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

MATLAB & SIMULINK®
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
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
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'.
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

1. Log Signals
2. Add Viewer
3. Signal Table
4. Step Time
5. Normal
6. Fast Restart
7. Step Back
8. Run
9. Step Forward
10. Step
11. Data Inspector
12. shift_logic Messages
13. Logic Analyzer
14. Simscape Results
Selection-based Filtering

Format tab automatically adds/removes sections based on current selection on canvas.
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 Jobs
Simulation Results

batchsim

MathWorks Desktop

Simulation Details

Run ID: 1
Status: Completed
Progress: 100%
Elapsed Time: 00:00:10

Parameters
Type | Name | Value
--- | --- | ---
Block Parameter | Cf | 7.5

Total Simulations: 200
Elapsed Time: 00:03:06
Number of Active Workers: 8
Estimated Time Remaining: 00:01:36

Errors/Aborted: 0
Completed: 68
Active: 8
Queued: 124

Scaling
Graphical analysis helps identify failures quickly
Viewing Generated Code Alongside the Model

Fuel Rate Control Subsystem

- single D2 (g/s) in
- validate_sample_time
- sensors
- EngSensors D2
- es_i
- O2_normal
- control_logic
- O2_normal
- fuel_mode
- fuel_rate
- fuel_calc
- airflow_calc
- fb_correction
- single D2 (g/s)
- est_airflow (g/s)
- fuel_mode
- fuel_rate (g/s)
Include Custom Code in Test & Verification

Simulink

C/C++

Simulink Design Verifier

= ×

= ✔

Simulink Coverage

<table>
<thead>
<tr>
<th>D1</th>
<th>C1</th>
<th>MCDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Stateflow

C/C++
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs  ➔  Design  ➔  Outputs
<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</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 Greens, McLaren Automotive Ltd</td>
</tr>
<tr>
<td>12:15</td>
<td>Extreme Quantum Mechanics in MATLAB</td>
<td>Ilya 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 CPUs</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 O'Ferrall and Rayner Saggi, 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, Leonardo</td>
</tr>
<tr>
<td></td>
<td>Simulating Passenger Comfort and Motion Sickness in Autonomous Vehicles</td>
<td>Michael Wheeler, 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 Pacilli, 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>Peonio Rani, Cummins Generator Technologies</td>
</tr>
<tr>
<td></td>
<td>Sensor Fusion and Tracking for Autonomous Systems</td>
<td>Mario Willerton, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Simplifying Requirements-Based Verification with Model-Based Design</td>
<td>Fraser Macwhirter, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Predictive Maintenance with MATLAB</td>
<td>Phil Rothier, MathWorks</td>
</tr>
<tr>
<td>16:15</td>
<td>Industrial IoT and Digital Twins</td>
<td>Coorunz Miahatat, MathWorks</td>
</tr>
<tr>
<td></td>
<td>Developing Fit-For-Purpose Simscape Models to Support System and Control Design</td>
<td>Nick Hyde, MathWorks</td>
</tr>
<tr>
<td>17:00</td>
<td>End of Day</td>
<td></td>
</tr>
</tbody>
</table>
Read the Release Notes
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