**MathWorks Products:**

**Required**
- MATLAB
- Simulink
- Embedded Coder
- MATLAB Coder
- Simulink Coder

**Recommended**
- Polyspace Client for Ada
- Polyspace for C/C++

**Dynamic embedded software testing integrated for SIL, PIL, and Polyspace® code verification**

**Highlights**
- Measure industry-specific levels of code coverage from model execution
- Build unit-test environments automatically from Embedded Coder™ generated code
- Create VectorCAST™ unit tests automatically from Simulink® test vectors
- Run Polyspace® static analysis from within the VectorCAST dynamic test environment

**Make Embedded Software Testing More Reliable**

Reduce the time, effort, and cost to validate your software for safety-critical industries such as avionics, medical devices, automotive, industrial controls, and railway. Rigorous testing is more than just good software development practice; it is a critical part of the software certification process.

Traditionally, embedded software testing is a tedious and labor-intensive process. VectorCAST automates the tasks associated with unit, integration, and system testing of C, C++, and Ada applications, resulting in measurable reductions in cost and measurable improvements in quality.
Capture Source Code Coverage During Software-in-the-Loop Testing

Software-in-the-loop (SIL) simulation captures the behavior of the generated C code in the simulator environment of the target controller. The simulated results and the model-in-the-loop (MIL) simulation results should be identical. If they are not, the results can be used to evaluate the cause of deviation.

When developing software for a safety-critical standard, one of the artifacts required is structural code coverage. The VectorCAST/Cover™ integration with Simulink allows users to measure code coverage from the source code that has been executed when conducting SIL testing.

Unit-Test Generated Source Code Modules from Simulink with VectorCAST/C++

VectorCAST/C++™ will work with code generated by Embedded Coder from Simulink models. VectorCAST can analyze the Simulink model to automatically build a unit-test framework for the generated code. It can then analyze the execution data from Mil testing to create unit-test cases for validating the generated code inside a unit-test environment. The resulting VectorCAST environments can then be expanded for testing additional scenarios that were impractical in MIL testing.

Finally, the unit-test environments can easily be used for SIL or processor-in-the-loop (PIL) testing seamlessly. Any updates to the Simulink model or the associated data to validate the model can automatically be converted into VectorCAST test cases.

Combine Polyspace Static Analysis and VectorCAST Dynamic Testing

VectorCAST integration with Polyspace products leverages the natural synergies between dynamic testing and static analysis by combining full test automation with formal methods-based static analysis to empower engineers with advanced verification and validation capabilities.

VectorCAST collects a complete profile of the test environment, including source file location and compiler options, and leverages that information to launch the Polyspace code verifier. After launch, the Polyspace code verifier performs an exhaustive static analysis-based verification for the presence—or absence—of run-time errors in the code.

Potential errors found by Polyspace products can be used by VectorCAST to automatically build the test harnesses and test cases necessary to isolate and validate the code. Polyspace products also provide function and variable range data that VectorCAST can use to ensure code coverage. The VectorCAST integration works with Polyspace Client™ for C/C++, Polyspace Server™ for C/C++, Polyspace Client for Ada, and Polyspace Server for Ada.