Polyspace Model Link TL 5.6
for dSPACE TargetLink

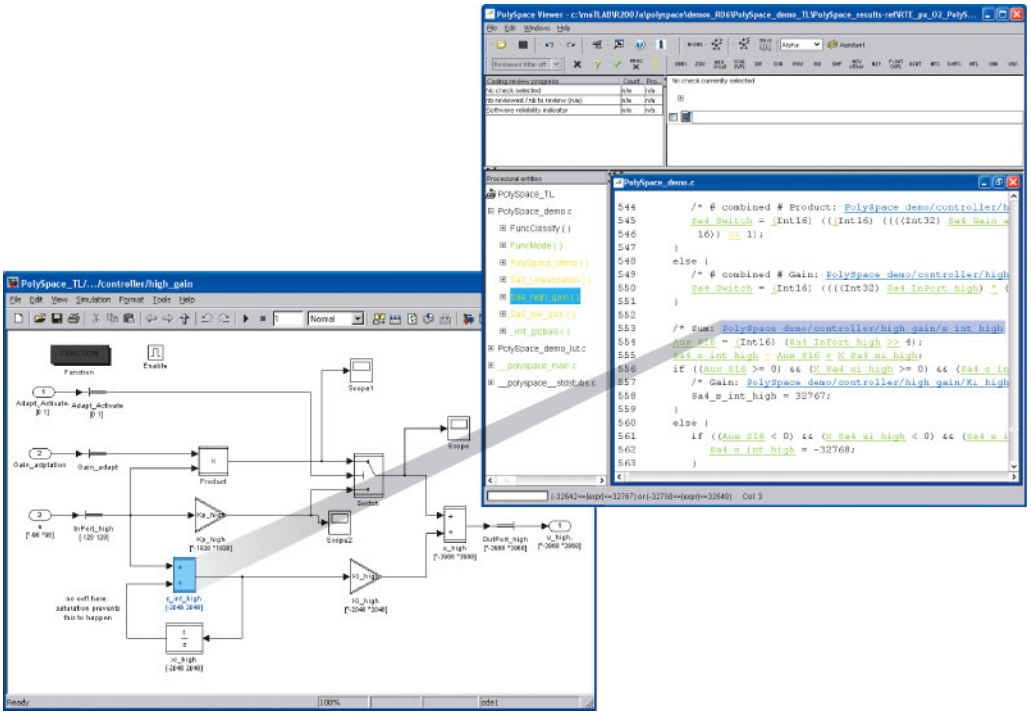
Trace Polyspace results to dSPACE TargetLink blocks

Polyspace Model Link™ TL extends Polyspace Client™ for C/C++ and Polyspace Server™ for C/C++ with tools that let you verify C code generated by TargetLink® and trace Polyspace® results from the generated C code to your model. As a result, you can identify which parts of the model are reliable, and correct design problems that will cause run-time errors in the code. With Polyspace Model Link TL, you work in the Simulink environment to verify C code generated by TargetLink. You can verify a mix of generated and hand-written code before it is compiled.

You can start the code verification directly from within your TargetLink model, using context-sensitive predefined settings.

Key Features

- Uses advanced code-based verification techniques to automatically verify all program executions
- Incorporates the entire range of parameter values specified in the model
- Traces run-time errors back to the model
- Launches from within TargetLink



Polyspace Client for C/C++ GUI showing a Simulink model of a high-gain control system (left) and the Polyspace results of the C code generated from the Simulink model by TargetLink (right). PolySpace Model Link TL highlights comments in the Polyspace results that link back to the corresponding block in the model.

Working with Polyspace Model Link TL

Polyspace Model Link TL integrates with Polyspace Client for C/C++ and Polyspace Server for C/C++ (both available separately) to support the model development workflow. You can:

- Fix implementation errors caused by latent design deficiencies
- Independently verify generated code
- Determine how robust the code is on a particular target
- Correct errors during implementation, before deployment and testing

You can verify a subsystem or any reusable component of your system. You can quickly navigate from the code to the relevant section of your TargetLink model, facilitating design edits and debugging.

Typical Run-Time Errors Detected

In Models

- Overflows and underflows
- Division by zero and other arithmetic errors
- Out-of-bounds array access
- Dead code

In Code

- Illegally dereferenced pointers
- Read-only access to noninitialized data
- Dangerous type conversions

Resources

Product Details, Demos, and System Requirements
www.mathworks.com/products/polyspacemodelitl

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