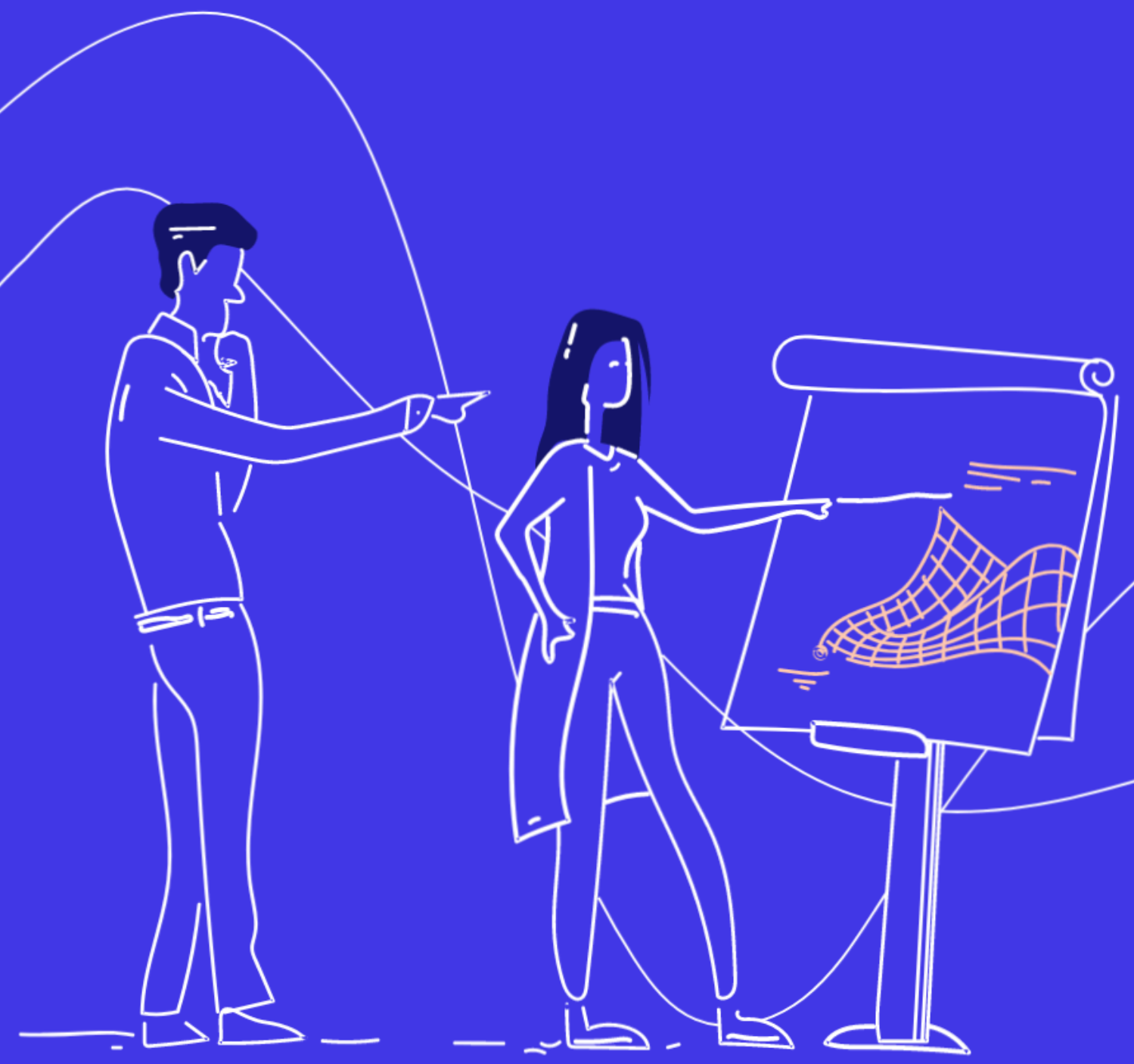
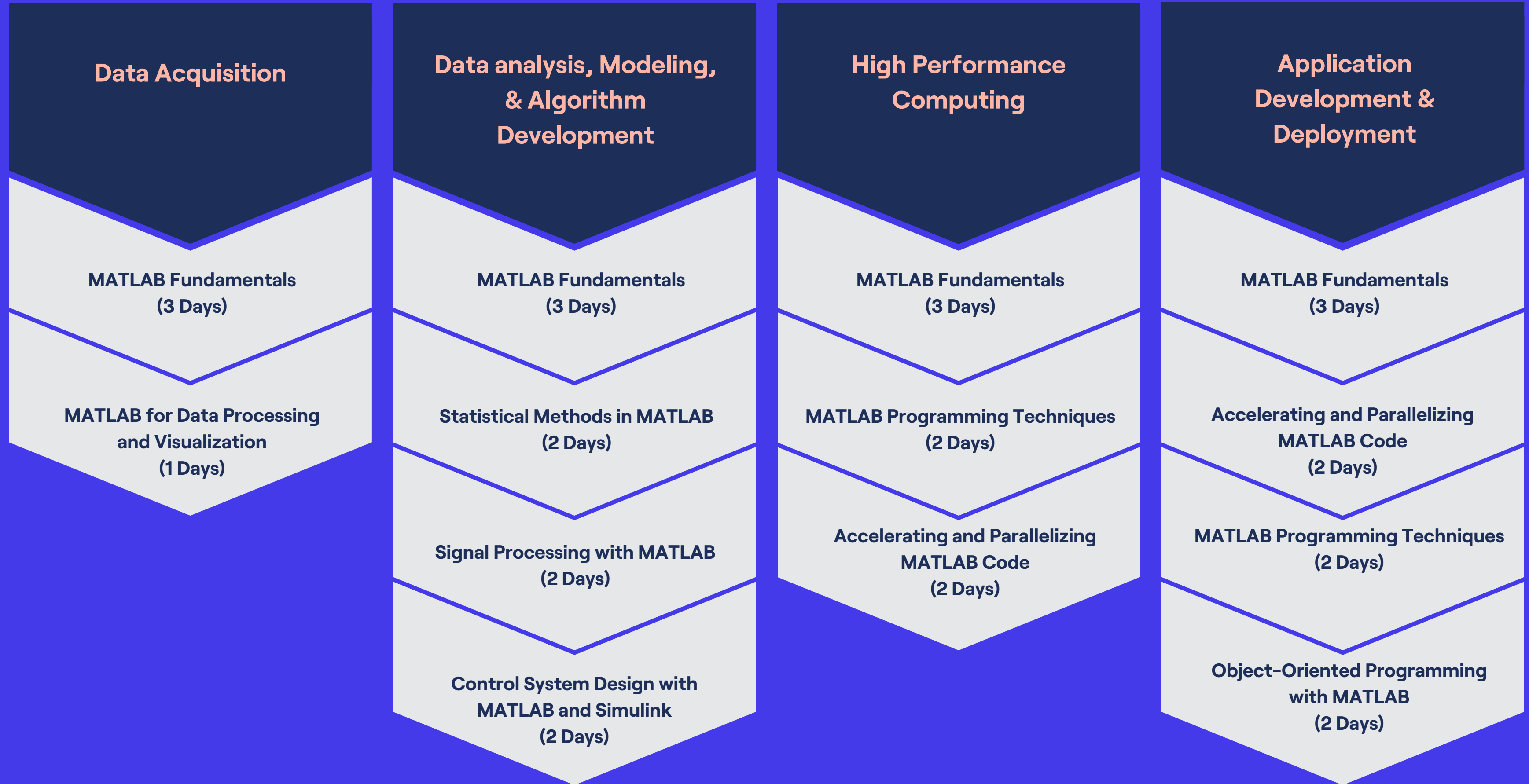




# Powertrain Systems Curriculum



SciEngineer's training courses are designed to help organizations and individuals close skills gaps, keep up-to-date with the industry-accepted best practices and achieve the greatest value from MathWorks® and COMSOL® Products.



### Modeling & Simulation

**MATLAB Fundamentals  
(3 Days)**

**Simulink Fundamentals  
(2 Days)**

**Signal Processing with Simulink  
(3 Days)**

**Stateflow for Logic-Driven System  
Modeling  
(2 Days)**

**Modeling Physical Systems with  
Simscape  
(1 Day)**

### Code Generation

**MATLAB Fundamentals  
(3 Days)**

**Simulink Fundamentals  
(2 Days)**

**Simulink Model Management and  
Architecture  
(2 Days)**

**Integrating Code with Simulink  
(1 Day)**

**Embedded Coder for Production  
Code Generation  
(3 Days)**

**Code Generation for Classic  
AUTOSAR Software Components  
(2 Days)**

**MATLAB to C with MATLAB Coder  
(2 Days)**

**Modeling Multibody Mechanical  
Systems with Simscape  
(1 Day)**

**Modeling Electrical Power Systems  
with Simscape  
(1 Day)**

### Rapid Prototyping

**MATLAB Fundamentals  
(3 Days)**

**Simulink Fundamentals  
(2 Days)**

**Real-Time Testing with Simulink  
Real-Time and Speedgoat  
Hardware  
(2 Days)**

### Test, Verification, & Tuning

**MATLAB Fundamentals  
(3 Days)**

**Simulink Fundamentals  
(2 Days)**

**Stateflow for Logic-Driven System  
Modeling  
(2 Days)**

**Simulink Model Management and  
Architecture  
(2 Days)**

**Simulation-Based Testing with  
Simulink  
(1 Day)**

### Development Process Support

**MATLAB Fundamentals  
(3 Days)**

**Simulink Fundamentals  
(2 Days)**

**Simulink Model Management and  
Architecture  
(2 Days)**

**Simulation-Based Testing with  
Simulink  
(1 Day)**

**Real-Time Testing with Simulink  
Real-Time and Speedgoat  
Hardware  
(2 Days)**

## Code Verification

**Polyspace for C/C++ Code  
Verification  
(3 Days)**

# MATLAB Fundamentals

This three-day course provides a comprehensive introduction to the MATLAB technical computing environment. Themes of data analysis, visualization, modeling, and programming are explored throughout the course. This course is intended for beginning users and those looking for a review.

## Prerequisites

Undergraduate-level mathematics and experience with basic computer operations.

[Detailed course outline >>](#)

DURATION	LEVEL
3 Days	Basic
	

### TOPICS

## Day 1

- Working with the MATLAB User Interface
- Variables and Commands
- Analysis and Visualization with Vectors

## Day 2

- Analysis and Visualization with Matrices
- Tables of Data
- Conditional Data Selection
- Organizing Data

## Day 3

- Analyzing Data
- Increasing Automation with Programming Constructs
- Increasing Automation with Functions

# MATLAB for Data Processing and Visualization

This one-day course focuses on importing and preparing data for data analytics applications. Topics discussed include importing data from multiple sources, processing data, producing informative customized graphics and working with irregular data.

## Prerequisites

MATLAB Fundamentals

[Detailed course outline >>](#)

DURATION	LEVEL
1 day	Medium
	

### TOPICS

## Day 1

- Importing Data
- Processing Data
- Customizing Visualizations
- Working with Irregular Data

# Statistical Methods in MATLAB

This two-day course focuses on performing statistical data analysis with MATLAB and Statistics and Machine Learning Toolbox. Examples and exercises demonstrate the use of appropriate MATLAB and Statistics and Machine Learning Toolbox functionality throughout the analysis process; from importing and organizing data, to exploratory analysis, to confirmatory analysis and simulation.

## Prerequisites

MATLAB Fundamentals and knowledge of basic Statistics and Machine Learning Toolbox.

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Medium
	

### TOPICS

## Day 1

- Importing and Organizing Data
- Exploring Data
- Distributions
- Hypothesis Tests

## Day 2

- Analysis of Variance
- Regression
- Working with Multiple Dimensions
- Random Numbers and Simulation



# Signal Processing with MATLAB

This two-day course shows how to analyze signals and design signal processing systems using MATLAB and Signal Processing Toolbox. Parts of the course also use DSP System Toolbox. This course focuses on creating and analyzing signals, performing spectral analysis, designing and analyzing filters, designing multirate and adaptive filters.

## Prerequisites

MATLAB Fundamentals or equivalent experience using MATLAB, and a good understanding of signal processing theory, including linear systems, spectral analysis, and filter design

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Medium
	

### TOPICS

## Day 1

- Signals in MATLAB
- Spectral Analysis
- Linear Time Invariant Systems

## Day 2

- Filter Design
- The Signal Analysis App
- Multirate Filters
- Adaptive Filter Design

# Control System Design with MATLAB and Simulink

This two-day course provides a general understanding of how to accelerate the design process for closed-loop control systems using MATLAB and Simulink.

Topics included: control system design overview, system modeling, identification and analysis, control design and controller implementation.

## Prerequisites

MATLAB Fundamentals and Simulink Fundamentals or equivalent experience using MATLAB and Simulink. Also, an understanding of terminology and concepts related to common control systems.

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Medium
	

### TOPICS

#### Day 1

- Control System Design Overview
- Model Representations
- System Identification
- Parameter Estimation
- System Analysis

#### Day 2

- Linearization
- PID Control in Simulink
- Classical Control Design
- Response Optimization
- Controller Implementation

# MATLAB Programming Techniques

This two-day course provides hands-on experience using the features in the MATLAB language to write efficient, robust, and well-organized code. These concepts form the foundation for writing full applications, developing algorithms, and extending built-in MATLAB capabilities. Details of performance optimization, as well as tools for writing, debugging, and profiling code are covered.

## Prerequisites

MATLAB Fundamentals or equivalent experience using MATLAB

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Medium
	

### TOPICS

#### Day 1

- Structuring Data
- Managing Data Efficiently
- Utilizing Development Tools

#### Day 2

- Creating Robust Applications
- Structuring Code
- Verifying Application Behavior

# Accelerating and Parallelizing MATLAB Code

This two-day course covers a variety of techniques for making your MATLAB code run faster. You will identify and remove computational bottlenecks using techniques like pre-allocation and vectorization. In addition, you will compile MATLAB code into MEX-files using MATLAB Coder. On top of that, you will take advantage of multiple cores on your computer by parallelizing for-loops with Parallel Computing Toolbox and scale up across multiple computers using MATLAB Parallel Server.

## Prerequisites

MATLAB Fundamentals, or equivalent experience using MATLAB

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Medium
	

### TOPICS

## Day 1

- Improving Performance
- Generating MEX-Files
- Parallelizing Computations

## Day 2

- Parallel for-Loops
- Offloading Execution
- Working with Clusters
- GPU Computing

# Object-Oriented Programming with MATLAB

This two-day course focuses on using object-oriented programming techniques to develop and maintain complex MATLAB applications. The main topics are creating custom data types, designing a MATLAB class, building class hierarchies, facilitating multiple references, writing unit tests and synchronizing objects.

## Prerequisites

MATLAB Programming Techniques or equivalent experience using MATLAB

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Medium
	

### TOPICS

#### Day 1

- Creating Custom Data Types
- Designing a MATLAB Class
- Building Class Hierarchies

#### Day 2

- Facilitating Multiple References
- Writing Unit Tests
- Synchronizing Objects

# Simulink Fundamentals

This two-day course provides a comprehensive introduction to the Simulink environment. It demonstrates how to create, modify Simulink models, improve simulation accuracy and speed and create reusable model components using subsystems, model references and libraries.

## Prerequisites

MATLAB Fundamentals

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Basic
	

### TOPICS

## Day 1

- Creating and Simulating a Model
- Modeling Programming Constructs
- Modeling Discrete Systems
- Modeling Continuous Systems

## Day 2

- Solver Selection
- Developing Model Hierarchy
- Modeling Conditionally Executed Algorithms
- Combining Models into Diagrams
- Creating Libraries

# Stateflow for Logic-Driven System Modeling

This two-day course shows how to model and simulate decision logic using Stateflow. The course focuses on how to employ flow charts, state machines, truth tables, state transition tables and component-based modeling in Simulink designs.

## Prerequisites

MATLAB Fundamentals and Simulink Fundamentals

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Basic
	

### TOPICS

#### Day 1

- Modeling Flow Charts
- Modeling State Machines
- Hierarchical State Diagrams
- Parallel State Diagrams

#### Day 2

- Using Events in State Diagrams
- Calling Functions from Stateflow
- Truth Tables and State Transition Tables
- Component-Based Modeling in Stateflow

# Modeling Physical Systems with Simscape

This one-day course discusses how to model systems in several physical domains such as electrical, mechanical, and hydraulic. This course focuses on interpreting Simscape diagrams, combining them with Simulink models, modeling energy transfer between different physical domains, and creating userdefined Simscape components.

## Prerequisites

MATLAB Fundamentals and Simulink Fundamentals

[Detailed course outline >>](#)

DURATION	LEVEL
1 day	Medium
	

### TOPICS

## Day 1

- Introduction to Simscape and the Physical Network Approach
- Working with Simscape Components
- Connecting Physical Domains
- Combining Simscape Models and Simulink Models
- Creating Custom Components with the Simscape Language



# Modeling Multibody Mechanical Systems with Simscape

This one-day course focuses on how to model rigid-body mechanical systems in the Simulink environment using Simscape Multibody. Topics include: modeling simple multibody systems, creating reusable models of mechanical systems, importing models from CAD software and combining Simulink, Simscape and Simscape Multibody blocks.

## Prerequisites

MATLAB Fundamentals, Simulink Fundamentals, and Modeling Physical Systems with Simscape

[Detailed course outline >>](#)

DURATION	LEVEL
1 day	Medium
	

### TOPICS

## Day 1

- Introduction to Multibody Simulation
- Refining Components
- Assembling Mechanisms
- Importing CAD Models
- Connecting to Simscape and Simulink

# Modeling Electrical Power Systems with Simscape

This one-day course discusses how to model electrical power systems in the Simulink environment using the Simscape Electrical Specialized Power Systems library. This course focuses on creating three-phase systems with passive elements and with electrical machines, analyzing and controlling electrical power systems, modeling power electronic components and speeding up simulation of electrical models.

## Prerequisites

MATLAB Fundamentals, Simulink Fundamentals, and Modeling Physical Systems with Simscape

[Detailed course outline >>](#)

DURATION	LEVEL
1 day	Medium
	

### TOPICS

## Day 1

- Introduction to Three-Phase Systems
- Three-Phase Systems with Electrical Machines
- Controlling Electrical Machines
- Power Electronics

# Simulink Model Management and Architecture

This two-day course describes techniques for applying ModelBased Design in a common design workflow. It provides guidance on managing and sharing Simulink models when working in a large-scale project environment. This course is intended for intermediate or advanced Simulink users.

## Prerequisites

MATLAB Fundamentals and Simulink Fundamentals. This course is intended for intermediate or advanced Simulink users.

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Advanced
	

### TOPICS

## Day 1

- Model-Based Design
- Requirements Linking and Interface Control
- Model Architecture
- Project Management

## Day 2

- Data Management
- Data Customization
- Modeling Standards
- Reporting

# Integrating Code with Simulink

This one-day course presents multiple methods for integrating C code and MATLAB code into Simulink models. Topics discussed include writing C MEX S-functions, integrating MATLAB code, and the Legacy Code Tool for wrapping external C functions into Simulink.

## Prerequisites

Simulink Fundamentals, MATLAB Fundamentals, and knowledge of C programming

[Detailed course outline >>](#)

DURATION	LEVEL
1 day	Medium
	

### TOPICS

## Day 1

- Code Integration Methods
- Transitioning from MATLAB to Simulink
- Calling External Routines
- Writing Wrapper S-Functions
- Code Generation Considerations
- Code Integration Methods Review

# Embedded Coder for Production Code Generation

This two-day course describes techniques for generating, validating, and customizing embedded code using Embedded Coder.

Topics include: Generated code structure and execution; Code generation options and optimizations; Integrating generated code with external code; Generating code for multirate systems; Customizing generated code and data.

## Prerequisites

- Simulink Fundamentals (or Simulink Fundamentals for Automotive Applications or Simulink Fundamentals for Aerospace Applications)
- Knowledge of C programming language.

[Detailed course outline »](#)

DURATION	LEVEL
2 Days	Medium
	

### TOPICS

## Day 1

- Generating Embedded Code
- Optimizing Generated Code
- Integrating Generated Code with External Code
- Controlling Function Prototypes
- Customizing Data Characteristics in Simulink®

## Day 2

- Customizing Data Characteristics Using Data Objects
- Customizing Generated Code Architecture
- Model Referencing and Bus Objects
- Scheduling Generated Code Execution
- Improving Code Efficiency and Compliance

# Code Generation for AUTOSAR Software Components

This one-day course discusses AUTOSAR-compliant modeling and code generation using the Embedded Coder Support Package for AUTOSAR Standard.

Topics included: Importing and exporting AUTOSAR software components; Configuring AUTOSAR communication elements; Modeling AUTOSAR events; Creating calibration parameters.

## Prerequisites

- Simulink for System and Algorithm Modeling (or Simulink for Automotive System Design or Simulink for Aerospace System Design)
- Embedded Coder for Production Code Generation
- Knowledge of C programming language
- Knowledge of the AUTOSAR standard

[Detailed course outline >>](#)

A dark blue rounded rectangle containing course metadata. On the left, under 'DURATION', it says '1 Day' with three circles below it (one orange, two black). On the right, under 'LEVEL', it says 'Advanced' with three orange bars below it.

DURATION	LEVEL
1 Day	Advanced

### TOPICS

## Day 1

- Importing AUTOSAR Software Components
- Exporting AUTOSAR Software Components
- Modeling AUTOSAR Ports and Interfaces
- Modeling AUTOSAR Events
- Modeling Calibration Parameters and Per-Instance Memory

# MATLAB to C with MATLAB Coder

This two-day course covers C code generation from MATLAB code using MATLAB Coder. The focus is on making existing MATLAB code compliant, generating C code that meets optimization requirements, and integrating generated code with external modules. Topics include: preparing MATLAB code for code generation, working with fixed-size and variable-size data, integrating with external code and optimizing generated code.

## Prerequisites

MATLAB Fundamentals and knowledge of C programming language

[Detailed course outline >>](#)

DURATION	LEVEL
2 Days	Advanced
	

### TOPICS

#### Day 1

- Code Generation with MATLAB Coder
- Preparing MATLAB Code for Code Generation
- Working with Fixed-Size Data
- Working with Variable-Size Data

#### Day 2

- Working with Global Data, Structures, and Cell Arrays
- Integrating with External Code
- Optimizing Generated Code

# Real-Time Testing with Simulink Real-Time and Speedgoat Hardware

This two-day course focuses on real-time testing workflows using Simulink Real-Time and Speedgoat real-time target computers.

Topics include: Converting desktop-based simulation applications into real-time applications; Conducting rapid control prototyping with physical device under control; Creating interactive interfaces and formal test suites, Using standard communication protocols; Optimizing real-time applications and hardware-in-the-loop testing.

## Prerequisites

- Simulink Fundamentals (or Simulink Fundamentals for Automotive Applications or Simulink Fundamentals for Aerospace Applications)
- Knowledge of Simscape™ preferred

[Detailed course outline >>](#)



### TOPICS

## Day 1

- Workflow Overview
- Developing Real-Time Applications
- Building Interactive Interfaces

## Day 2

- Automating Real-Time Tests
- Using Communications Protocols
- Optimizing Plant Models for Real-Time Execution
- Hardware-in-the-Loop Testing



# Simulation-Based Testing with Simulink

This one-day course describes techniques for testing Simulink model behavior against system requirements using Simulink Test, Simulink Requirements, and Simulink Coverage. This course focuses on verification and validation, developing test cases, analyzing test results and creating repeatable groups of tests.

## Prerequisites

MATLAB Fundamentals and Simulink Fundamentals

[Detailed course outline >>](#)

A dark blue rounded rectangular card containing course metadata. It is divided into two columns by a vertical line. The left column is titled 'DURATION' and shows '1 day' with a progress indicator of one orange circle followed by two black circles. The right column is titled 'LEVEL' and shows 'Advanced' with a progress indicator of three orange rectangles.

DURATION	LEVEL
1 day	Advanced

### TOPICS

## Day 1

- Verification and Validation in Model-Based Design
- Developing Test Cases
- Analyzing Test Results
- Building Test Suites

# Polyspace for C/C++ Code Verification

This two-day course discusses the use of Polyspace Code Prover to prove code correctness, improve software quality metrics, and ensure product integrity. This course describes techniques for creating a verification project, reviewing and understanding verification results, emulating target execution environments, handling missing functions and data, managing unproven code, applying MISRA-C rules and reporting analysis results.

## Prerequisites

Strong knowledge of C or C++

[Detailed course outline >>](#)

DURATION	LEVEL
3 days	Advanced
	

### TOPICS

#### Day 1

- Polyspace Workflow Overview
- Polyspace Bug Finder Analysis
- Analyzing Polyspace Code Prover Results
- Code Verification Checks

#### Day 2

- Managing Polyspace Code Prover Verifications and Results
- Adding Precision to Polyspace Code Prover Verifications
- Integration Analysis
- Application Analysis

#### Day 3

- (optional, available with private training only)
- Hands-On Instruction (Optional)

# The Value of an Experienced Training Expert

Our training courses are developed by MathWorks' team of training engineers with exclusive product knowledge gained from working closely with product developers. They acquire significant hands-on experience by using new products months before they are released and are always current on new capabilities.

## Learn Relevant Skills

Each course contains a set of learning objectives designed to help participants quickly master necessary skills. Our hands-on approach allows participants to practice, apply, and evaluate their knowledge in the classroom.

## Receive Expert Instruction

Our training employs industry-accepted best practices for adult learning and technical instruction, and has developed course content that facilitates a "Presentation, Practice, Test" approach to learning. All training engineers have been selected based on their theoretical knowledge, technical education, experience, and teaching ability.

## Increase Team Success Rates

According to post-training surveys, teams who receive 40 hours of training meet project objectives three times as often as those who receive 30 hours or less. This increase in training time raises the likelihood of meeting objectives by 90%.



**Expand your  
knowledge**

