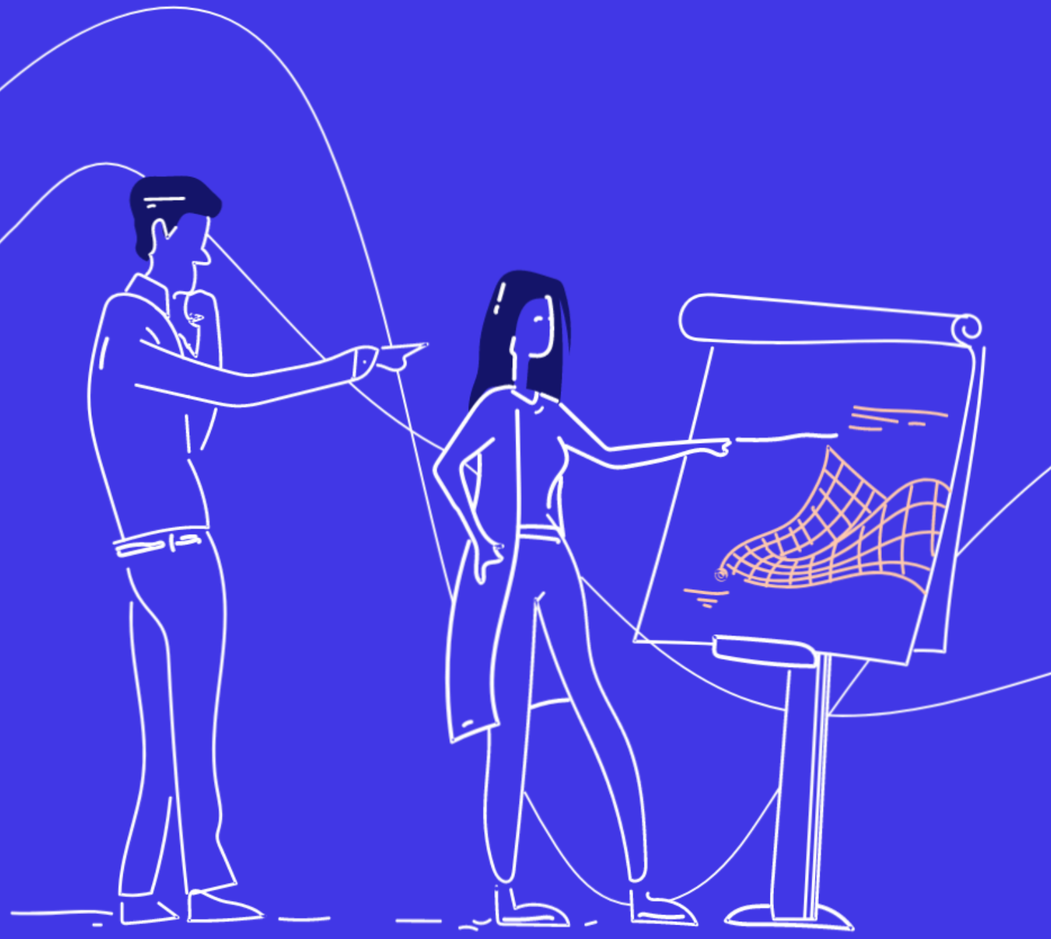
sciengineer

# Aerospace Digital Transformation 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.

Aerospace Digital Transformation Curriculum

SciEngineer's Training Services

## Core Model-Based Engineering Platform

- MATLAB Essentials for Simulink (1 day)
- Simulink for Aerospace System Design (2 Days)
- Stateflow for Logic-Driven System Modeling (2 Days)

## Architecting, Managing, Testing & verifying Models

- System Composer for Architecture Modeling (1 day)
- Simulink Model Management and Architecture (2 Days)
- Simulation-Based Testing with Simulink (1 day)
- Design Verification with Simulink (1 day)

## Plant Modeling & Controlling

- Control System Design with MATLAB and Simulink (2 Days)
- Modeling Physical Systems with Simscape (1 day)
- Modeling Multibody Mechanical Systems with Simscape (1 day)
- Modeling Fluid Systems with Simscape (1 day)
- Power Electronics Control Design with Simulink and Simscape (1 day)

## Generating, Testing & Verifying Code

- Embedded Coder for Production Code Generation (2 Days)
- Polyspace for C/C++ Code Verification (2 Days)

# MATLAB Essentials for Simulink

This one-day course provides an introduction to the MATLAB® technical computing environment including topics most useful for Simulink workflows. No prior programming experience or knowledge of MATLAB is assumed. Themes of modeling, visualization, and programming are explored throughout the course.

## Prerequisites

Undergraduate-level mathematics and experience with basic computer operations

**Detailed course outline »**


TOPICS

## Day 1

- Exploring the MATLAB Environment
- Scripting with MATLAB Commands
- Working with Data in MATLAB
- Controlling Flow and Creating Functions

# Simulink Fundamentals for Aerospace Applications

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

## Prerequisites

MATLAB Fundamentals for Aerospace Applications

**Detailed course outline »**


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 »**


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

# System Composer™ for Architecture Modeling

This one-day course focuses on developing and analyzing model-based architectures with System Composer™ and Requirements Toolbox™.

## Prerequisites

Simulink Fundamentals or equivalent experience using Simulink is needed. Basic knowledge of Systems Engineering concepts is recommended.


[Detailed course outline »]

TOPICS

## Day 1

- Introduction to Model-Based Design
- Working with Requirements
- Logical System Architectures
- Physical System Architectures
- Capturing and Analyzing System Properties
- Enhancing Architecture Traceability

# 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 »**

TOPICS

## Day 1

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

## Day 2

- Data Management
- Data Customization
- Modeling Standards
- Reporting

# 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 »**


TOPICS

## Day 1

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

# Design Verification with Simulink

This one-day course focuses on using Simulink Design Verifier to ensure that a design is devoid of possible design errors, is fully tested, and satisfies necessary requirements. Themes of detecting design errors, automatically generating tests, property proving and managing model complexity are explored throughout the course.

## Prerequisites

MATLAB Fundamentals and Simulink Fundamentals

**Detailed course outline »**


TOPICS

## Day 1

- Understanding the Verification Workflow
- Detecting Design Errors
- Automatically Generating Tests
- Property Proving
- Managing Model Complexity

# 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 »**


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

# 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 »**


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 »**


TOPICS

## Day 1

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

# Modeling Fluid Systems with Simscape

This one-day course focuses on modeling fluid systems in Simulink using Simscape Fluids. Topics discussed include modeling fluid power systems, actuating and controlling fluid system models, modeling thermal liquid and gas cooling systems and exchanging heat between fluid domains.

## Prerequisites

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

**Detailed course outline »**


TOPICS

## Day 1

- Hydromechanical Systems
- Hydraulic Actuation and Control
- Thermal Liquid Systems
- Gas Systems

# Power Electronics Control Design with Simulink and Simscape

This one-day course focuses on modeling and controlling power electronic systems in the Simulink environment using Simscape Electrical. Themes of DC power electronic systems, converter model fidelity, linearization and control, three-phase power electronic systems, and motor control are explored throughout the course.

## Prerequisites

MATLAB Fundamentals, Simulink for System and Algorithm Modeling, and Modeling Physical Systems with Simscape

**Detailed course outline »**


TOPICS

## Day 1

- Introduction to Power Electronics
- Converter Model Fidelity
- Linearization and Control
- Modeling Three-Phase Power Electronic Systems
- Motor Control

# 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 optimalizations; 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 »**


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

# 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 »**


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%.

sciengineer

# Expand your knowledge