Verification and Validation

Verifying and validating embedded systems comprising software and electronics is a daunting challenge, given the increasing complexity of these systems and the need to meet tight schedules. Aerospace, automotive, communications, mechatronics, and other organizations solve this challenge by starting verification as soon as they have system models and then reusing models, data, and other artifacts to verify the embedded-software or HDL implementation.

YOKOGAWA ELECTRIC CORPORATION

Reusing Simulink test benches for HDL implementations

Yokogawa Electric Corporation developed two key optical network components: an ultra-high-speed optical switching device and an optical packet transmitter/receiver that provides an interface between the optical packet network and conventional networks. Yokogawa used Simulink® models to identify and correct design errors during simulation. When the components were implemented, they reused their Simulink system-level test benches to test the HDL implementations with Simulink connected to Mentor Graphics® ModelSim® via EDA Simulator Link™ MQ. Models and test data were reused for verification.

www.mathworks.com/nn8/yokogawa

ASTRIUM

Generating code from control-system and physical models for HIL and flight tests

Astrium, a subsidiary of EADS, developed controls for a laser optical link between an aircraft and a communication satellite capable of 50 megabits-per-second transmission with an error rate of less than one error bit per billion. Using Simulink, Real-Time Workshop®, and xPC Target™, they developed a control system model and a physical model of the pointing hardware. The models were used for continuous functional verification and to generate code for HIL tests and for the real-time demonstration system used in flight tests.

www.mathworks.com/nn8/astrium

MAN ROLAND

Integrating models and data for performance verification

To enhance print quality on its state-of-the-art printing press, MAN Roland used MATLAB® and Simulink models of both the controller and the press. To ensure that the model of the press reflected the system's behavior, they incorporated performance data collected from an operating press. They used the plant model to simulate abnormal press behavior, which is often difficult to replicate on the actual system. This approach enabled them to design a more robust controller, which was then generated and deployed using Real-Time Workshop and xPC Target.

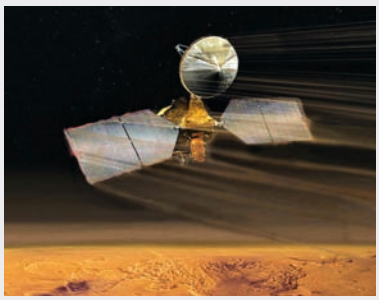
www.mathworks.com/nn8/man_roland

LOCKHEED MARTIN SPACE SYSTEMS

Using plant models for both design and custom HIL test systems

To assist in designing the Mars Reconnaissance Orbiter (MRO) satellite's guidance, navigation, and control system, Lockheed Martin Space Systems used Simulink and SimMechanics™ to model the plant, including actuators, sensors, control algorithms, and vehicle dynamics. They used code generated by Real-Time Workshop from those models to configure their in-house HIL system, Orbiter Test Bed, for real-time testing of the MRO.

www.mathworks.com/nn8/lockheed_mro

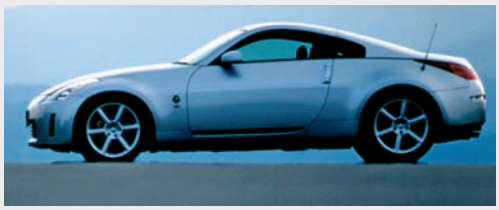


NISSAN MOTOR COMPANY

Streamlining code review processes with suppliers

The Software Quality Group at Nissan Motor Company used PolySpace™ products to improve the quality of supplier-provided software. Previously, quality goals were defined mainly in terms of classical techniques based on unit, integration, or system tests. Now, a more accurate and efficient software review process based on PolySpace software is used with suppliers at the beginning of the project and again at each major milestone. In addition, many Nissan suppliers are working to integrate PolySpace products into their own internal development processes.

www.mathworks.com/nn8/nissan

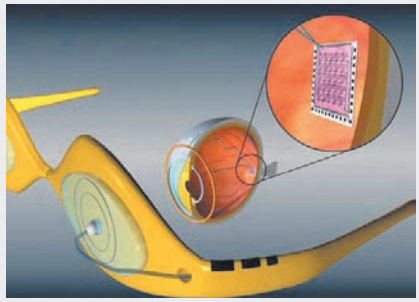


DOHENY EYE INSTITUTE

Debugging on-chip DSP code

Doheny Eye Institute researchers developed a retinal prosthesis that converts images into stimulus signals that are delivered to retinal nerve cells through an implanted electronic chip. The researchers used MATLAB and Simulink to test ideas and develop image and video processing algorithms, then used Real-Time Workshop to generate C code implementations and download them onto a TI DSP. The code on the DSP chip was verified and debugged using MATLAB and Code Composer Studio™, connected using Embedded IDE Link™ CC.

www.mathworks.com/nn8/doheny



Resources

VERIFICATION, VALIDATION, AND TEST IN MODEL-BASED DESIGN
www.mathworks.com/nn8/vv

USER STORIES *www.mathworks.com/nn8/user_stories*

The MathWorks™

91608v00 10/08