Introduction to Simulink and Stateflow</td>
<td>Tim Johns, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Software Development Practices with MATLAB</td>
<td>David Sampson, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Developing a Battery Management System Using Simulink</td>
<td>Chris Lim, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Accelerating Embedded Software Verification with PolySpace Static Code Analysis</td>
<td>Stefan David, MathWorks</td>
</tr>
<tr>
<td>14:30</td>
<td>Becoming a Data-Centric Engineering Team: Catching Up to the Data Deluge</td>
<td>Paul Peeling, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Automated Driving System Design and Simulation Using MATLAB and Simulink</td>
<td>GianCarlo Pacitti, MathWorks</td>
</tr>
<tr>
<td>15:15</td>
<td>Break</td>
<td></td>
</tr>
<tr>
<td>15:45</td>
<td>Developing Smart IoT Sensors Using the MathWorks Toolchain</td>
<td>Samuel Bailey, Skyrad Consulting</td>
</tr>
<tr>
<td></td>
<td>Synchronous Machine Modelling Using Simscape</td>
<td>Ponnio Rami, Cummins Generator Technologies</td>
</tr>
<tr>
<td></td>
<td>Sensor Fusion and Tracking for Autonomous Systems</td>
<td>Marco Willerton, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Simplifying Requirements-Based Verification with Model-Based Design</td>
<td>Fraser Macniven, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Predictive Maintenance with MATLAB</td>
<td>Phil Rottier, MathWorks</td>
</tr>
<tr>
<td>16:15</td>
<td>Industrial IoT and Digital Twins</td>
<td>Conous Aktalali, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Developing Fit-For-Purpose Simscape Models to Support System and Control Design</td>
<td>Rick Hyde, MathWorks</td>
</tr>
<tr>
<td>17:00</td>
<td>End of Day</td>
<td></td>
</tr>
</tbody>
</table>
Read the Release Notes

Explore What's New
Get more out of MATLAB and Simulink by downloading the latest release.
Download now

Release Highlights

Deep Learning
Develop controllers and decision making systems using reinforcement learning, train deep learning models on NVIDIA DGX and cloud platforms, and apply deep learning to 3-D data.
Learn more

Automotive
Design and simulate AUTOSAR software, interface with HERE HD maps, and generate energy balance reports.
Learn more

Systems Engineering
Design and simulate a software architecture of an electrical power steering system.
Learn more

Deep Learning
Use automatic differentiation, shared weights, and custom training loops to build advanced deep learning architectures, like GANs and Siamese networks.
Learn more

Stateflow Onramp
Learn the basics of how to create, edit, and simulate Stateflow models through an interactive tutorial.
Learn more

Automotive
Author AUTOSAR compositions and simulate with basic software services; test automated driving algorithms in 3D simulation; and leverage the deep learning engine model and P2-F4 HEV models for control algorithm testing and system simulation.
Learn more
Get Started

MATLAB Onramp
Quickly learn the essentials of MATLAB.

Simulink Onramp
Learn to create, edit, and troubleshoot Simulink models.

Deep Learning Onramp
Learn to use deep learning techniques in MATLAB for image recognition.

Stateflow Onramp
Learn the basics of how to create, edit, and simulate state machines in Stateflow®
MATLAB EXPO 2019