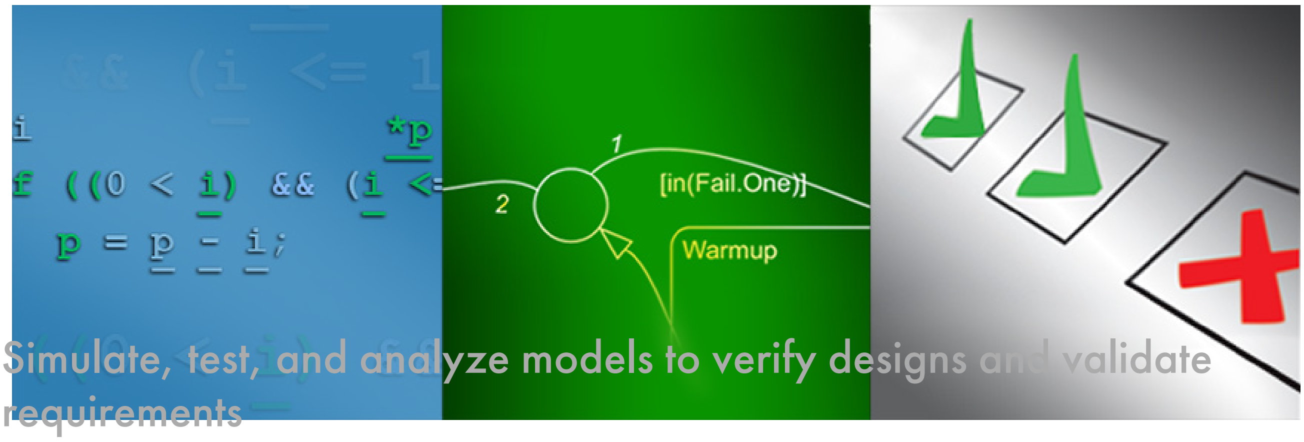


Simulate, test, and analyze models to verify designs and validate requirements

Verification and validation techniques applied throughout the development process enable you to find errors before they can derail your project. Most system design errors are introduced in the original specification, but aren't found until the test phase. When engineering teams use models to perform virtual testing early in the project, they eliminate problems and reduce development time by as much as 50%.

Activities for verification, validation, and test with Model-Based Design can be applied at every stage of the development process.

Early Requirements Validation

Uncover incorrect requirements and design flaws early by simulating system behavior to validate requirements and by specifying system design properties that formalize functional behavior.

Create High-Level System Models and Run Simulations

Gain further insight into system behavior through simulation by creating a system model that includes the software model and the physical and environmental aspects of your system. Associate this system model with system requirements to analyze and validate requirements early in the development process. Physical modeling products such as Simscape let you build functional plant models that can be simulated together with your software model.

Write System-Level Tests and Link Them to Requirements

MATLAB, Simulink, and Simulink Test let you define key simulation scenarios and document and systematically analyze behavior captured in your system model. Analysis results provide early feedback on the completeness and integrity of requirements and can be used to define expected behavior of the software model for further design refinement.

You explore the entire parameter space in simulation to help select those tests that are critical to run on real-time targets or real-world hardware. With Simulink Verification and Validation, you can link your system model to requirements for early insight and validation. Requirements traceability helps manage change and reduce waste in the design lifecycle.

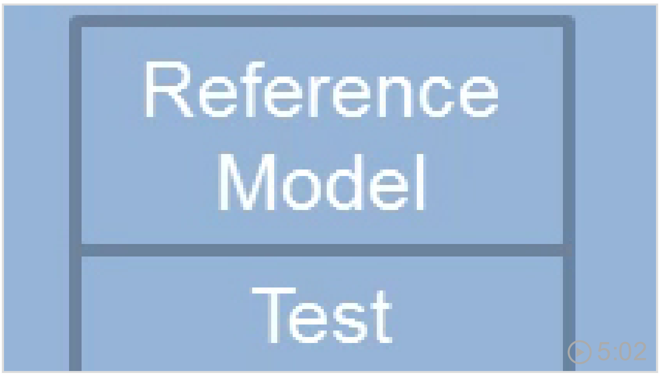
Refine Requirements and Define Design Properties

MATLAB and Simulink products let you capture design properties and functional requirements in the modeling environment. You model design properties and analyze models using formal methods to improve your designs and reveal unanticipated functionality that would be difficult to uncover by simulation alone. With Simulink Design Verifier you generate tests for your models and prove model properties.

Key Model V&V Techniques

- Collaboration: modeling standards checks
- Track design changes: traceability analysis
- Structural verification: model coverage
- Formal verification
  - test generation
  - static error detection
  - requirements proving

Track Design Changes with Requirements Traceability in MATLAB



Requirements-Based Testing

Explore Products for Early Requirements Validation

MATLAB®

Simulink®

Simscape™

Simulink Design Verifier™

Simulink Verification and Validation™

Simulink® Test™

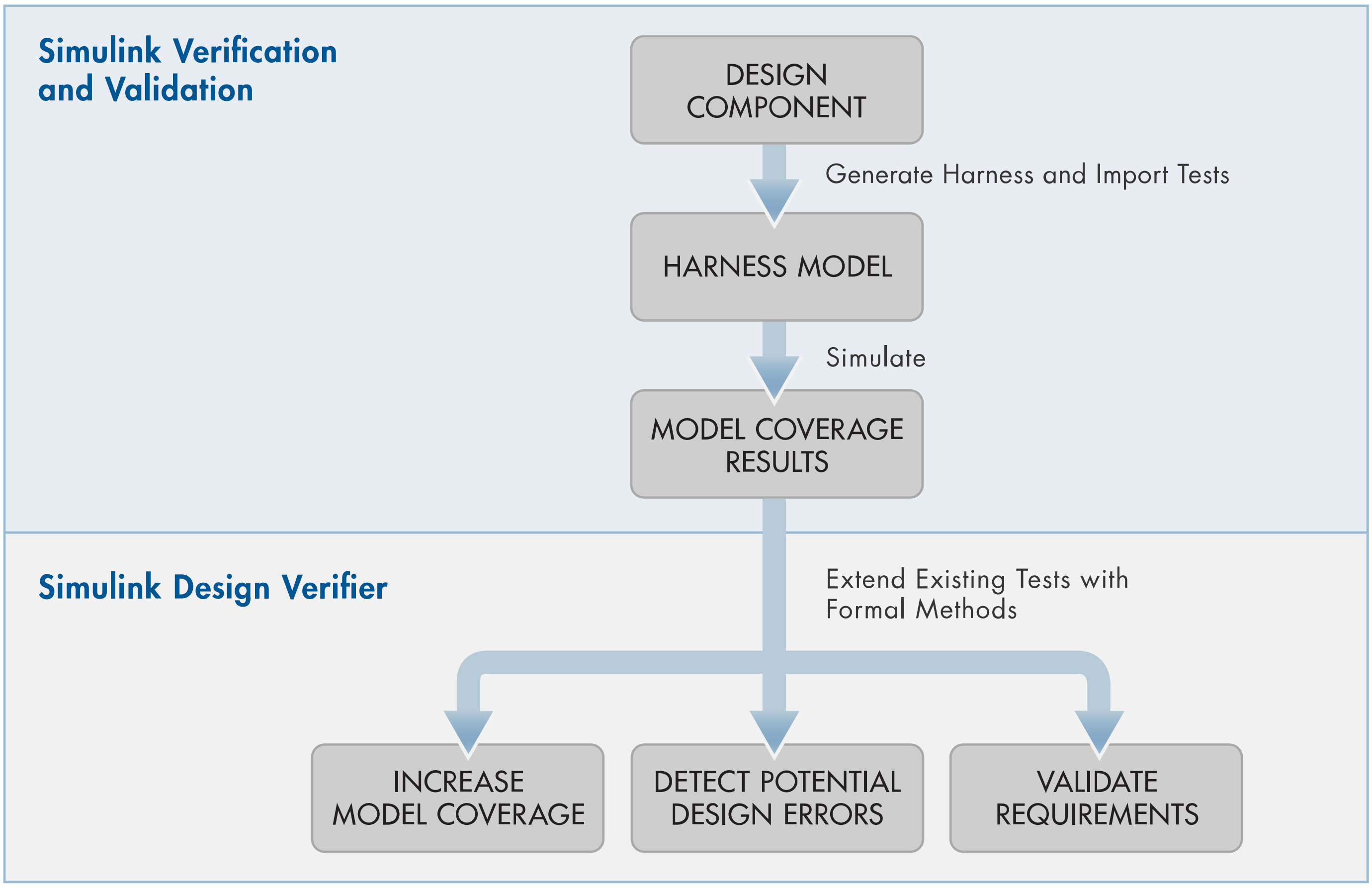
Design Verification

Model-Based Design incorporates simulation and testing to streamline the design verification process.

Design Verification Through Iterative Test and Analysis

Using MATLAB and Simulink products, you refine your design through rapid iterations and verification cycles in an interactive test environment. Early in the development process, you can:

- Detect design errors such as integer overflow, division by zero, and dead logic in models using formal verification methods
- Automatically generate reusable unit tests that satisfy model coverage and meet specific user-defined design verification objectives with Simulink Design Verifier
- Expose design flaws and inconsistencies in requirements using the model coverage capabilities in Simulink Verification and Validation


Simulink Verification and Validation
DESIGN COMPONENT
Generate Harness and Import Tests
HARNESS MODEL
Simulate
MODEL COVERAGE RESULTS
Simulink Design Verifier
Extend Existing Tests with Formal Methods
INCREASE MODEL COVERAGE
DETECT POTENTIAL DESIGN ERRORS
VALIDATE REQUIREMENTS

A model-based testing workflow

Use an Executable Specification to Verify Your Design

In Model-Based Design you create a system model that serves as an executable specification for component design and verification. Previously developed system-level tests can be reused to verify detailed component designs against the executable specification in simulation. You can also apply model coverage analysis tools from Simulink Verification and Validation to pinpoint untested elements of your design.

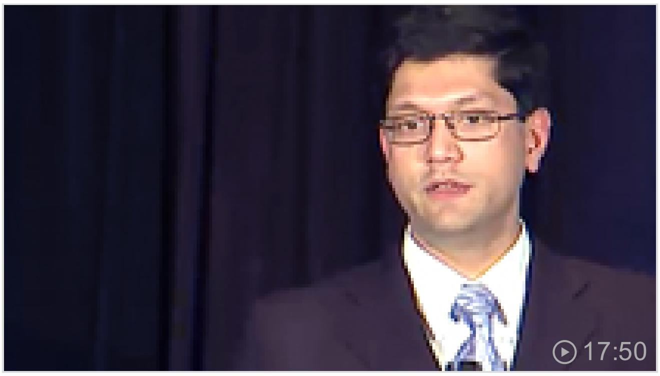
Generate Design Verification Documentation and Reports

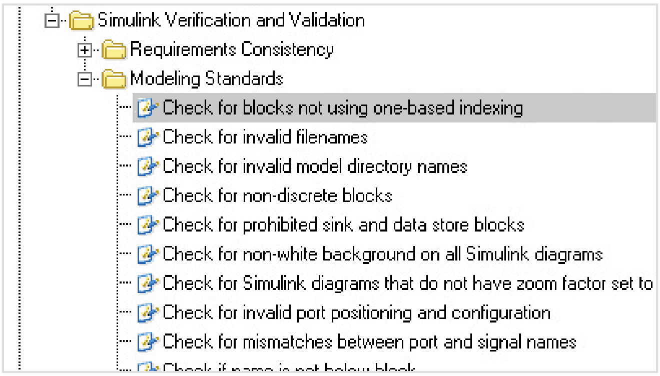
Well-maintained, consistent documentation helps you hold more effective design reviews and share designs among peers. MATLAB and Simulink products automate the creation of design documentation, test-result reports, and traceability reports, and let you build customized reports.

Develop Modeling Standards and Check for Compliance

Apply modeling standards to your design verification process to reduce the number of errors introduced early in the development process. You can create custom model checks or use modeling checks in Simulink Verification and Validation that help you comply with industry standards, such as DO-178B, IEC 61508, and ISO 26262.

Eliminating Design Errors in Your Algorithm Using Simulink Design Verifier




Simulink Verification and Validation
Requirements Consistency
Modeling Standards
Check for blocks not using one-based indexing
Check for invalid filenames
Check for invalid model directory names
Check for non-discrete blocks
Check for prohibited sink and data store blocks
Check for non-white background on all Simulink diagrams
Check for Simulink diagrams that do not have zoom factor set to
Check for invalid port positioning and configuration
Check for mismatches between port and signal names

Checking Modeling Standards Implementation (Article)

Explore Products for Design Verification

Simulink

DO Qualification Kit

IEC Certification Kit

Simulink Design Verifier

Simulink Report Generator™

Simulink Verification and Validation

Simulink Test

Embedded Software Test and Verification

MATLAB and Simulink products support the two standard approaches for verifying embedded software: model-to-software verification and run-time error detection in the source code. With model-to-software verification, a fully verified model of your embedded software serves as a golden reference for comparing its behavior with your handwritten or model-generated software. With run-time error detection, products for run-time analysis apply formal methods on handwritten or automatically generated source code to verify that the code does not have run-time errors. These verification processes are especially important for high-integrity embedded systems.

Verify Software Behavior with the Model

With Model-Based Design, you develop a model of your embedded software using Simulink. After verifying the system model against requirements or expected behavior, you can generate code automatically from the model to reduce the chance of errors introduced through hand coding.

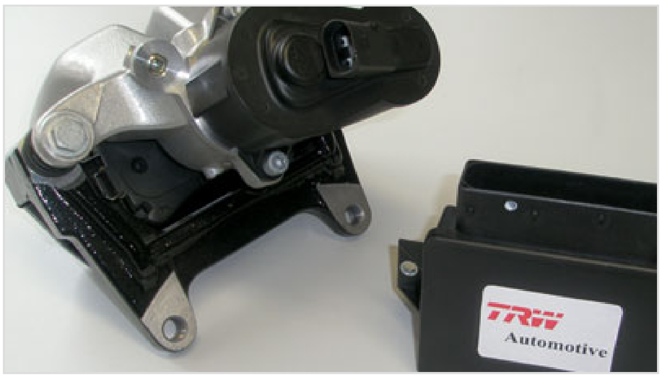
Model-to-software verification techniques like software-in-the-loop (SIL) testing and processor-in-the-loop (PIL) testing can be applied to handwritten or model-generated code to confirm that the behavior of the software matches the behavior of the model. Formal analysis methods available in Simulink Design Verifier automate the generation of SIL and PIL tests from your model, and Embedded Coder helps streamline the PIL test process.

Ensure Code Correctness and Detect Run-Time Errors

With Polyspace® code verification products, detect run-time errors and prove the absence of specific errors in C/C++ and Ada source code, whether handwritten, model-generated, or a combination of the two. Mathematical analysis techniques in Polyspace products prove the absence of overflow, divide-by-zero, out-of-bounds array access, and other run-time errors in source code, without requiring program execution, code instrumentation, or test cases.

Support Industry Standards for High-Integrity Systems, Including DO-178B, IEC-61508, and ISO-26262

Design embedded software for high-integrity systems that meet industry standards with Simulink and related tools. DO Qualification Kit provides documentation, test cases, and procedures that let you qualify Simulink or Polyspace software verification tools for projects based on the DO-178 standard. IEC Certification Kit includes certificates and reports from certification authority TÜV SÜD that are based on documented, application-specific verification workflows to help you use Embedded Coder or Polyspace code verification products for projects based on the IEC 61508 and ISO 26262 standards.



TRW Automotive Develops and Tests Electric Parking Brake (User Story)

Model-Based Approach for ERTMS Railway Wayside System Specification, Validation, and Proof

Explore Products for Embedded Software Test and Verification

Embedded Coder®

Polyspace Bug Finder™

Polyspace Code Prover™

Simulink Design Verifier

Digital and Analog Hardware Verification

A system-level model of your hardware lets you verify the correct behavior of your design against this golden reference. You can optimize design tradeoffs for your digital, analog, or mixed-signal design; verify the impact on system behavior; and reuse the model to verify HDL and circuit implementation.

Explore and Verify Your Design with System Simulation

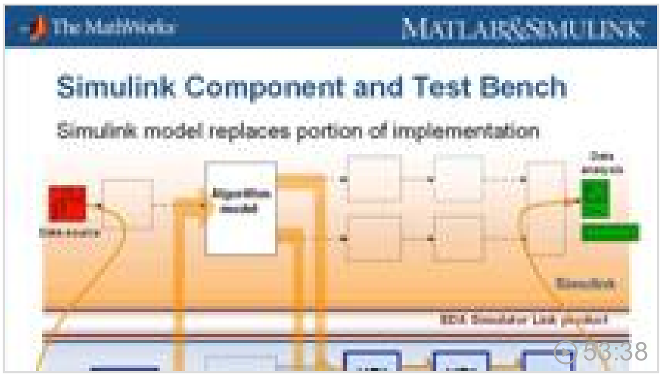
Verify that your algorithm and component designs meet system requirements by using MATLAB and Simulink models to specify behavior, evaluate alternative designs, and integrate components. You can refine designs to create bit-true and timing-accurate reference models for digital and analog hardware components as well as a reusable test bench.

Reuse Models to Verify FPGA and Digital ASIC Implementations

MATLAB and Simulink products help ensure that your HDL implementations of FPGA and ASIC components match the system-level behavior of your model without creating additional test benches. You verify your implementations by cosimulating MATLAB or Simulink models with HDL simulators from Cadence, Mentor Graphics, and Synopsys using HDL Verifier. You can also automatically generate HDL code and test benches for testing hardware designs in an HDL environment and for prototyping on FPGA development boards using HDL Coder and Filter Design HDL Coder™.

Unify Modeling and Verification of Analog and Mixed-Signal Systems

Use Simulink and related products to refine and verify behavioral models of analog and mixed-signal components. You can create designs to handle nonlinearities, timing jitter, and other impairments associated with analog hardware. By cosimulating your models with analog simulation tools from Cadence and Mentor Graphics, you can verify that the circuit implementation matches the golden reference.


The MathWorks MATLAB&SIMULINK

Simulink Component and Test Bench

Simulink model replaces portion of implementation

HDL Functional Verification with MATLAB and Simulink

Explore Products for Digital and Analog Hardware Verification

Simulink

HDL Verifier™

Fixed-Point Designer™

SimPowerSystems™

HDL Coder™

Hardware-in-the-Loop Simulation for Embedded Systems

Hardware-in-the-loop (HIL) simulation minimizes costs and risks in embedded system development by enabling you to test your embedded system before deploying it in a production environment. You test your embedded system in real time by connecting it to a simulation of the remaining design. This approach enables you to use the same system-level models throughout the development process, from design to real-time HIL simulation.

Test Embedded Systems with Real-Time System Simulation

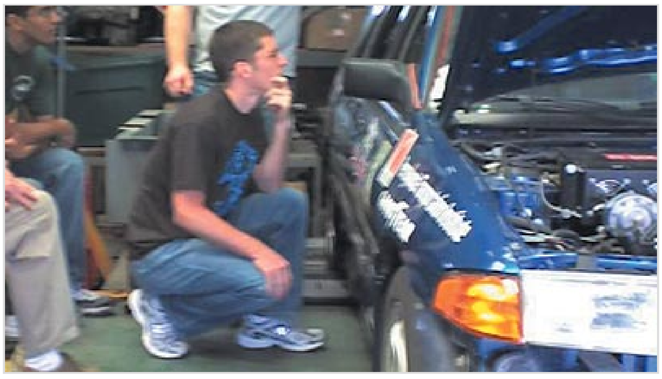
HIL simulation begins with a system-level model that includes your embedded system algorithm and its operating environment. You can automatically generate C code and HDL code from the plant and environment models to run on a real-time simulator that delivers inputs and receives outputs from the embedded system as the real system would. As a result, you obtain greater value from the system-level model for testing and verifying the real-time performance of your embedded system.

HIL simulation is especially valuable when:

- The system is not yet built.
- Safety and performance concerns dictate testing the system prior to human involvement.
- You need to minimize expensive downtime for the real system.
- You need to test operation and failure conditions that are difficult to physically replicate.

Verify Algorithm Behavior on FPGAs Using Simulink Cosimulation

FPGA HIL simulation enables you to verify FPGA implementations of algorithms by leveraging a Simulink system model as a virtual test harness. Blocks in your Simulink system model define connections to an FPGA development board running handwritten or automatically generated HDL code. During simulation, signals from the system model are automatically routed to the FPGA, where the algorithm runs, and the signals from the FPGA return to the next Simulink block. Products from Altera, Xilinx, and MathWorks automatically generate the interconnectivity for FPGA HIL simulation.



Automotive Research Lab at Penn State Gives Students Practical Hardware-in-the-Loop Experience (User Story)

Cessna Enhances Antiskid Technology with Hardware-in-the-Loop Testing (User Story)