

# Control System Design and Rapid Prototyping Using Simulink

**Chirag Patel**

**Sr. Application Engineer – Modeling and Simulink**

**MathWorks India**

Are you using different tools for  
design & real-time testing?

Do you make design changes  
during field & real-time testing?

Is your test data under utilized  
or not fully leveraged for  
design improvements?

# Solution?

# Integrated Design Workflow

Based on

Control Design Products and  
Simulink Real-Time

# Agenda

## Tuning Controller Parameters

- Classical control tuning techniques
- Optimization-based system response tuning

## Real-Time Testing and Simulation

- Prepare models for real-time execution
- Connect models with hardware under test using flexible real-time testing hardware
- Tune parameters and log\monitor signals during real-time execution using Simulink Real-Time™ Explorer
- Automate real-time test routines using MATLAB® scripts

# Agenda

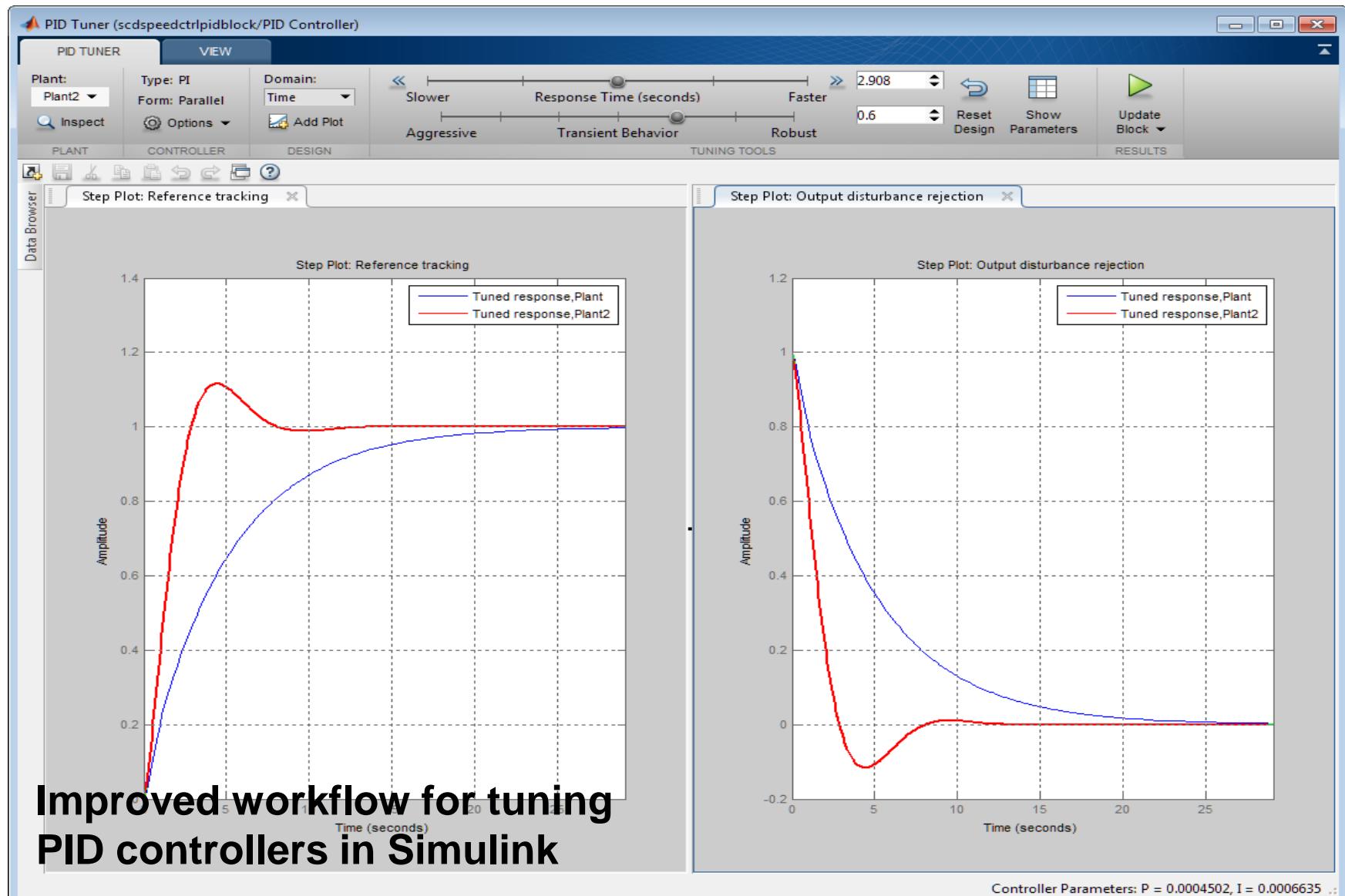
## Tuning Controller Parameters

- **Classical control tuning techniques**
- Optimization-based system response tuning

## Real-Time Testing and Simulation

- Prepare models for real-time execution
- Connect models with hardware under test using flexible real-time testing hardware
- Tune parameters and log\monitor signals during real-time execution using Simulink Real-Time™ Explorer
- Automate real-time test routines using MATLAB® scripts

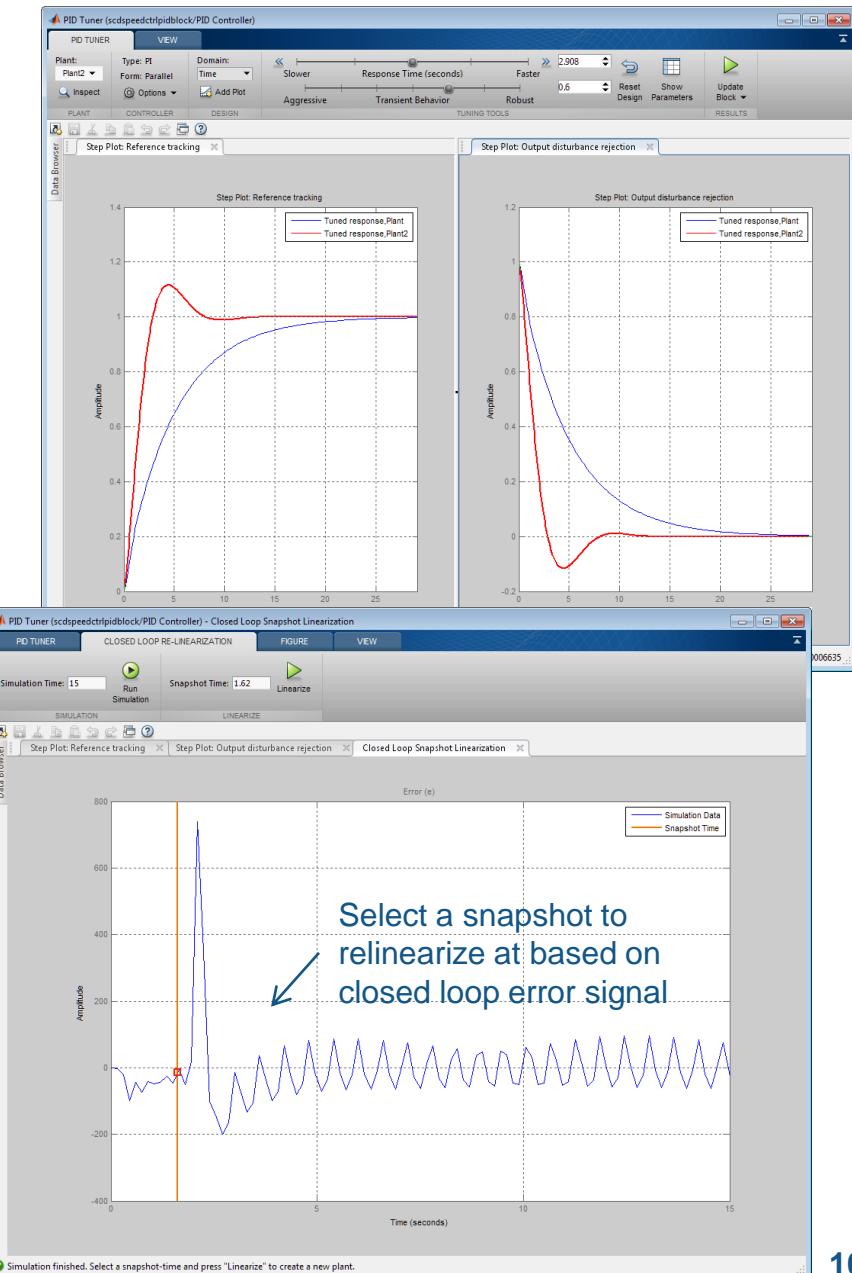
# Tuning Commonly used PID Controllers



# Redesigned PID Tuner

Improved workflow for tuning  
PID controllers in Simulink

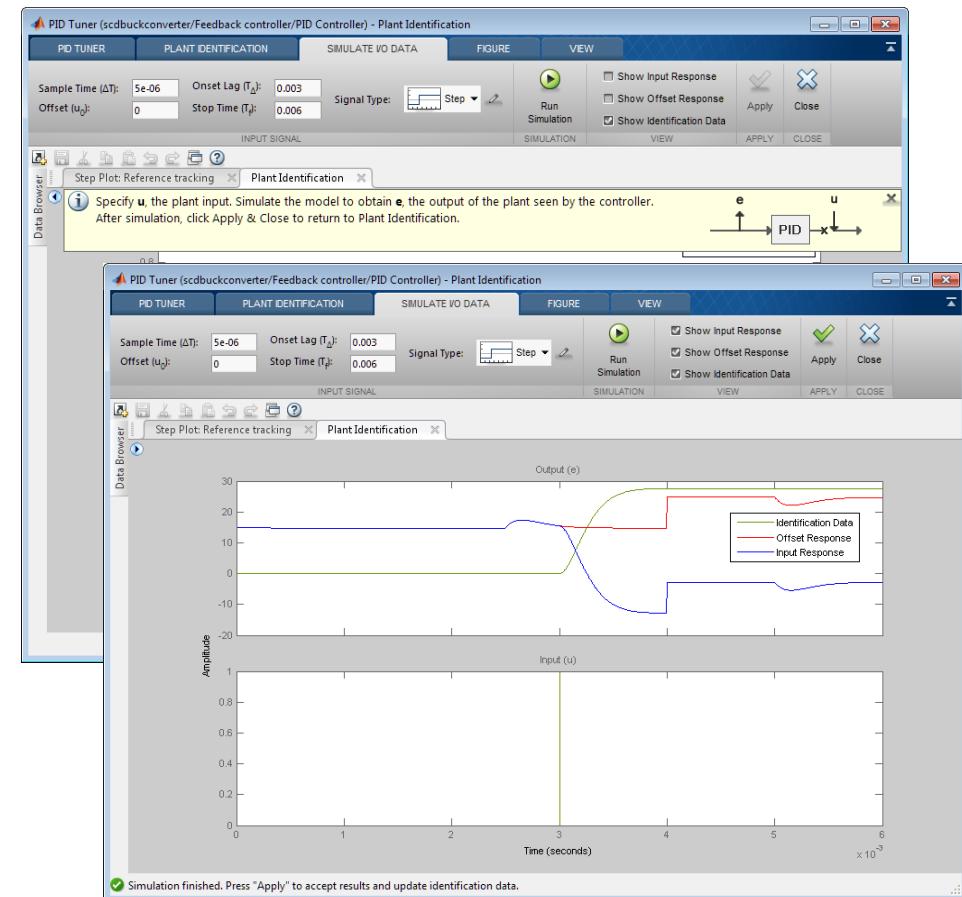
- Several response plots can be displayed simultaneously
- Controller can be evaluated against several plant models
- Model can be relinearized at a visually selected simulation snapshot



# System Identification Integrated into PID Tuner in Simulink Control Design

**Tune PID Controllers for Simulink models with discontinuities such as PWM and Stateflow logic**

- Compute plant transfer function from simulation input-output data when exact linearization fails
  
  
  
- Inject a step or an impulse at the plant input
  
  
  
- Interactively or automatically fit the transfer function to simulation input-output data



# Tuning Complex Fixed Structured Controllers

**Tune fixed-structure controllers in Simulink**

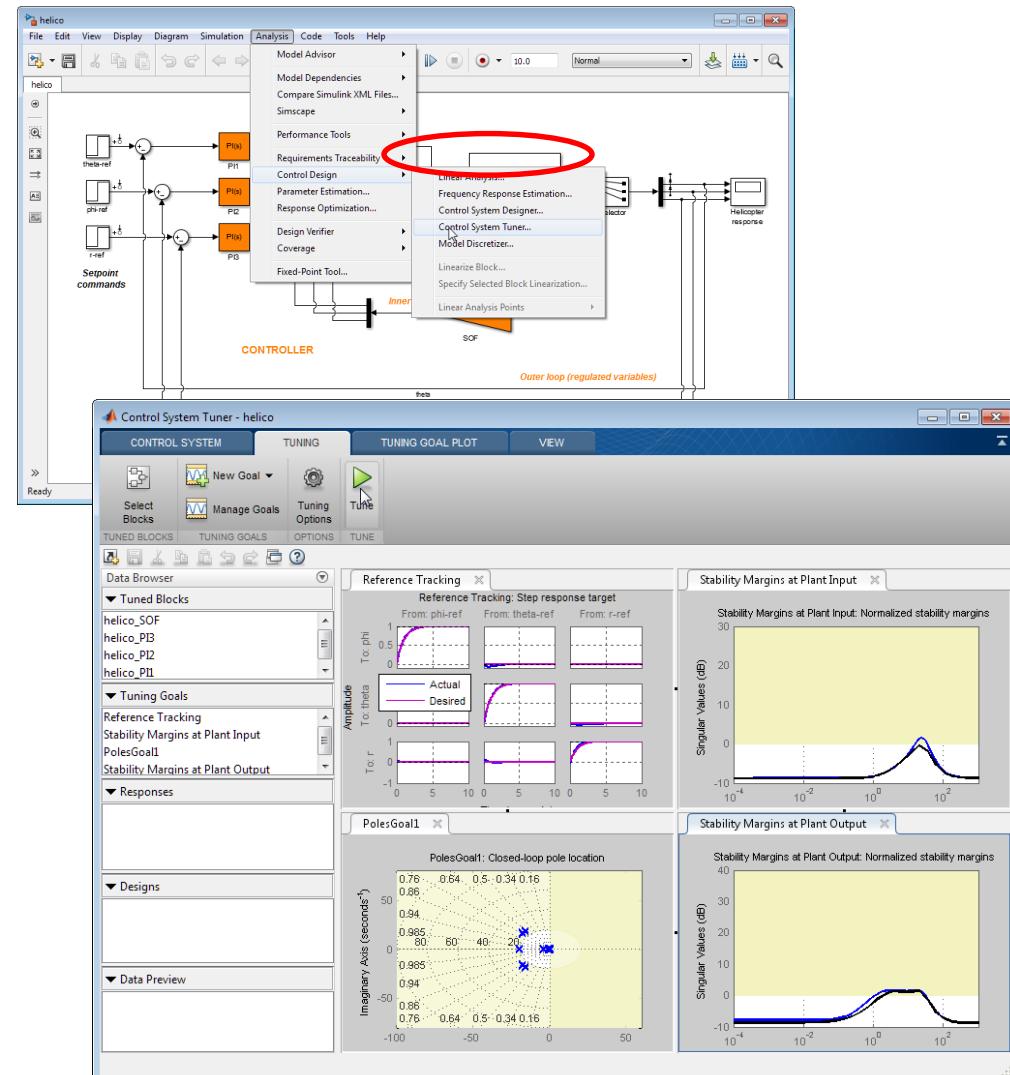
The screenshot shows the Control System Tuner app interface for a model named 'helico'. The main window title is 'Control System Tuner - helico'. The top menu bar includes 'CONTROL SYSTEM', 'TUNING' (selected), 'TUNING GOAL PLOT', and 'VIEW'. Below the menu are buttons for 'Select Blocks', 'New Goal', 'Manage Goals', 'Tuning Options', and a 'TUNE' button. The left sidebar contains sections for 'Tuned Blocks' (listing 'helico\_SOF', 'helico\_PI3', 'helico\_PI2', 'helico\_PI1'), 'Tuning Goals' (listing 'Reference Tracking', 'Stability Margins at Plant Input', 'PolesGoal1', 'Stability Margins at Plant Output'), 'Responses', 'Designs', and 'Data Preview'. The central area displays four plots:

- Reference Tracking:** Shows step response targets for 'phi-ref', 'theta-ref', and 'r-ref' with 'Actual' (blue) and 'Desired' (magenta) responses.
- Stability Margins at Plant Input:** A Nichols chart showing Singular Values (dB) versus Frequency (log scale from  $10^{-4}$  to  $10^2$ ). The plot is mostly yellow, indicating good stability margins.
- PolesGoal1:** A Nichols chart showing Imaginary Axis (seconds $^{-1}$ ) versus Real Axis (from -100 to 50). It displays closed-loop pole locations as blue crosses within a grid.
- Stability Margins at Plant Output:** A Nichols chart showing Singular Values (dB) versus Frequency (log scale from  $10^{-4}$  to  $10^2$ ). Similar to the input plot, it shows a yellow region.

# Control System Tuner App - Robust Control Toolbox

Tune fixed-structure controllers in Simulink

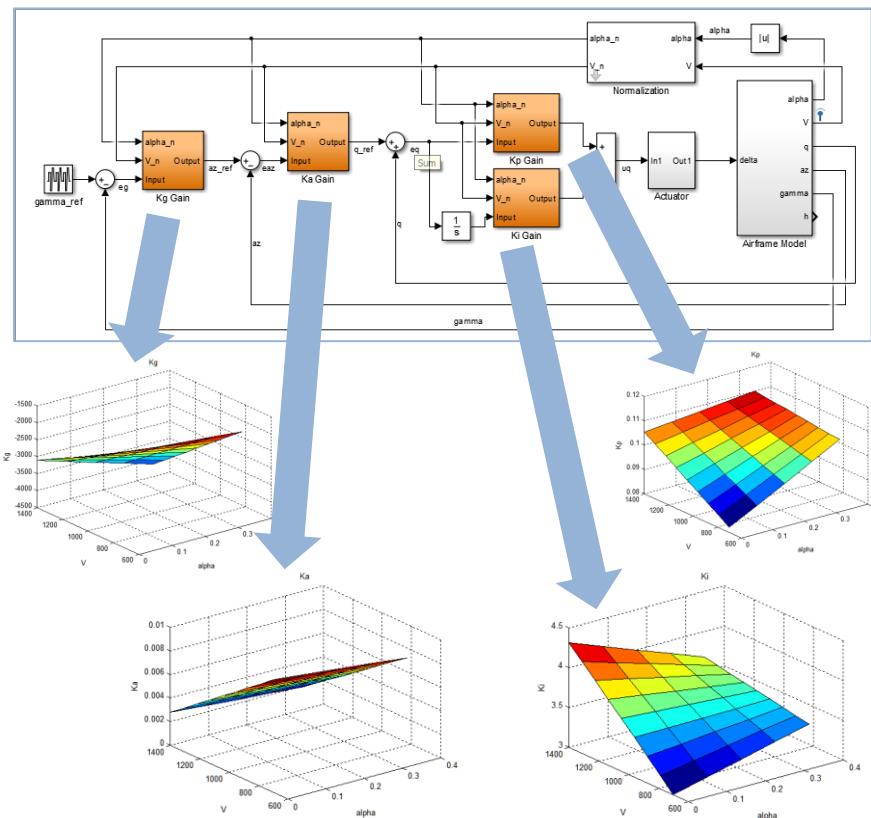
- Specify blocks to tune
- Add tuning goals
- Visualize tuning results
- Update tuned Simulink blocks from app



# Tuning of Gain-Scheduled Controllers with *systune* and *looptune*

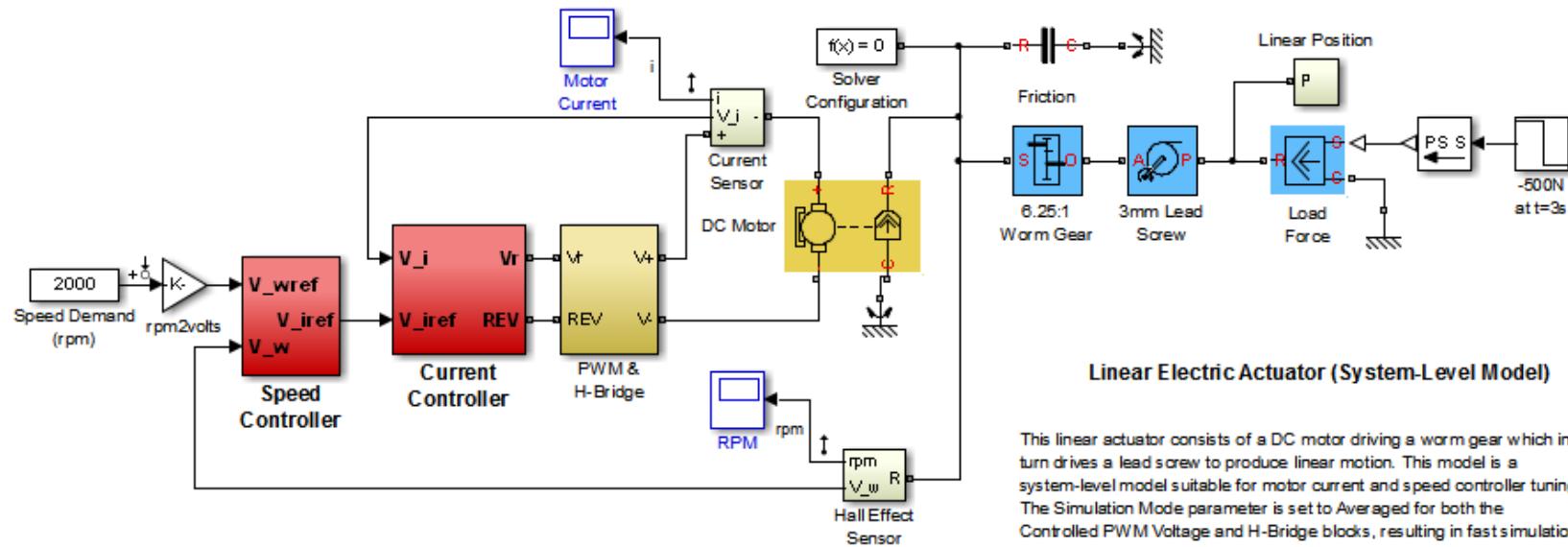
**Automatic tuning of controller gains at all operating conditions to meet design requirements and create smooth gain surfaces**

- New *gainsurf* command for parameterizing controller gains as functions of scheduling variables
- Software automatically tunes coefficients of that parameterization



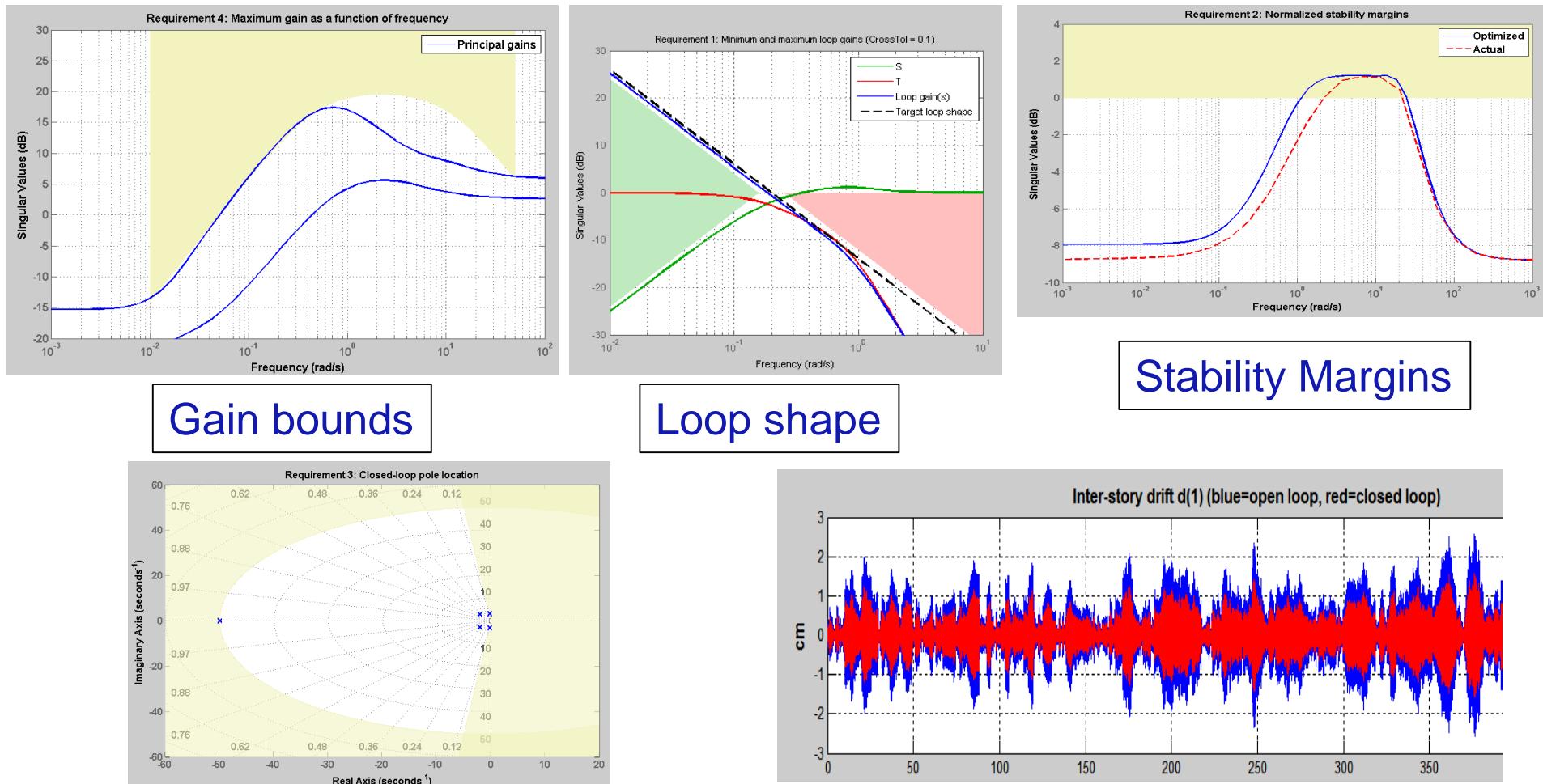
# SYSTUNE Inside

- Gain surface tuning leverages *SYSTUNE* technology
- No restriction on control structure, number of feedback loops, or compensator types



# SYSTUNE Inside

- Variety of tuning goals available to express control objectives



# Agenda

## Tuning Controller Parameters

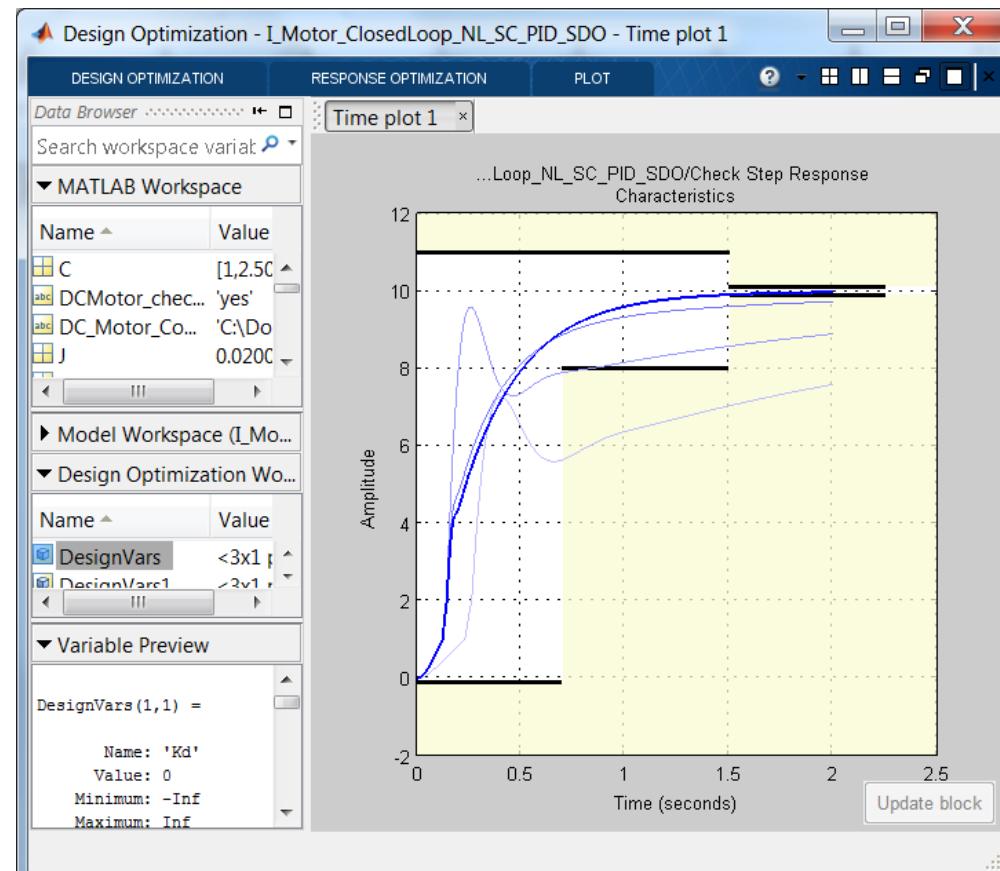
- Classical control tuning techniques
- **Optimization-based system response tuning**

## Real-Time Testing and Simulation

- Prepare models for real-time execution
- Connect models with hardware under test using flexible real-time testing hardware
- Tune parameters and log\monitor signals during real-time execution using Simulink Real-Time™ Explorer
- Automate real-time test routines using MATLAB® scripts

# Controller Tuning Using Optimization Based Techniques

- Tune model parameters using numerical optimization to meet desired system performance
- Handle plants with discontinuous events and nonlinearities
- Used by system and control engineers to optimize physical system, controller, and overall design



# Family of Control Design Products

- Control System Toolbox
- Simulink Control Design
- System Identification Toolbox
- Robust Control Toolbox
- Simulink Design Optimization
- Model Predictive Control Toolbox
- Fuzzy Logic Toolbox
- Neural Network Toolbox

# Control Design Is Done.

## What's Next?

### Real-Time Simulation and Testing?

# Agenda

## Tuning Controller Parameters

- Classical control tuning techniques
- Optimization-based system response tuning

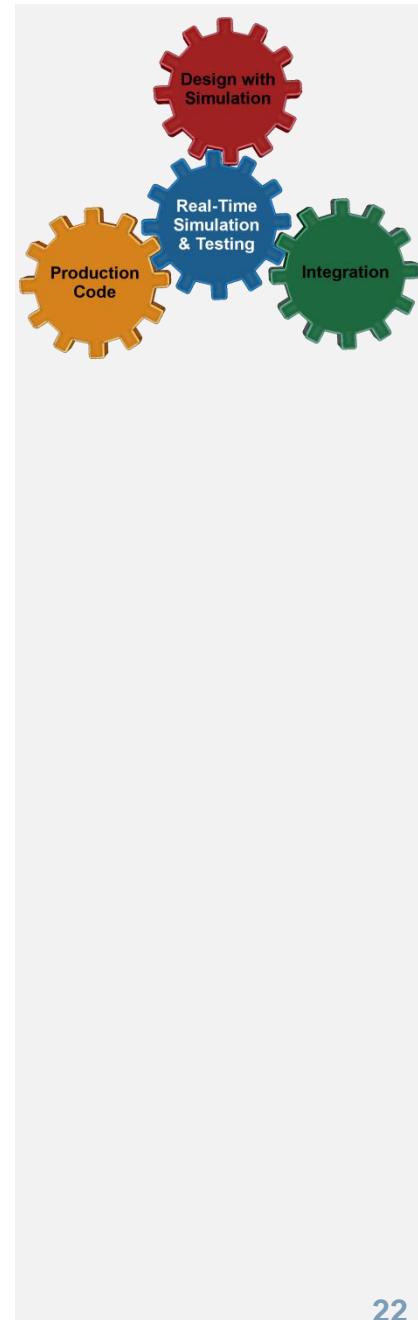
## Real-Time Testing and Simulation

- Prepare models for real-time execution
- Connect models with hardware under test using flexible real-time testing hardware
- Tune parameters and log\monitor signals during real-time execution using Simulink Real-Time™ Explorer
- Automate real-time test routines using MATLAB® scripts

# Why do Model-Based Design Real-Time Simulation & Testing?

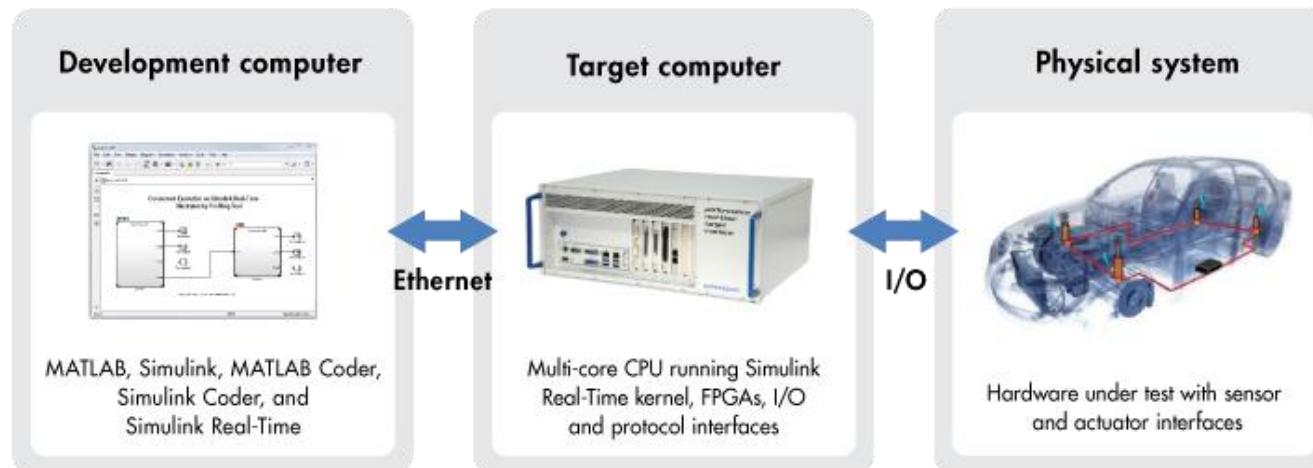
**It enables you to:**

- Test, verify, and validate your algorithmic and system designs earlier
- Achieve determinism through system modeling, automatic code generation and real-time software/hardware execution
- Evaluate new ideas using a flexible, scalable, production independent development platform
- Minimize risk, reduce costs, shorten time-to-market



# Simulink Real-Time™ Enables Real-Time Simulation & 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.



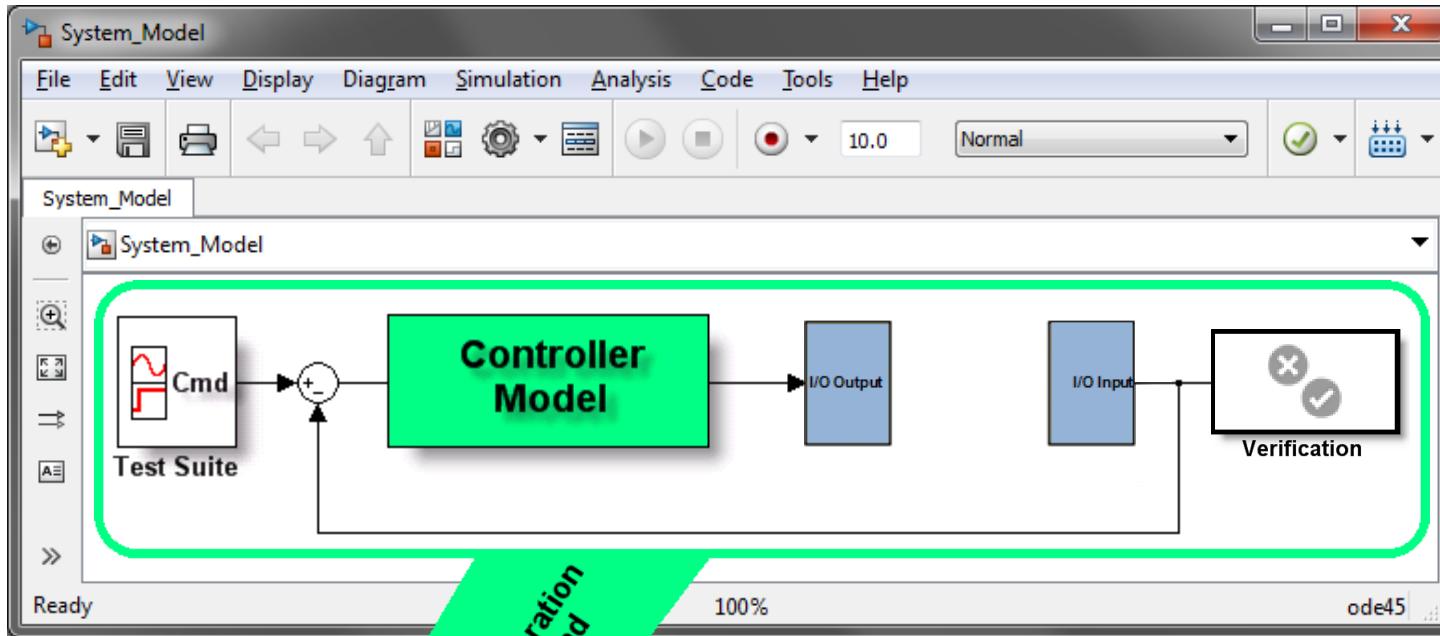
*"Using Model-Based Design with MATLAB and Simulink, we achieved multiple goals simultaneously. We developed a sophisticated controller for digital hydraulics that is more reliable, accurate, and efficient than previous systems, and we accelerated development, which gives us a competitive advantage."*

*Simulations in Simulink and real-time testing with Simulink Real-time helped us deliver an exceptionally reliable control system."*

*Kari Leminen, Metso*

# Real-Time Simulation & Testing Tasks:

## Rapid Controls Prototyping



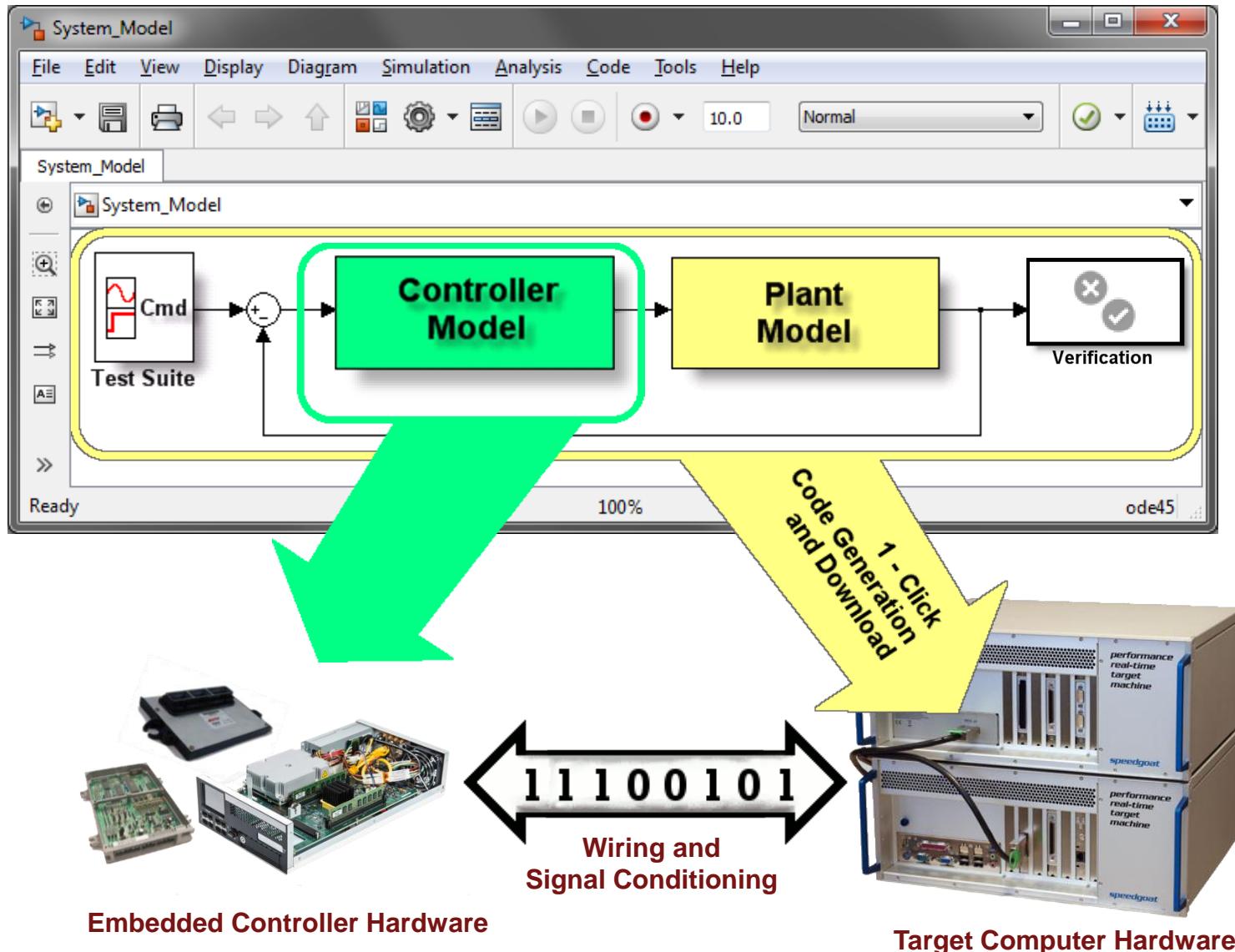
Target Computer Hardware



Physical Plant Hardware

# Real-Time Simulation & Testing Tasks:

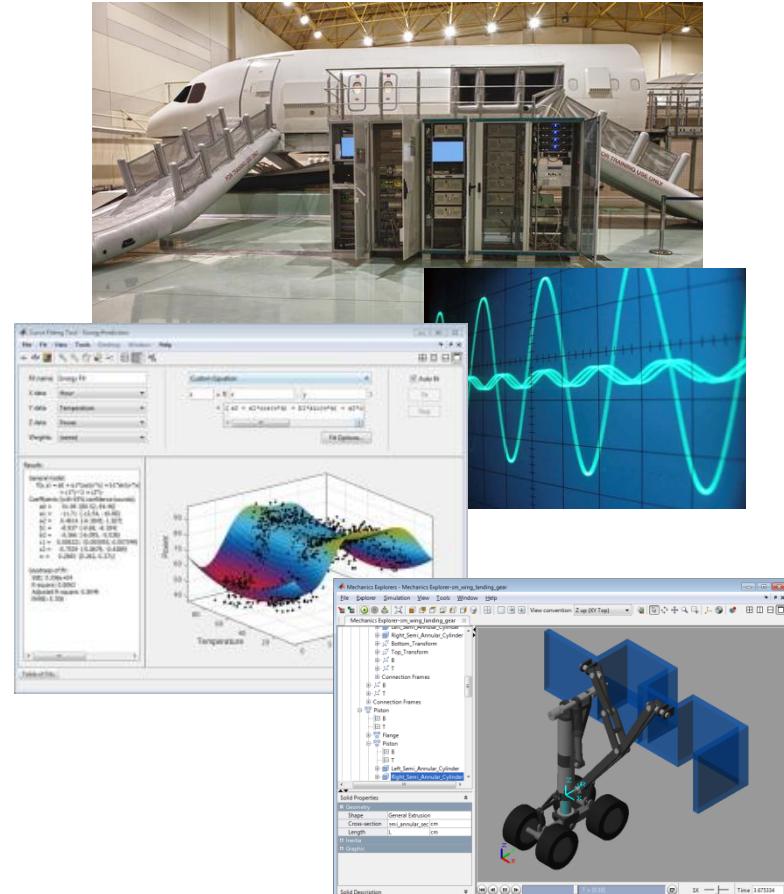
## Hardware-in-the-loop (HIL) Simulation



# Additional Real-Time Simulation & Testing Tasks:

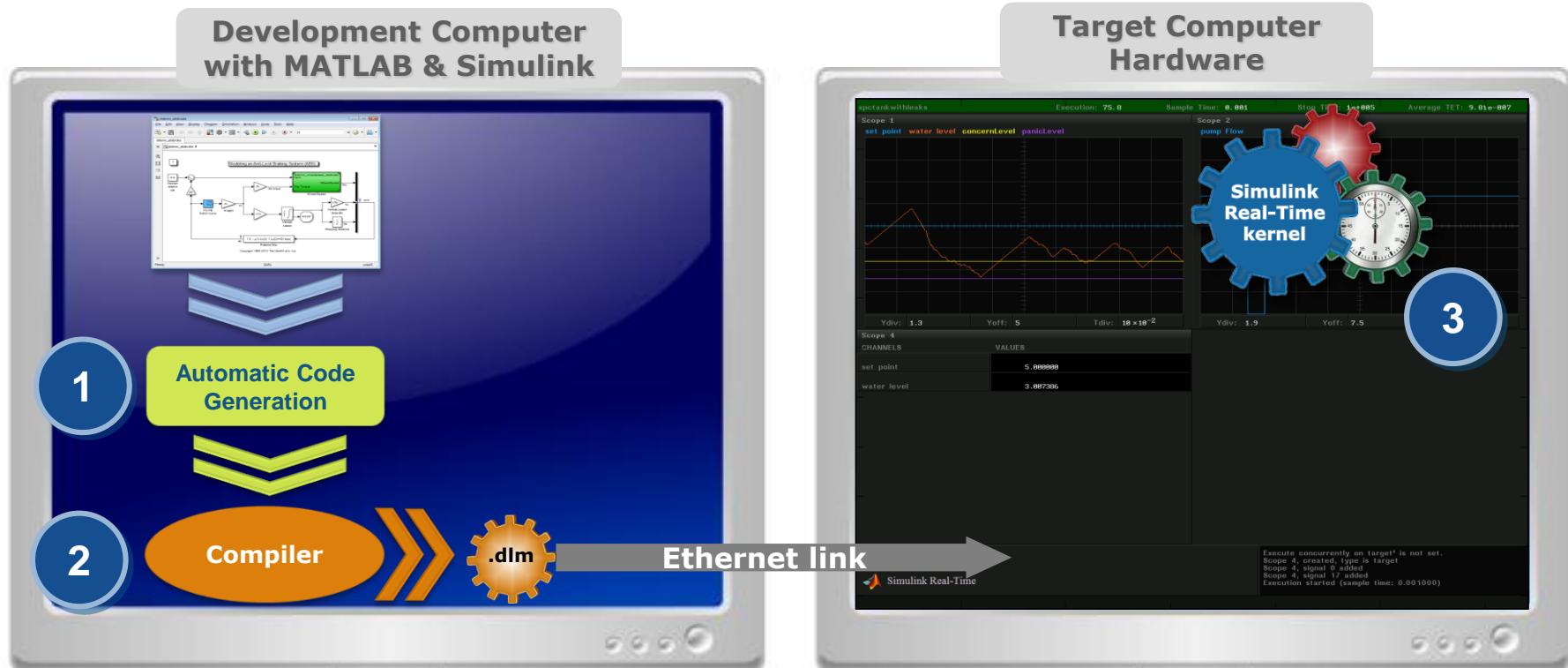
## *Parametric Evaluation and Performance Assessment*

- System robustness
  - Monte Carlo analysis
  - Operational envelope testing
- Human factors
  - Human-in-the-loop simulation
  - Virtual reality simulators
- Calibration
  - Tune algorithmic coefficients
  - Optimize performance



# What is Simulink Real-Time?

*From desktop simulation to real-time*

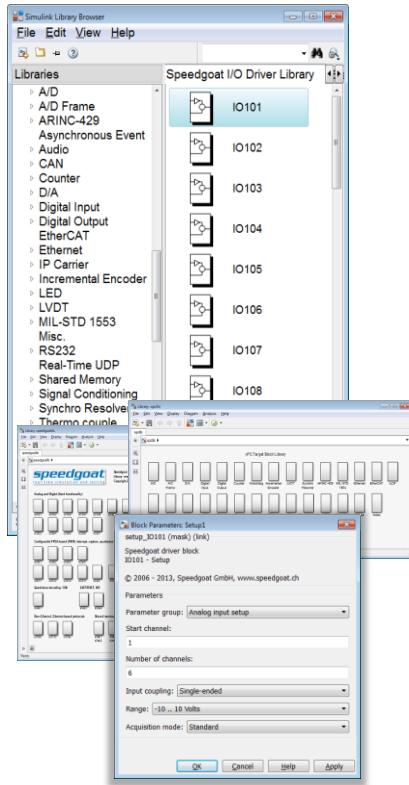


Creation of real-time applications from Simulink models and loading them onto dedicated target computer hardware in 3 automated steps:

- 1 Code Generation
- 2 Compile & Link
- 3 Download & Ready to Run

# What is Simulink Real-Time?

*Connect to your physical system*



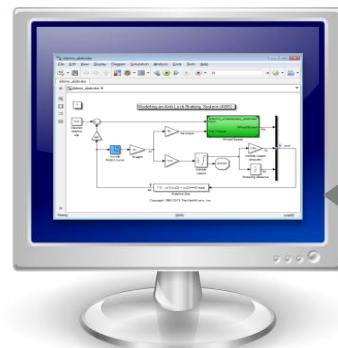
- Support for a broad range of I/O types and communication protocols
- Easy drag & drop and configuration within a Simulink model

# What is Simulink Real-Time?

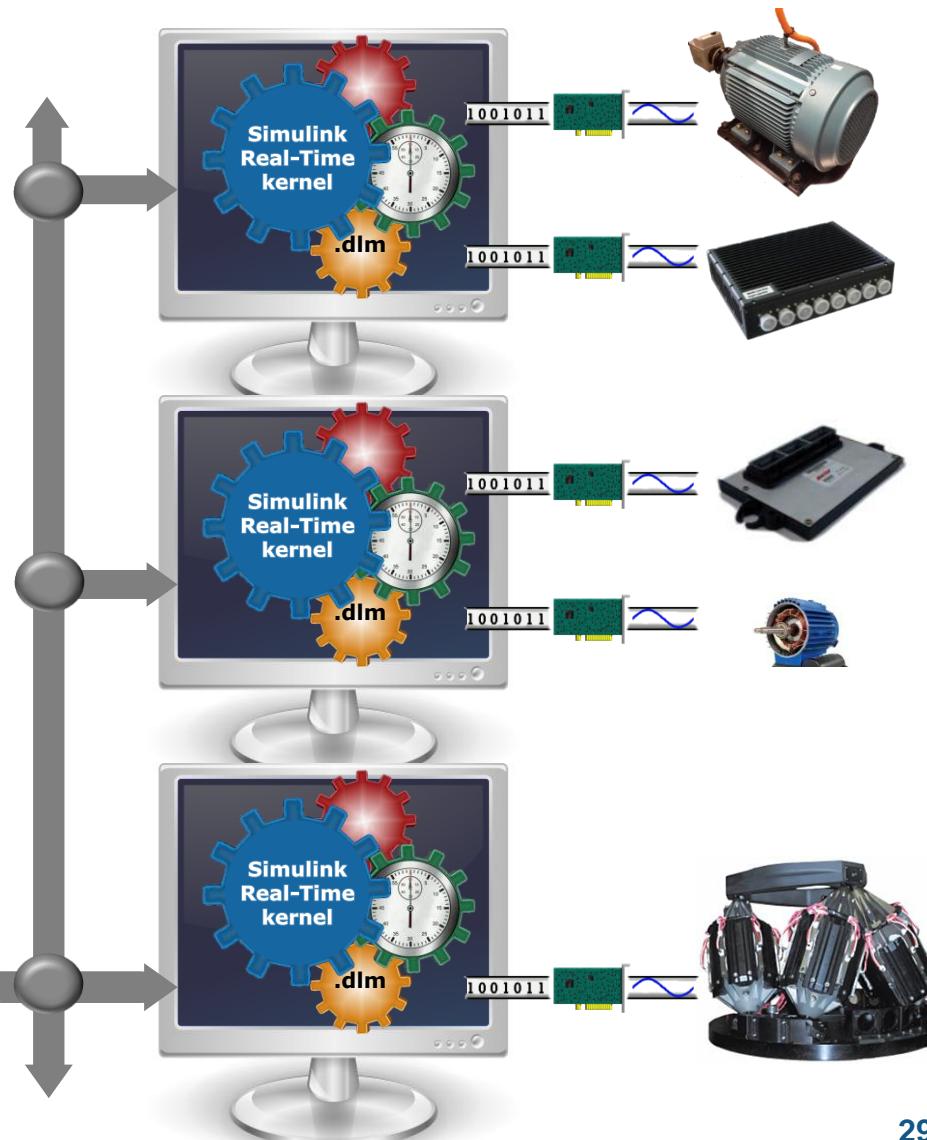
*Multiple systems and multiple uses*

## One license supports

- Creation and simultaneous control of many systems
- Many real-time uses
  - Real-time simulation and testing
  - Data acquisition and instrumentation
  - Lab controllers
- Stand alone operation

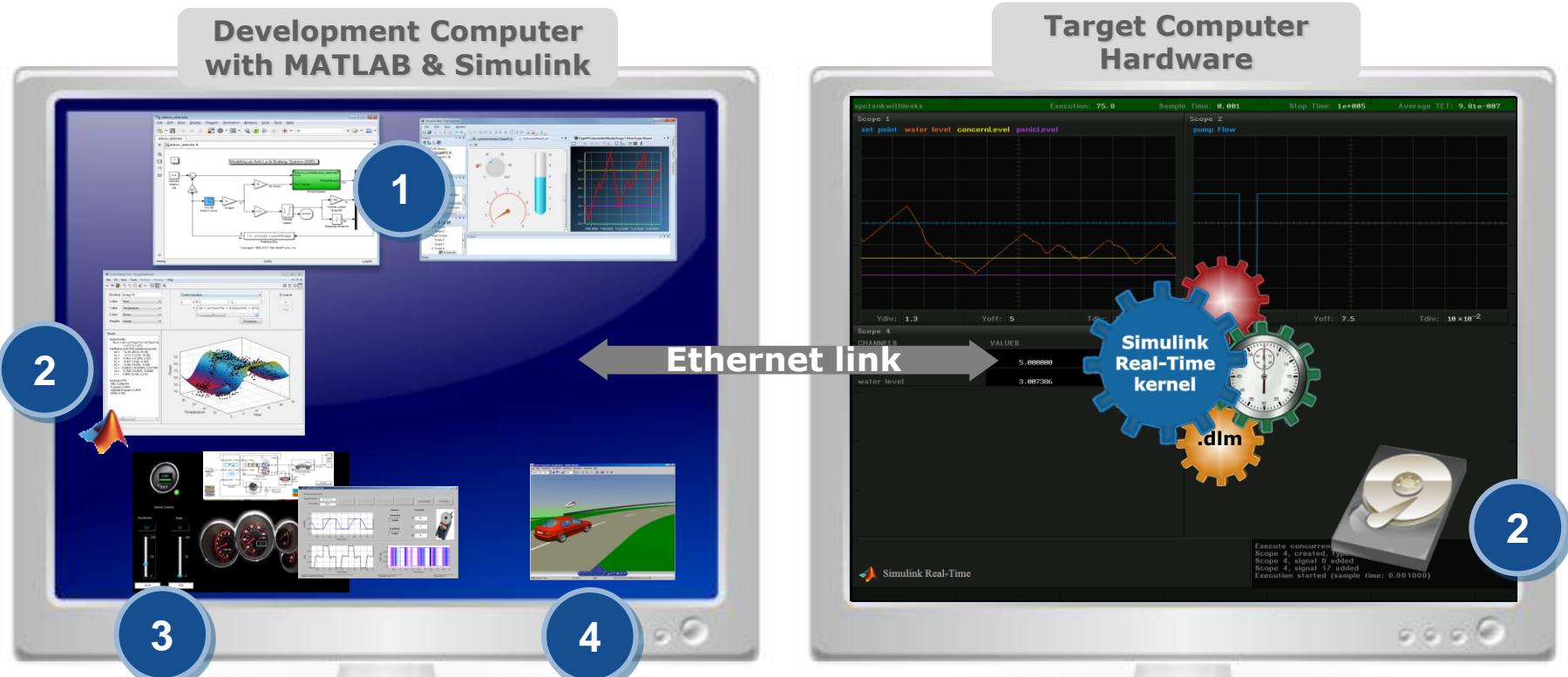


Ethernet link



# What is Simulink Real-Time?

*Extendable, integrated, and interactive*



**1** Live parameter tuning,  
signal monitoring, and  
execution control

**2** Data logging for offline  
analysis in MATLAB

**3** UI/HMI connectivity

**4** Extensibility with  
other software tools  
(e.g. virtual reality)

# Simulink Real-Time™

***Build, run and test real-time applications***

## What it is...

Simulink Real-Time lets you create real-time applications from Simulink models and run them on dedicated target computer hardware connected to your physical system. It supports real-time simulation and testing, including rapid control prototyping, DSP and vision system prototyping, and hardware-in-the-loop (HIL) simulation.

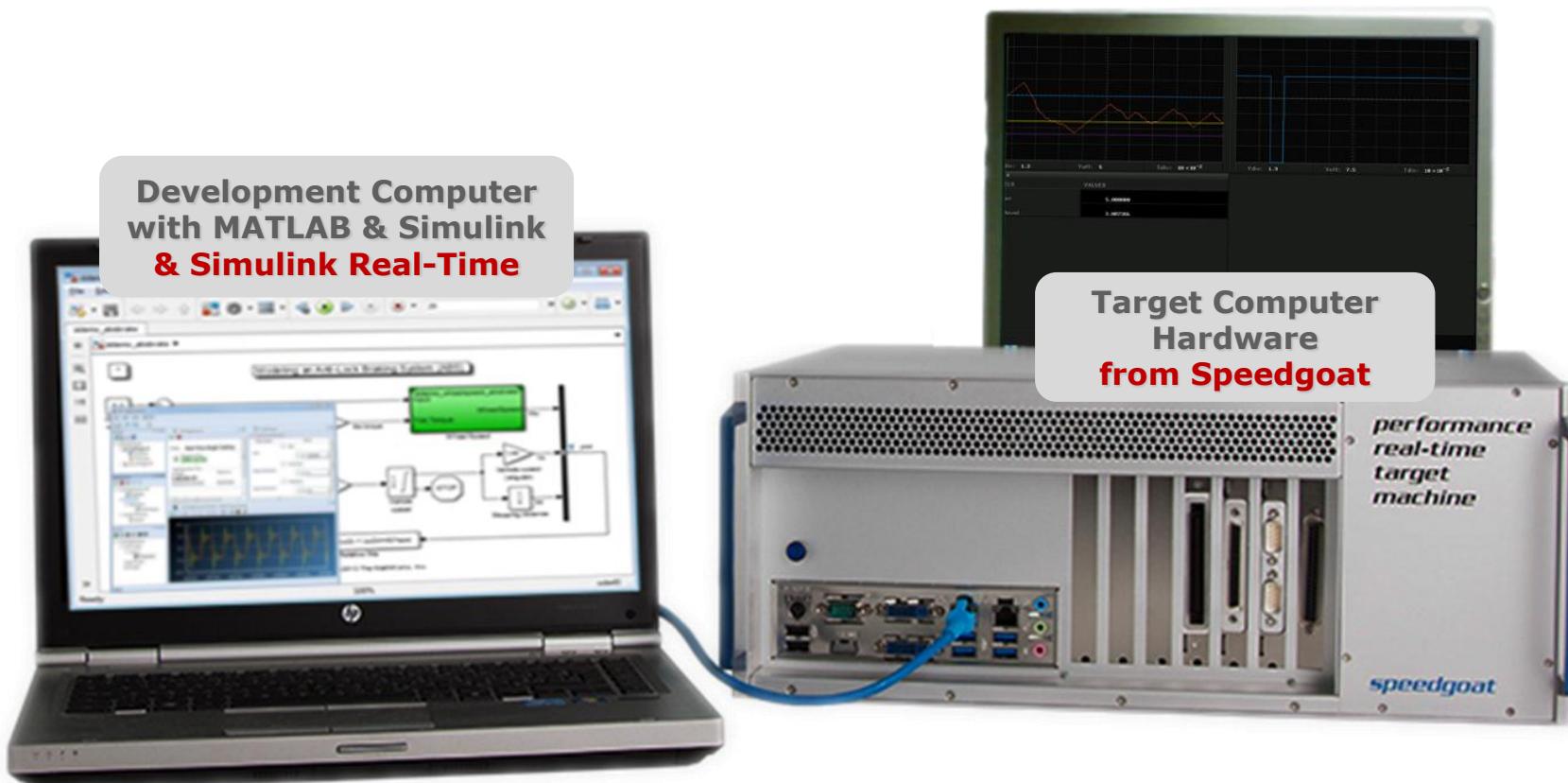
## What is included...

- Tools, UIs, functions, and blocks to control, monitor, and tune real-time applications
- Connectivity interfaces to external physical systems and external software
- A multicore/multitasking real-time kernel with microsecond granularity and concurrent execution support
- Ability to perform co-execution of applications running on a real-time target computer with FPGAs
- Integration with the whole MathWorks' environment



# What Hardware is used with Simulink Real-Time?

*Real-time software environment + real-time target computer*



# Speedgoat Provides Real-Time Target Computers

*Made for use with Simulink Real-Time*

**Speedgoat develops and sells Real-Time Machines consisting of**

- An industrial PC (Real-time target machine)
- I/O modules
- Software drivers, cables and tools to connect with a prototype

**Simulink Real-Time and Speedgoat target computer hardware are expressly designed to work together**



Real-time target machine



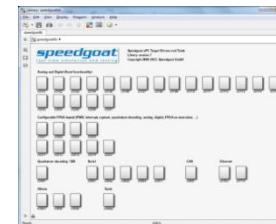
I/O modules installed in target machine



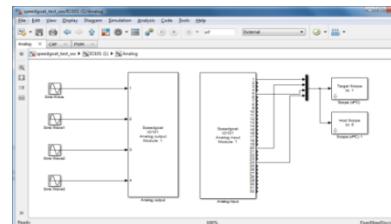
I/O Cable



Terminal board



Simulink drivers



Simulink test models



# About Speedgoat

- Highly specialized developer of turnkey real-time target machines
- Incorporated in 2007 by former MathWorks employees
- Located in Bern, the Swiss capital
- Over 1,000 Real-time target machines sold to date, all for use with Simulink
- ~75% of all new Simulink Real-Time seats are accompanied by a Speedgoat system



# 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

# Fixed-Function I/O Modules

**Powerful “as is” functionality**



IO Type	Functionality
Analog	High-resolution, high-speed, simultaneous sampling, BNC and XLR panels, ...
Digital	TTL/LVCMOS, RS422/RS485/LVDS, 06-48V, low/high side, opto-coupled, ...
Serial	RS232, RS422, RS485, SDLC, HDLC
Ethernet-based	EtherCAT, EtherNet/IP, Modbus TCP, POWERLINK, real-time UDP, ...
Protocols	CAN, SAE J1939, LIN, Profibus, Modbus, SPI, I2C, SSI, ARINC-429, MIL-STD-1553, FlexRay, ...
Video	CameraLink, USB WebCam
Audio/Speech	Audio/Speech optimized analog IO modules
Shared Memory	Reflective Memory for high speed data transfer in multi-processor systems
Various	LVDT/RVDT, Synchro/Resolver, reed relays, programmable resistors, external signal conditioning modules (current to voltage, voltage to current, temperature, ...)

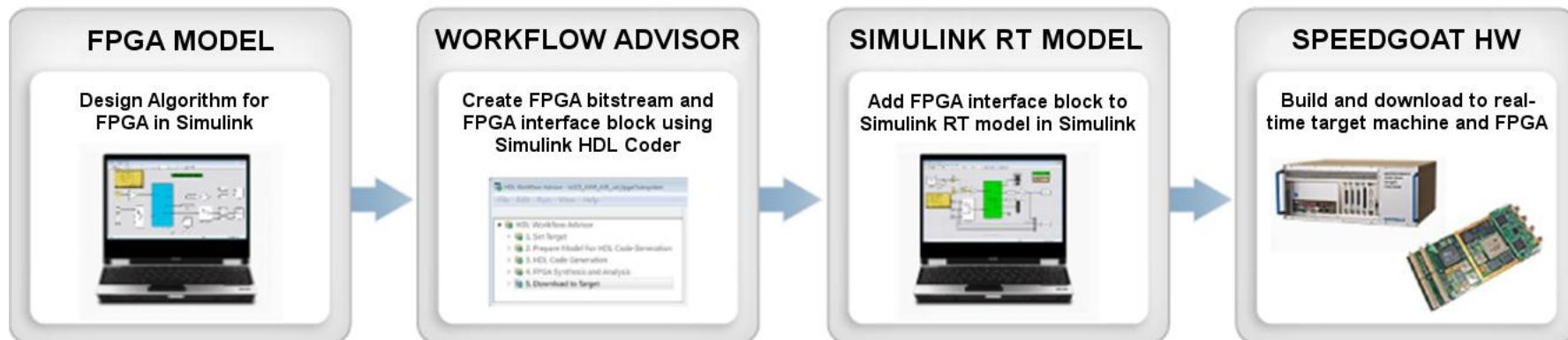
- Delivery includes I/O cables, terminal boards, test models, and Simulink driver blocks
- 3 years of warranty, and long-term availability (7+ years for most I/O modules)

# Multi-Function I/O Modules

***Reconfigurable to support your application***



- Execute high-speed algorithms on an FPGA connected to a model running in real time with Simulink Real-Time.
- Automatically program the FPGA without needing to know HDL code
- Quick reconfiguration of FPGA I/O promotes a flexible real-time testing environment.
- Three different use cases supported
  - Pre-configured FPGA Code Module functionality
  - Execute Simulink Applications on FPGA using automatic HDL Code Generation
  - Write and implement your own HDL Code using Speedgoat FPGA Engineering Kits

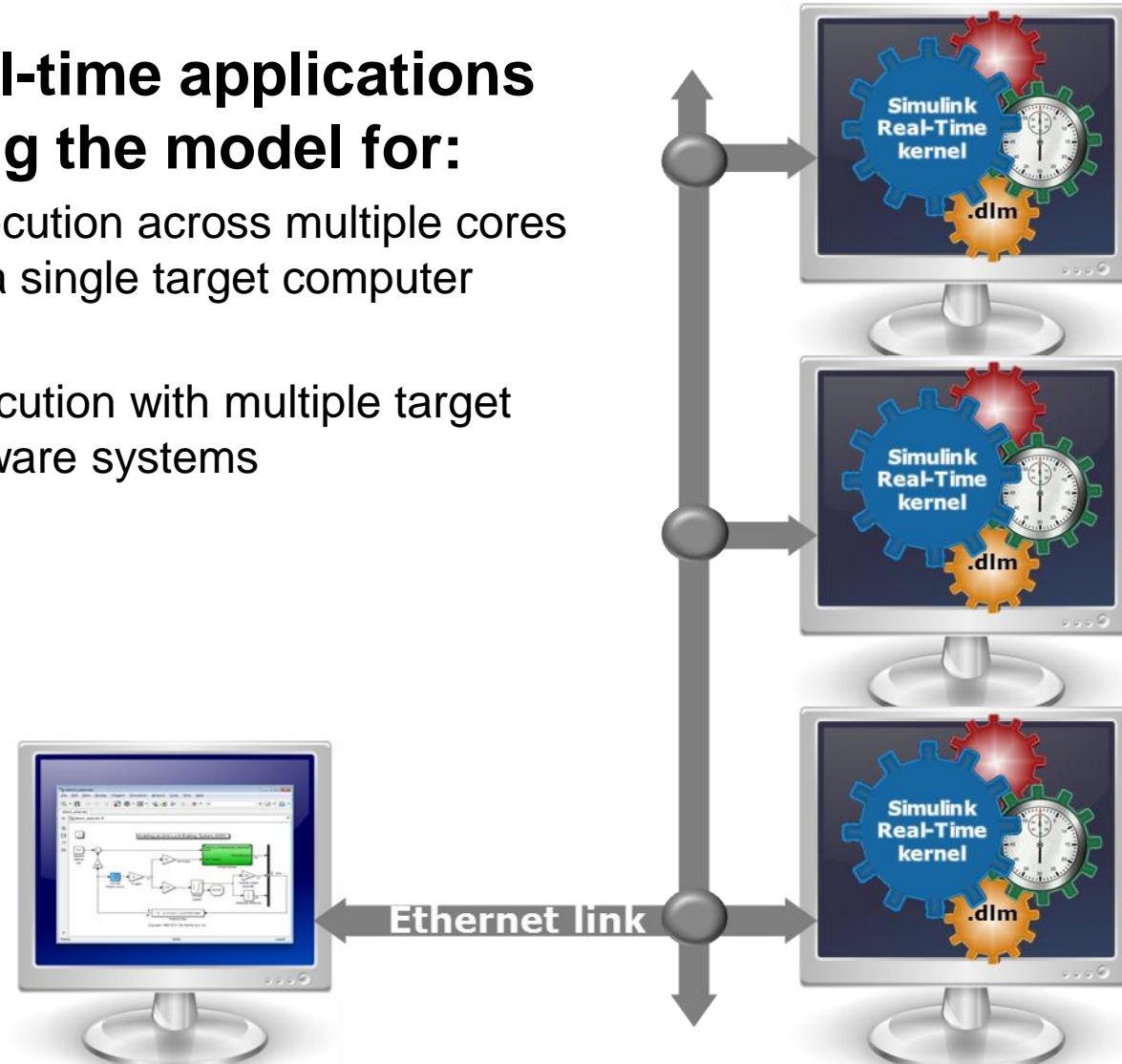


# Take Advantage of the Hardware

*To increase speed and handle application complexity*

## Speed up real-time applications by partitioning the model for:

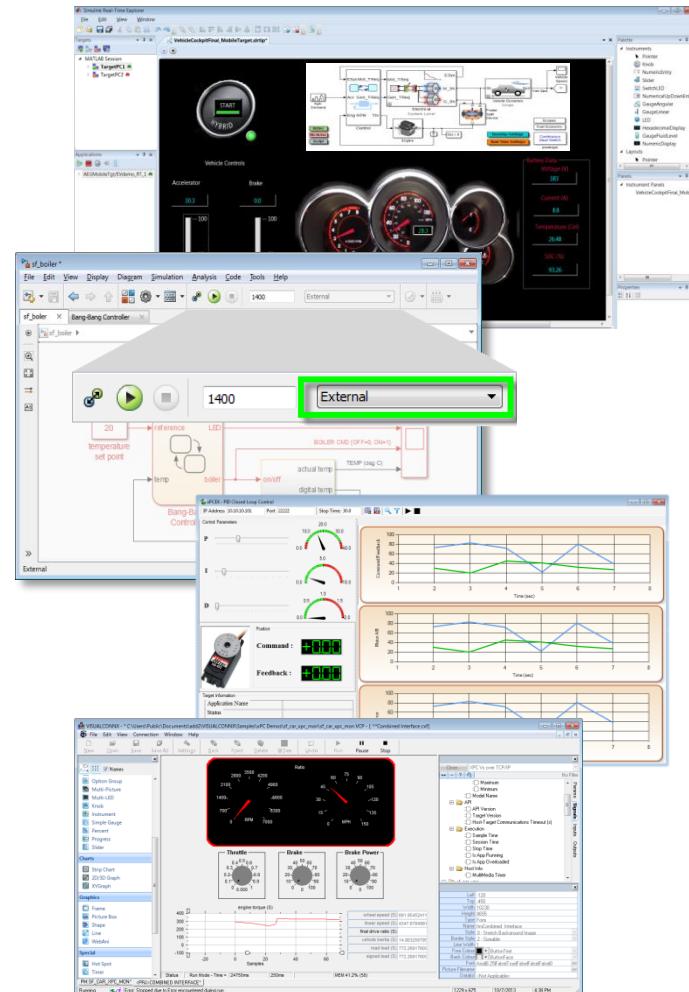
- Concurrent Execution across multiple cores and FPGAs in a single target computer
- Distributed Execution with multiple target computer hardware systems



# Interacting with the Simulink Real-Time Application

## *Monitoring, tuning, and control of real-time applications*

- Support multiple ways of working and interacting
  - Simulink Real-Time Explorer
  - Simulink External Mode
  - MATLAB command scripts
  - MATLAB UIs
  - Simulink Real-Time external APIs
  - 3<sup>rd</sup> party visualization tools
- Manage and control multiple target computers simultaneously

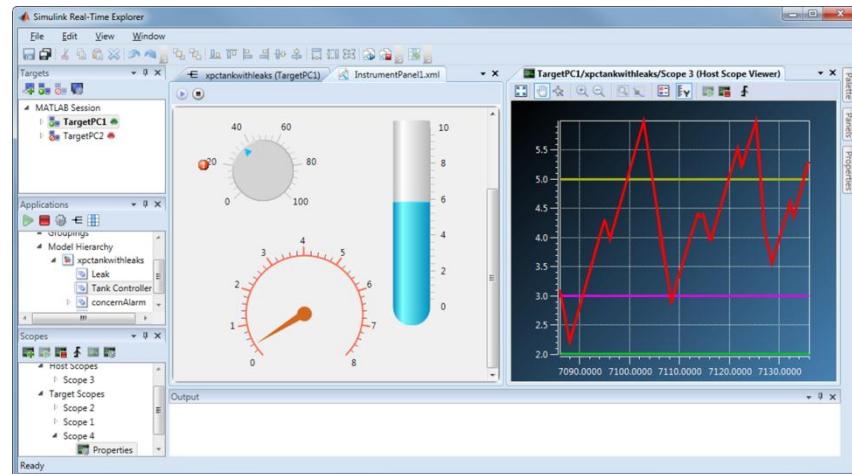


# Control and Explore Your Real-Time Application

*Built-in control and monitoring User Interface*

## Simulink Real-Time Explorer

- Control target computer specific properties
- Easy access to the model hierarchy
- Commit parameter updates individually or as groups
- Add scopes and data logging on the fly
- Graphical controls and displays to design and run instrument panels

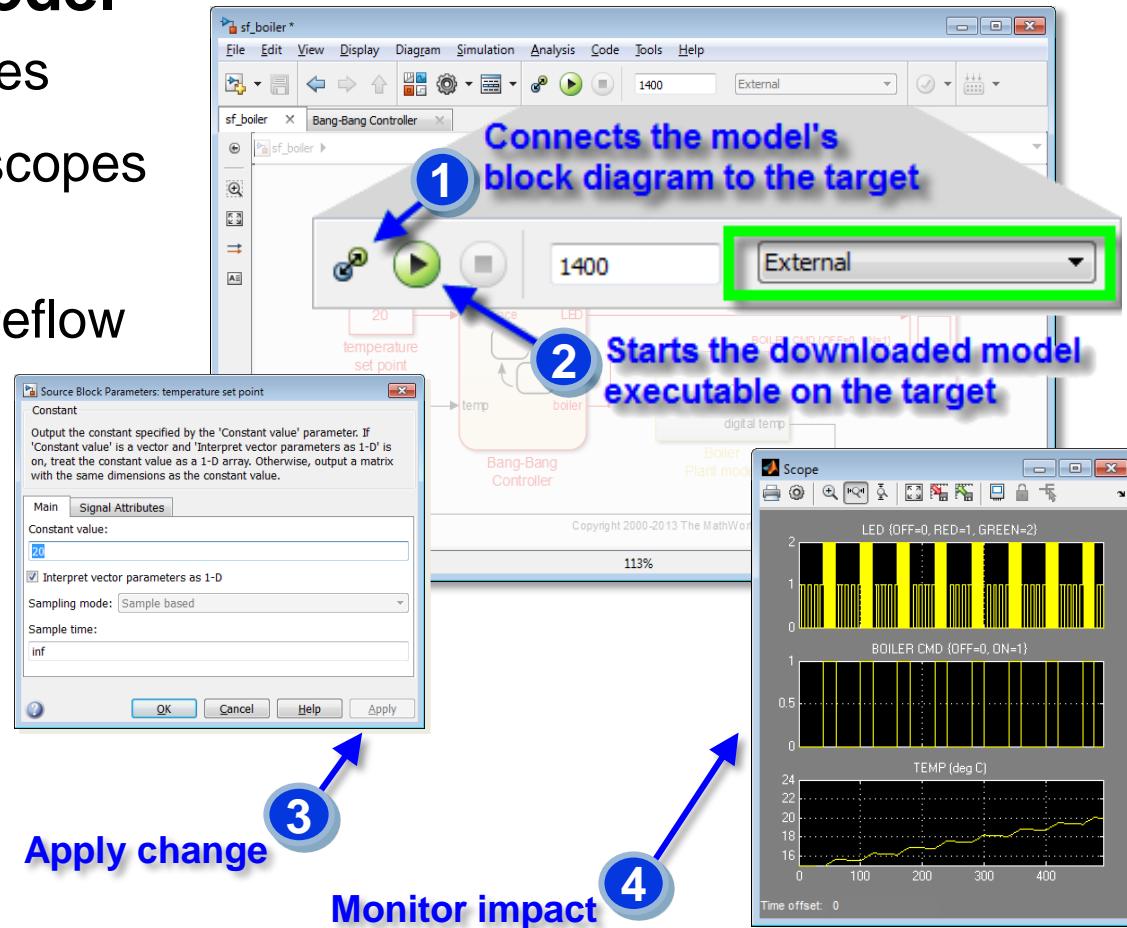


# Monitor and Tune Real-Time Application

*Directly from Simulink using Simulink Coder's External Mode*

## Direct interactive access from the Simulink model

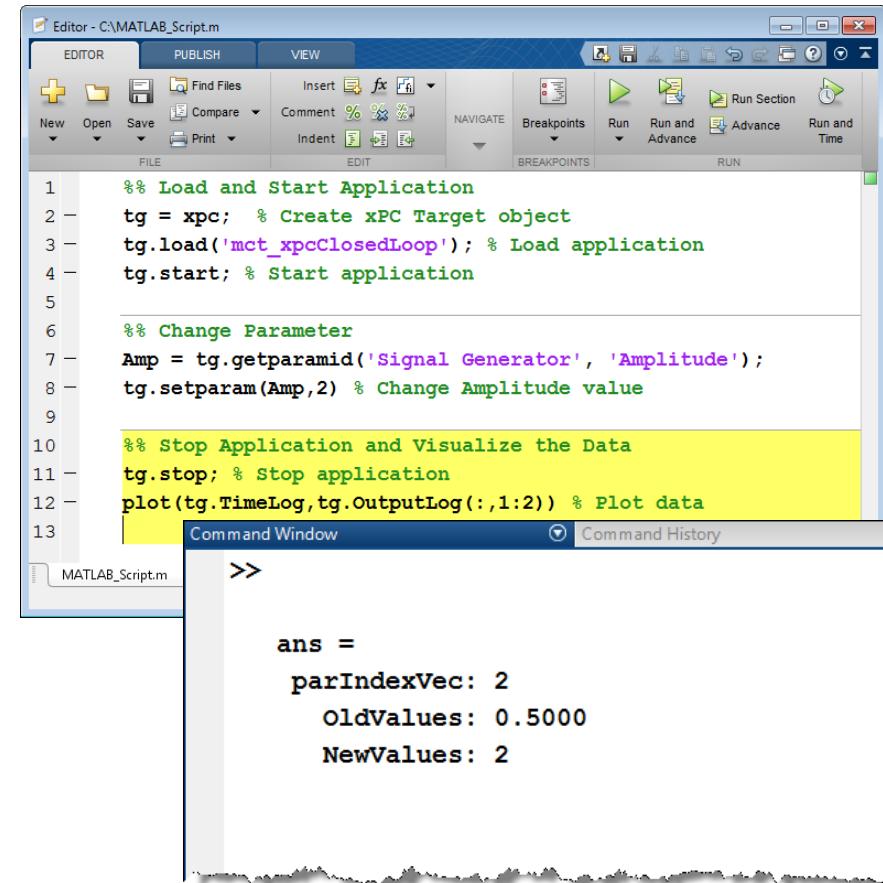
- Apply parameter changes
- Monitor the impacts in scopes and displays
- Support for 3D and Stateflow animations
- Log test data



# Access Real-Time Application from MATLAB

*MATLAB scripts incorporate and automate testing with analysis*

- Leverage entire MATLAB language to access and control all aspect of your real-time application
- Use interactively from the command line as well as from scripts
- Augment testing with MATLAB toolboxes and functions such as
  - Optimization Toolbox for tuning parameters
  - Signal Processing Toolbox for post-processing data
  - Report Generator for automation of tests and publishing results



The screenshot shows the MATLAB interface with the following details:

- Editor - C:\MATLAB\_Script.m**: The script file contains MATLAB code for interacting with an xPC Target object. Lines 10 through 13 are highlighted in yellow.

```

1 %>> %% Load and Start Application
2 tg = xpc; % Create xPC Target object
3 tg.load('mct_xpcClosedLoop'); % Load application
4 tg.start; % Start application
5
6 %% Change Parameter
7 Amp = tg.getparamid('Signal Generator', 'Amplitude');
8 tg.setparam(Amp, 2) % Change Amplitude value
9
10 %% Stop Application and Visualize the Data
11 tg.stop; % Stop application
12 plot(tg.TimeLog,tg.OutputLog(:,1:2)) % Plot data
13

```

- Command Window**: The output of the script execution is shown in the Command Window.

```

>>

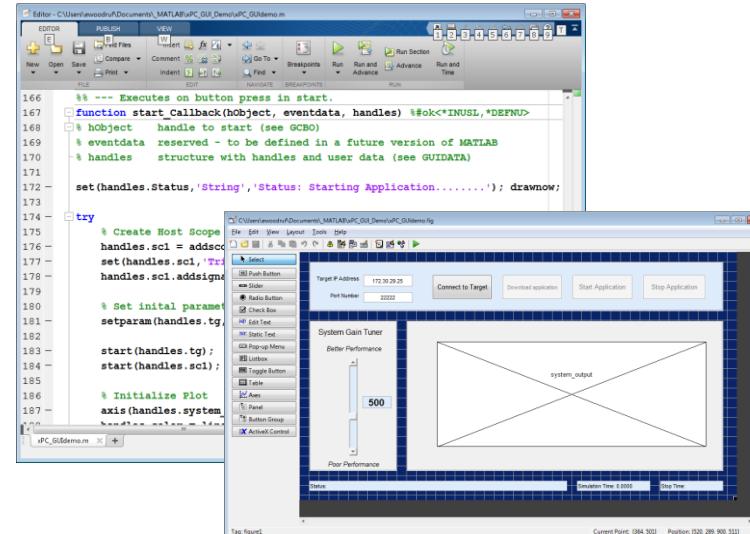
ans =
    parIndexVec: 2
        OldValues: 0.5000
        NewValues: 2

```

# Create UIs in MATLAB

# **Graphical front end for your MATLAB scripts and programs**

- **Create a MATLAB UI Interactively**
    - Easily design your UI graphically in the GUIDE Layout Editor
    - Automatically generates MATLAB code to add your scripts to
  - **Create a MATLAB UI Programmatically**
    - Allows more control and customization
    - Built-in functions and graphical controls
    - Add your own graphical controls with Java and ActiveX
  - **Deploy as a MATLAB App**

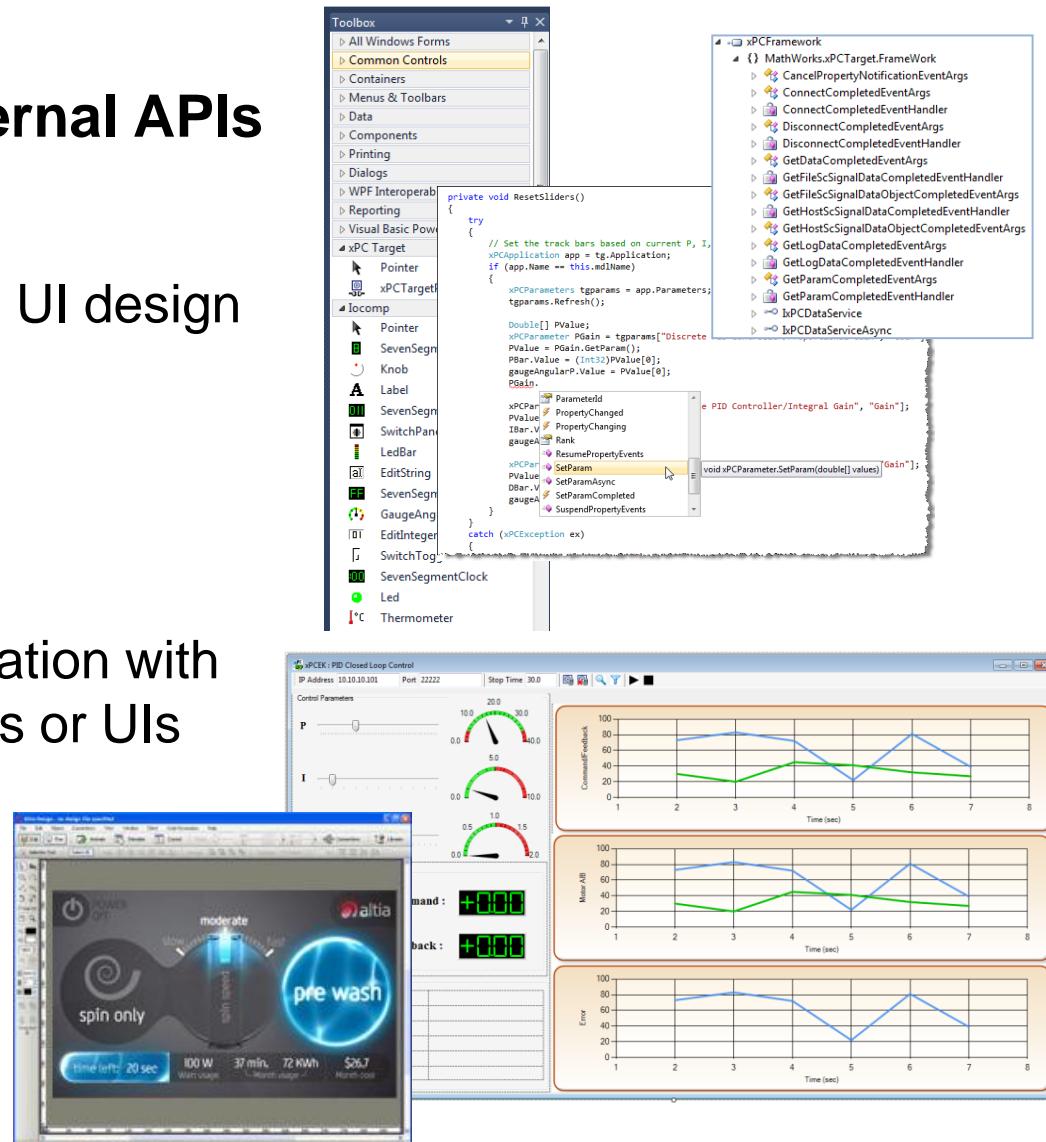


# Create UIs Independent of MATLAB

*Use built-in, flexible APIs*

## Simulink Real-Time External APIs

- .NET, C, COM
- Can be used with leading UI design environments such as
  - Microsoft Visual Studio
  - Qt Creator
  - Altia
- Connect the target application with other external applications or UIs



# Works With 3<sup>rd</sup> Party Drag & Drop HMI Tools

*HMI tool to interface with real-time applications*

## Example: VISUALCONNX

- Intuitive "drag and drop" multi-windowed UI
- Data aware controls
- Supports scripting for advanced functionality
- Supports both model testing and real-time testing
- MathWorks Connection Partner



# Benefits of a Simulink Real-Time Solution

## Fully assembled solution

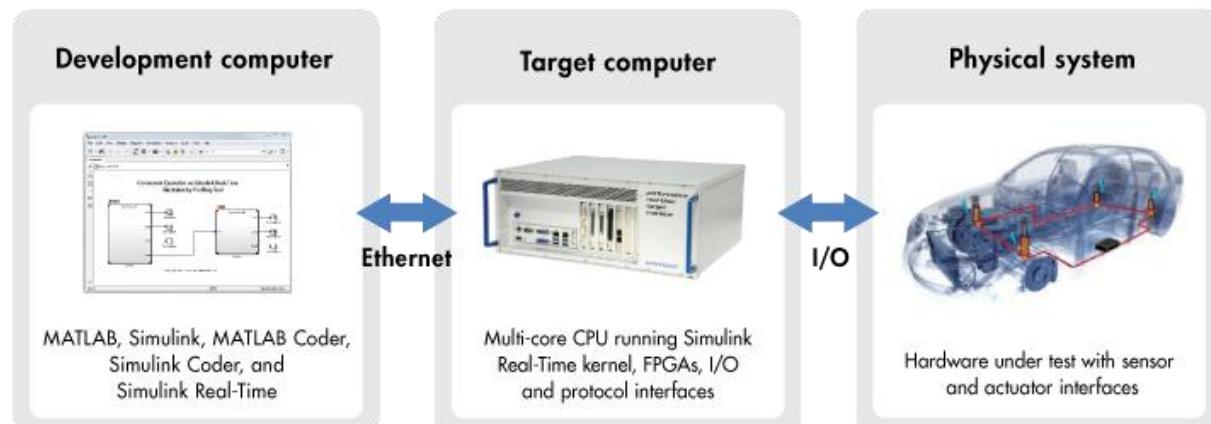
- ✓ Focus on developing your next generation software and hardware designs instead of developing the tools & hardware infrastructure

## Shorten time-to-market

- ✓ Benefit from a flexible and production independent platform which can be easily adapted to changing requirements
- ✓ Prove and improve your Simulink design with your hardware at the earliest possible stage and continuously try new ideas

## Reduce costs

- ✓ Avoid otherwise costly design flaws by detecting errors at a stage where they are still cost effective to correct
- ✓ Simulate and automate test scenarios and hardware interactions which are otherwise complex, expensive, or dangerous to perform



# Recorded Webinars & Examples

Watch a recorded webinar:

## **A Simulink Real-Time Testing Solution for Power Electronics & Motor Control**

<http://www.mathworks.com/company/events/webinars/wbnr68656.html>

## **Prove Your Simulink Designs with Real-Time Hardware Testing**

<https://www.mathworks.com/company/events/webinars/wbnr73147.html>

Explore example models:

## **Field-Oriented Control of a Permanent Magnet Synchronous Machine**

This example shows the basic workflow and key APIs for generating C code from a motor control algorithm, and for verifying its compiled behavior and execution time.

[http://mathworks.com/products/demos/shipping/rtw/rtwdemo\\_pmsmfoc\\_script.html](http://mathworks.com/products/demos/shipping/rtw/rtwdemo_pmsmfoc_script.html)

## **Simulink Real-Time**

Examples demonstrating features of Simulink Real-Time.

(real-time parameter tuning, signal monitoring, data logging, and more)

<http://www.mathworks.com/products/simulink-real-time/examples.html>

# Summary

# Integrated Workflow

Seamless transition from  
design to real-time testing

# Early Verification

Discover hardware/software integration issues during lab testing

# Leveraging Test Data

Improve your models and  
design from test data