

SystemTest 2.6.1

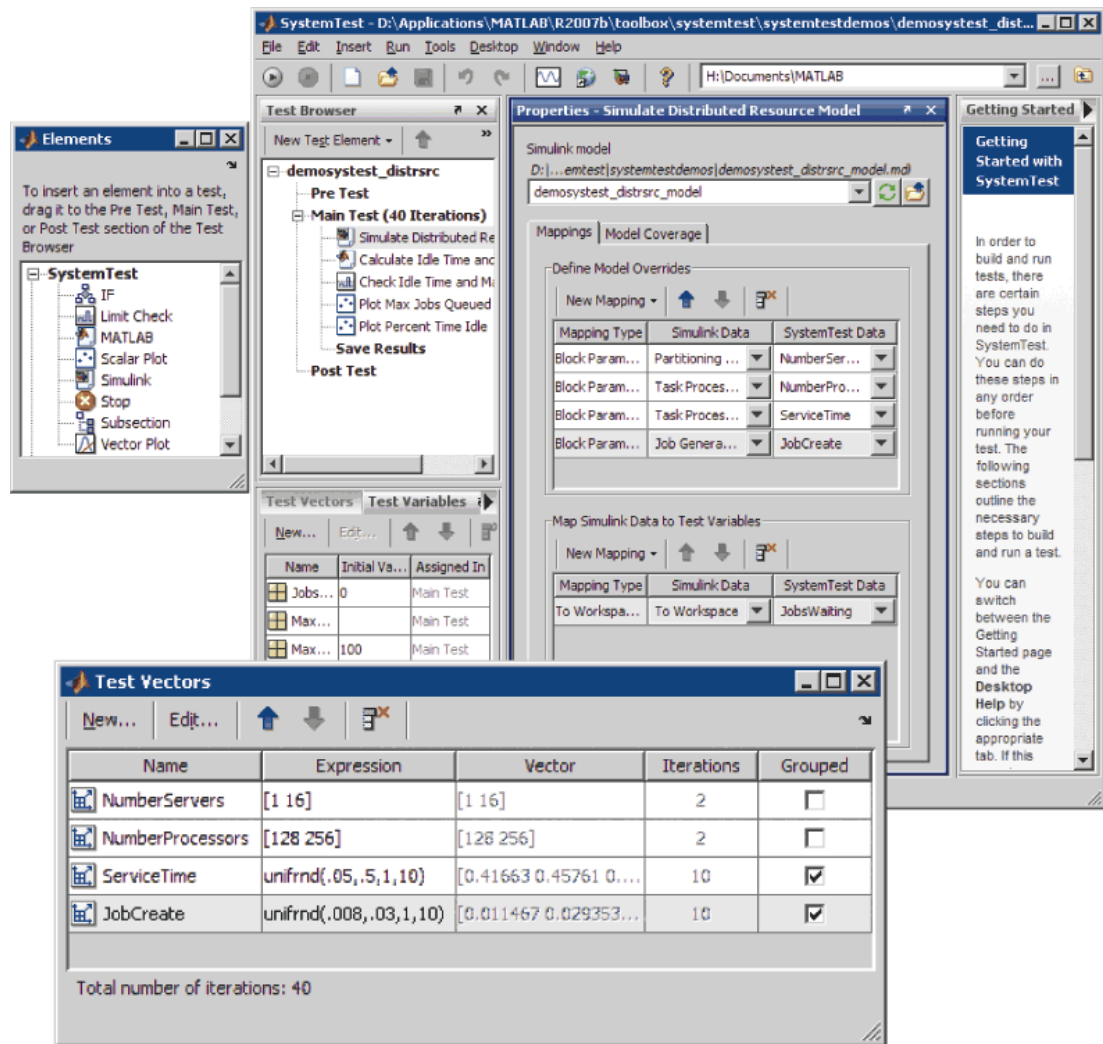
Manage tests and analyze results for system verification and validation

SystemTest™ lets you develop and execute tests that exercise MATLAB® algorithms and Simulink® models. It includes predefined test elements that let you build and maintain standard test routines. You can map test variables into a result set for analysis.

SystemTest lets you design tests that become part of the complete design specification. You can save and share tests throughout a development project, ensuring standard and repeatable test verification from research and development through preproduction. When used with Parallel Computing Toolbox™ (available separately), SystemTest can run test iterations on multiple processors or machines.

Key Features

- Develops, manages, and edits test structures using predefined test elements
- Stores tests independently of the model under test, for repeatable test execution
- Defines pass/fail criteria for tests using Boolean constraints and tolerance limits
- Generates random test vector values using probability distribution functions
- Runs iterations of Simulink models on multiple processors with Parallel Computing Toolbox (available separately)
- Generates reports of test execution and results
- Visualizes and analyzes multidimensional test result



The SystemTest desktop showing an example of a Simulink model being tested using the test vectors defined in the Test Vector pane. You can build a test structure using predefined Test Elements.

Working with SystemTest

SystemTest helps you develop and run your test and analyze results. You can run a TEST-file from the MATLAB command line or the graphical user interface (GUI). From the GUI you can:

- Develop a test structure from predefined test elements
- Create test vectors using MATLAB expressions or probability distributions
- Select Simulink inputs or block parameters to vary
- Define pass/fail criteria
- Generate a test execution report
- Map test variables to a result set for subsequent analysis

Building a Test

SystemTest tests consist of elements, test vectors, and test variables. You can use each of these entities to create a variety of test scenarios ranging from a simple test that runs a series of elements once to a full parameter sweep that iterates over the values of test vectors that you define. Test vectors and test variables map data between

SystemTest and the model or unit under test. You can optionally map test variables to results and perform analysis using the Test Results Viewer.

Test elements are discrete actions that run during test execution. You can use the test elements to integrate capabilities from other MathWorks products into SystemTest. SystemTest includes the following test elements:

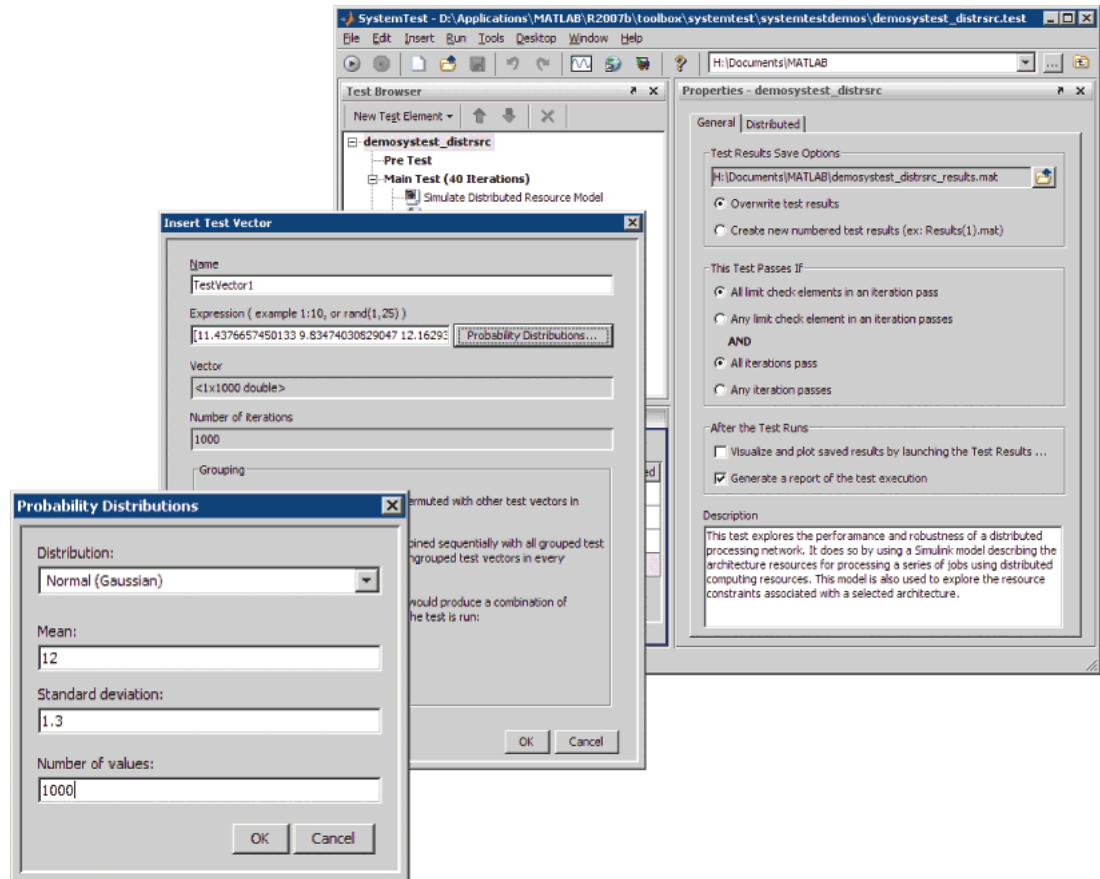
- MATLAB, which executes MATLAB code
- Simulink, which connects the test to a Simulink model and uses test vectors to sweep across the model's inputs and parameters
- Limit Check, which specifies pass/fail conditions
- Vector Plot, which plots array or vector data as the test is executing
- Scalar Plot, which plots scalar data for each iteration
- IF, which controls the flow of a test based on a logical condition
- Stop, which stops the test execution
- Subsection, which helps manage complex test structures

Detailed information about [additional test elements](#) for SystemTest is available.

Test vectors specify the values that vary during each test iteration. SystemTest lets you use multiple test vectors whose values are arguments (for algorithms) or parameters and input signals (for models). You can define test vectors using any MATLAB expression or generate random values using probability distributions.

SystemTest includes normal (Gaussian) and uniform probability distribution functions. When used with Statistics Toolbox (available separately), SystemTest lets you also specify exponential, gamma, lognormal, Student's t , or Weibull probability distributions.

Test variables are used to store data for use by test elements during execution. You can initialize variables in the Pre Test or Main Test with any MATLAB expression.



Test Vector values generated using a probability distribution for Monte Carlo testing.

Executing Tests

When you run a SystemTest TEST-file from the GUI or using the `strun` command, it executes using the test structure you defined using test elements. The test structure is defined in three sections.

Pre Test runs once prior to Main Test and can be used for:

- Initializing variables
- Opening a model
- Accessing system resources
- Initializing external test equipment

Main Test runs one or more times based on the number of iterations defined by the test vectors. SystemTest will execute each test element in Main Test in sequence once during each iteration.

Post Test runs once after all Main Test iterations are complete or when a test stops during execution. It can be used for:

- Cleaning up your workspace
- Closing a model
- Releasing system resources
- Returning external test equipment to a safe state

The Run Status pane indicates which section is executing and how long the test has been running, and provides cumulative Pass/Fail status for the test iterations. After the test is complete, the pane displays the location of any generated files, any error messages generated during the test, and the start and stop time.

When used with Simulink Verification and Validation™ (available separately), SystemTest lets you view model coverage reports. These reports let you see how multiple iterations increase model coverage. Results are reported in HTML documents using industry-standard metrics for structural coverage.

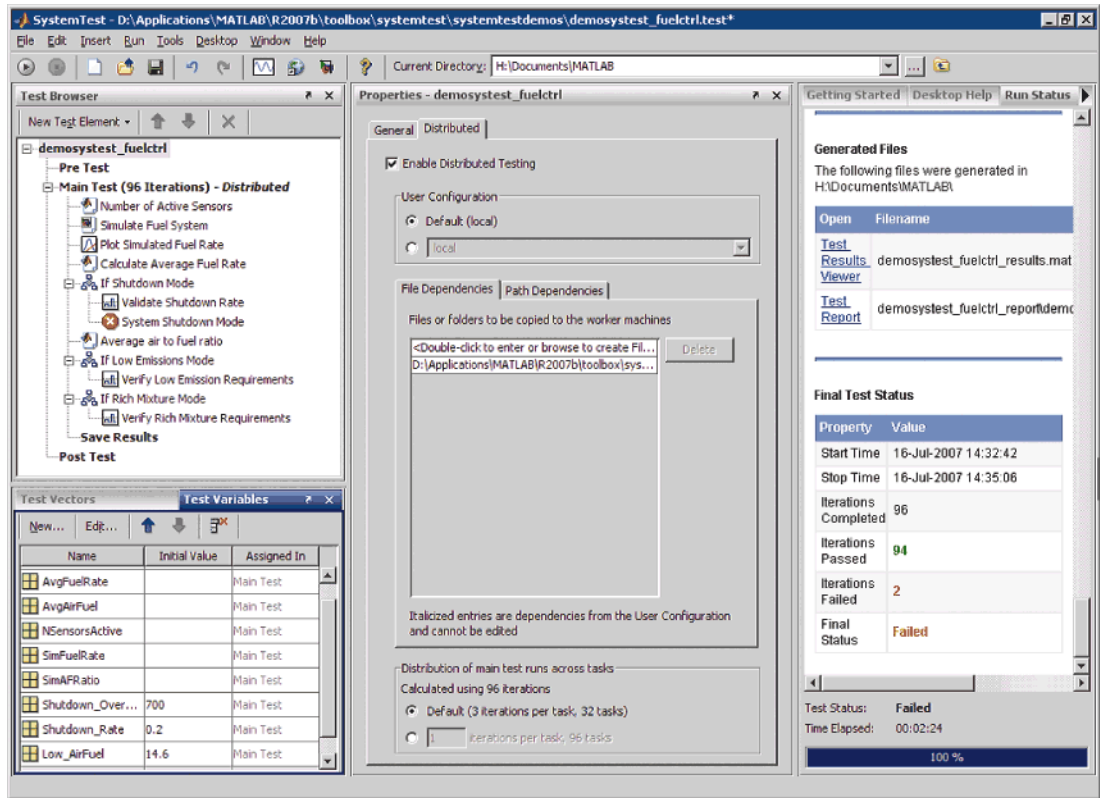
Distributing a Test

SystemTest combined with Parallel Computing Toolbox™ allows you to distribute your test to multiple processors. These processors can reside in one multiprocessor computer or, when the toolbox is used with MATLAB® Distributed Computing Server™ (available separately), on a computer cluster. You can distribute Main Test iterations as tasks, which are performed simultaneously by different processors.

To distribute a test, you select a check box in the Distributed tab of the Test Properties pane. You can also define any files that your test depends upon using the File or Path Dependencies tables. No changes to the existing TEST-file or additional lines of code are necessary. When you run a distributed test, the test executes in this way:

- Pre Test runs once on the client machine.
- Main Test iterations execute on the workers.
- Test results are saved at the end of each Main Test iteration and returned to the client machine once all Main Test iterations have finished executing.
- Post Test runs once on the client machine.
- Generated plots are not shown on the client machine while the test runs, but are captured and displayed in the Test Report.

Distributing a test to multiple processors or machines can speed up the total time a test takes to execute. This can enable you to run more test cases in the same amount of time as a test run on one processor.



The SystemTest desktop displaying test results obtained on multiple processors using Parallel Computing Toolbox.

Analyzing Test Results

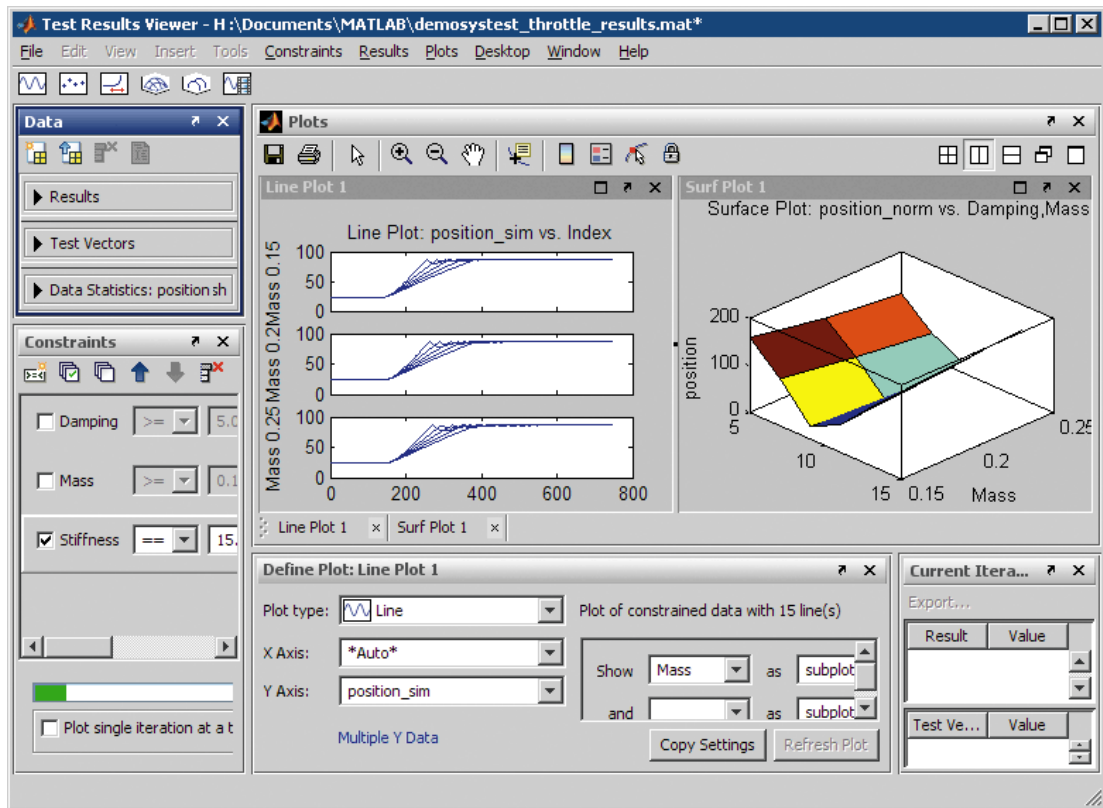
The Test Results Viewer helps you manage, visualize, inspect, and analyze the data in your test results using multiple plot types, including line, surface, scatter, time series, waterfall and image.

For post-test analysis, you can:

- Interactively plot and explore multidimensional results
- Filter and constrain results for further analysis

Within these plots, you can distinguish the effects of test vectors on the result set using color, data markers, and subplots. Plots let you graphically select data associated with test iterations. Details associated with the selection display in the Current Iteration pane.

The Test Results Viewer lets you use multiple plots simultaneously to examine different views of data generated from the same test.



The Test Results Viewer displaying a result set. You can analyze results using multiple plotting techniques and interactive constraints.

Resources

Product Details, Demos, and System Requirements

www.mathworks.com/products/systemtest

Trial Software

www.mathworks.com/trialrequest

Sales

www.mathworks.com/contactsales

Technical Support

www.mathworks.com/support

Online User Community

www.mathworks.com/matlabcentral

Training Services

www.mathworks.com/training

Third-Party Products and Services

www.mathworks.com/connections

Worldwide Contacts

www.mathworks.com/contact