Real-Time Testing Made Easy with Simulink Real-Time

Andreas Uschold
Application Engineer
MathWorks

Martin Rosser
Technical Sales Engineer
Speedgoat
Model-Based Design
Continuous Verification and Validation

Requirements

System Design
- Environment
- Physical Components
- Algorithms

System-Level Specification

Component Design

Subsystem Design

Implementation
- Embedded Software
  - C++, VHDL, Verilog
- Digital Electronics
  - MCU, DSP, FPGA, ASIC

Integration testing

Code Verification and Validation

Subsystem Implementation

Subsystem Integration & Test

System-Level Integration & Test

User Acceptance Testing

Complete Integration & Test
Model-Based Design
Continuous Verification and Validation

Requirements

System Design
- Environment
- Physical Components
- Algorithms

System-Level Specification

Rapid Prototyping

Verification and Validation

HiL Simulation

User Acceptance Testing

Complete Integration & Test

System-Level Integration & Test

Code Verification and Validation

Subsystem Integration & Test

Subsystem Implementation

Implementation

Subsystem Design

Generate

Embedded Software
- C
- C++
- VHDL, Verilog

Digital Electronics
- MCU
- DSP
- FPGA
- ASIC
Simulink Real-Time Enables Simulation and Testing

Rapidly create real-time applications from Simulink models and run and test them with your hardware under test at normal operating frequencies, speeds, and timing.
Real-Time Simulation and Testing Tasks:

**Rapid Control Prototyping**

![Diagram of System Model with Controller Model, Test Suite, and Hardware connections]

- **Target Computer Hardware**
- **Physical Plant Hardware**
- **Wiring and Signal Conditioning**

- 1-Click Code Generation and Download

- I/O Output
- I/O Input
- Verification

**Real-Time Simulation and Testing Tasks:**

- **Real-Time Simulation**
- **Testing Tasks:**
  - **Rapid Control Prototyping**
  - **Wiring and Signal Conditioning**

**Diagram Elements:**

- **System Model**
- **Controller Model**
- **Test Suite**
Real-Time Simulation and Testing Tasks: 
*Hardware-in-the-loop (HIL) Simulation*

![Diagram of HIL Simulation process]

- **Controller Model**
- **Plant Model**
- **Test Suite**
- **Verification**

**Code Generation and Download**

**Wiring and Signal Conditioning**

**Embedded Controller Hardware**

**Target Computer Hardware**
Additional Real-Time Simulation and Testing Tasks: 
*Parametric Evaluation and Performance Assessment*

- **System robustness**
  - Monte Carlo analysis
  - Operational envelope testing

- **Human factors**
  - Human-in-the-loop simulation
  - Virtual reality simulators

- **Calibration**
  - Tune algorithmic coefficients
  - Optimize performance
How does Simulink Real-Time work?

*From desktop simulation to real time*

Creation of real-time applications from Simulink models and loading them onto dedicated target computer hardware in 3 automated steps:

1. **Code Generation**
2. **Compile and Link**
3. **Download and Ready to Run**
Instrument your Real-Time Applications

- Simulink Real-Time provides a number of UI options that enable you to communicate with and control real-time applications running on the target machine:
  - Simulink Real-Time Explorer – `slrtexplr`
  - Simulink External Mode
  - Simulink Real-Time APIs
  - MATLAB UIs
  - MATLAB command scripts

- Use one or more of these tools based on your workflow and interface requirements.
Demonstration – Simulink Real-Time Explorer
Demonstration – Simulink External Mode
Model xpcosc:
simple xPC Target demo model
Demonstration – Batch Simulation
Control and Instrument Your Real-Time Application

*Built-in control and monitoring User Interface*
Create UIs in MATLAB

Graphical front end for your MATLAB scripts and apps
Simulink Real-Time Deployment

- Runs independently of Simulink
- One Simulink/Simulink Real-Time Target license
- Many target machines

- Field deployment
- Complete controller products
- Stand-alone HIL solutions
- Etc.
Speedgoat Real-Time Target Machines

Assembled based on your technical requirements

- Form factors available for office, lab, field, and classroom use
- Optimized for highest real-time performance (Multicore CPUs and FPGAs)
- Fully tested and works out-of-the-box
- Flexible, expandable architecture supporting a wide range I/O connectivity

* Custom engineering and I/O module development available
Demonstration – Controller Algorithm with Generated Encoder and PWM Peripherals
Example application

**Rapid motor control prototyping**

- Flexible real-time prototyping platform
- Use of CPU and FPGA
- PWM signal generation and encoder capture
- Synchronization of PWMs, current sensing, and model

![Rapid motor control prototyping diagram](image-url)
Hardware solutions

Leverage the FPGA without needing to know HDL coding

Different use cases

1. Flexible FPGA Code Module functionality
   - PWM generation and capture
   - Incremental and absolute encoders
   - SPI and I2C protocol support
   - And much more

2. Accelerate parts of your Simulink model's real-time application using automated HDL code generation

All based on the same hardware!
Pushing limits: More components, faster control algorithms
Example of partitioning controller for disparate rates

1 kHz Rate
- Velocity control
- Mode scheduler
- Encoder calibration

10 kHz Rate
- Field oriented control
- ADC to current
- Encoder to position
- Position to velocity
- Voltage to PWM

50 MHz Rate
- PWM peripheral
- Encoder peripheral
Where should a 50 kHz control loop be implemented?

- **Microprocessor**
  - 1 kHz Rate
    - Velocity control
    - Mode scheduler
    - Encoder calibration

- **50 kHz Rate**
  - Field oriented control
  - ADC to current
  - Encoder to position
  - Position to velocity
  - Voltage to PWM

- **FPGA**
  - 50 MHz Rate
    - PWM peripheral
    - Encoder peripheral
What else do we need to be aware of to implement the 50 kHz rate on the FPGA?

**Microprocessor**
- 1 kHz Rate
  - Velocity control
  - Mode scheduler
  - Encoder calibration

**FPGA**
- 50 kHz Rate
  - Field oriented control
  - ADC to current
  - Encoder to position
  - Position to velocity
  - Voltage to PWM
- 50 MHz Rate
  - PWM peripheral
  - Encoder peripheral
Examples of Speedgoat FPGA-based I/O modules for use with HDL Coder

Example: IO323
- Spartan 6 FPGA, 100k
- 32 x SE/16 DF AD
- 08 x DA
- 48 x configurable TTL
- MGT support

Latest Xilinx technologies available
- Artix 7
- Kintex 7
- Kintex Ultrascale
- ...

Connect multiple FPGAs
- MGTs (Multi-Gigabit Transceiver) for inter-module communication
- Front SFP connectors for inter-system communication
“With the Speedgoat system, changing parameters and tuning the system is very easy and straightforward. It saves us a lot of time.”

“There is no need to re-compile and burn each new version of the control algorithm.”

Eyal Bagon
Senior Director Autonomous Vehicle
Mobileye
User Story Examples: Various disciplines

Proterra, Greenville, SC, USA
Zero-Emmission Battery Electric Bus
Hardware-in-the-Loop simulation

Levant Power, Woburn MA, USA
Energy neutral active suspension system
In-vehicle Rapid Controller Prototyping

AGCO, France/Germany/USA
Agricultural vehicles with most energy efficient gearboxes
Hardware-in-the-loop simulation

www.speedgoat.ch/userstories
Speedgoat Target Machine and I/O Offerings

- **Performance real-time target machine**
  - Office and lab

- **Mobile real-time target machine**
  - Field and in-vehicle use

- **Education real-time target machine**
  - Academic use

- **Analog and digital I/O**

- **Reconfigurable FPGA I/O**

- **Protocol interfaces**
Typical I/O used in Automotive Industry

Protocol support
- CAN / LIN / FlexRay
- Shared memory
- SPI / I2C

Passive components
- High precision resistors (thermocouples / RTD)
- Strain gauges

Encoder emulation
- Absolute (SSI) / Incremental
- EnDat, BiSS
- Synchro/Resolver, LVDT/RVDT

Fault insertion
- Wide range of channel counts / fault bus configurations
- Designed for safety critical applications

Timing applications
- Precision Timing Protocol (IEEE1588), GPS, IRIG
Speedgoat Real-Time Target Machines

*Designed for Simulink Real-Time, tailored to your needs*

Speedgoat develops and sells real-time target machine solutions consisting of:
- Real-time target machine
- I/O modules
- Software drivers, cables and tools to connect with a prototype
Simulink Real-Time

Summary

- Seamless integration into MATLAB/Simulink
- An easy way to test your application in a real-time environment
- Runs on Speedgoat target hardware