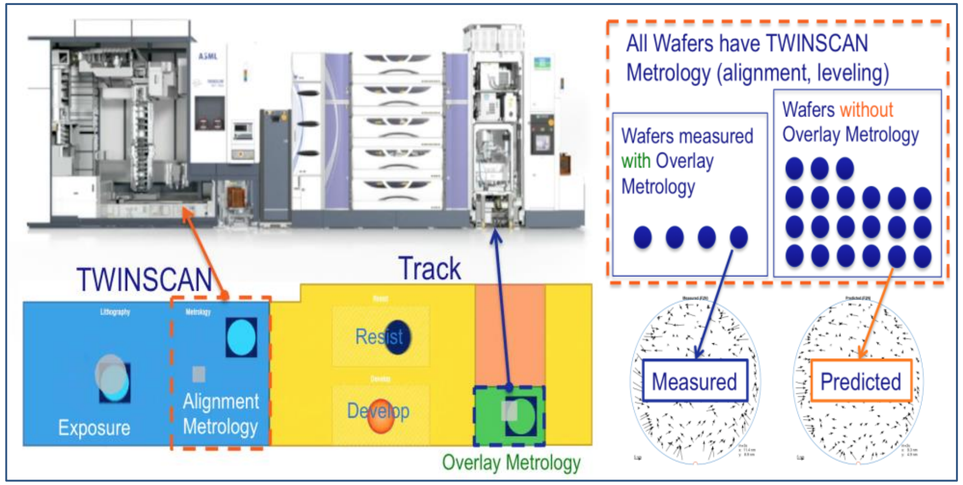
MATLAB EXPO 2016

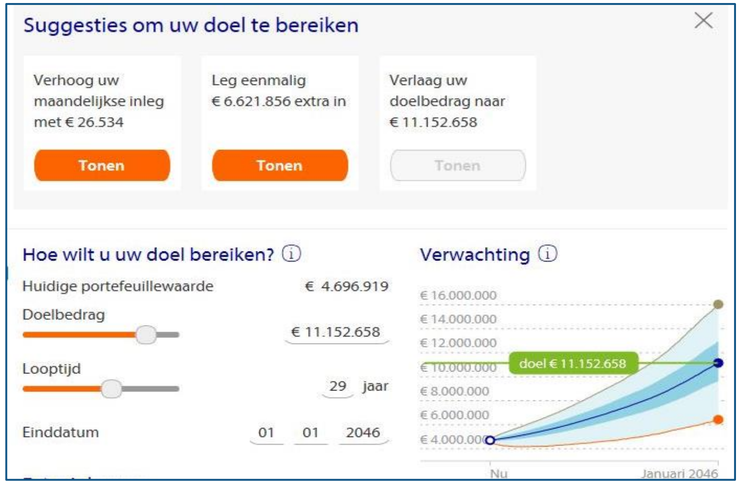
What’s New in MATLAB and Simulink

Mohamed Anas
Stephan van Beek

R2015b R2016a


All Wafers have TWINSCAN Metrology (alignment, leveling)
Wafers measured with Overlay Metrology
Wafers without Overlay Metrology
TWINSCAN
Track
Exposure
Alignment Metrology
Resist
Develop
Overlay Metrology
Measured
Predicted

ASML Develops Virtual Metrology Technology for Semiconductor Manufacturing with Machine Learning


Suggesties om uw doel te bereiken

Verhoog uw maandelijkse inleg met € 26.534

Tonen

Leg eenmalig € 6.621.856 extra in

Tonen

Verlaag uw doelbedrag naar € 11.152.658

Tonen

Hoe wilt u uw doel bereiken?

Huidige portefeuillewaarde	€ 4.696.919
Doelbedrag	€ 11.152.658
Looptijd	29 jaar
Einddatum	01 01 2046

Verwachting

Rabobank Develops Goal Monitor to Optimize Portfolio of their Customers

Vintecc Develops PLC System for Multi-Axle Harvesting Machine Using Model-Based Design

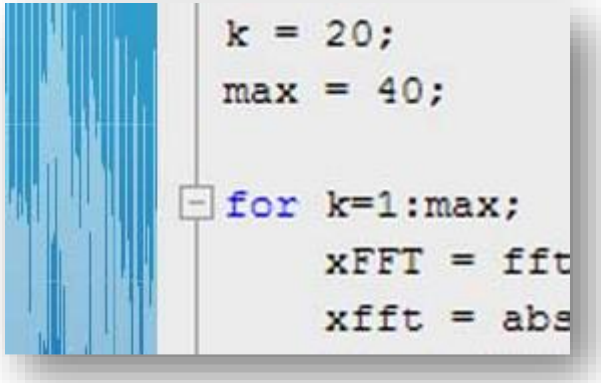
Punch Powertrain Develops Faster and more Accurate Control for Switched Reluctance Motors using Zynq SoC

InMotion Student Team Develops the Racing Car of the Future

MATLAB EXPO 2016

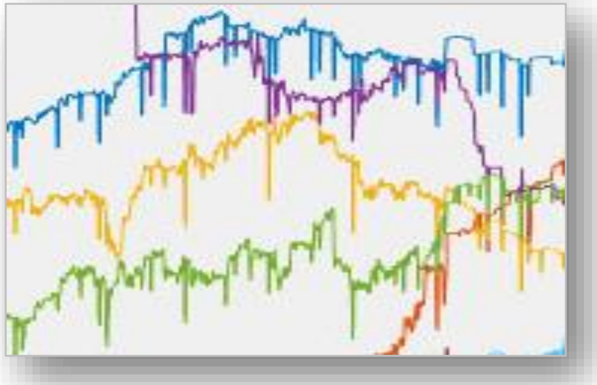
Engineers and scientists…

Engineers and scientists…



```
k = 20;
max = 40;

for k=1:max;
    xFFT = fft
    xfft = abs
```

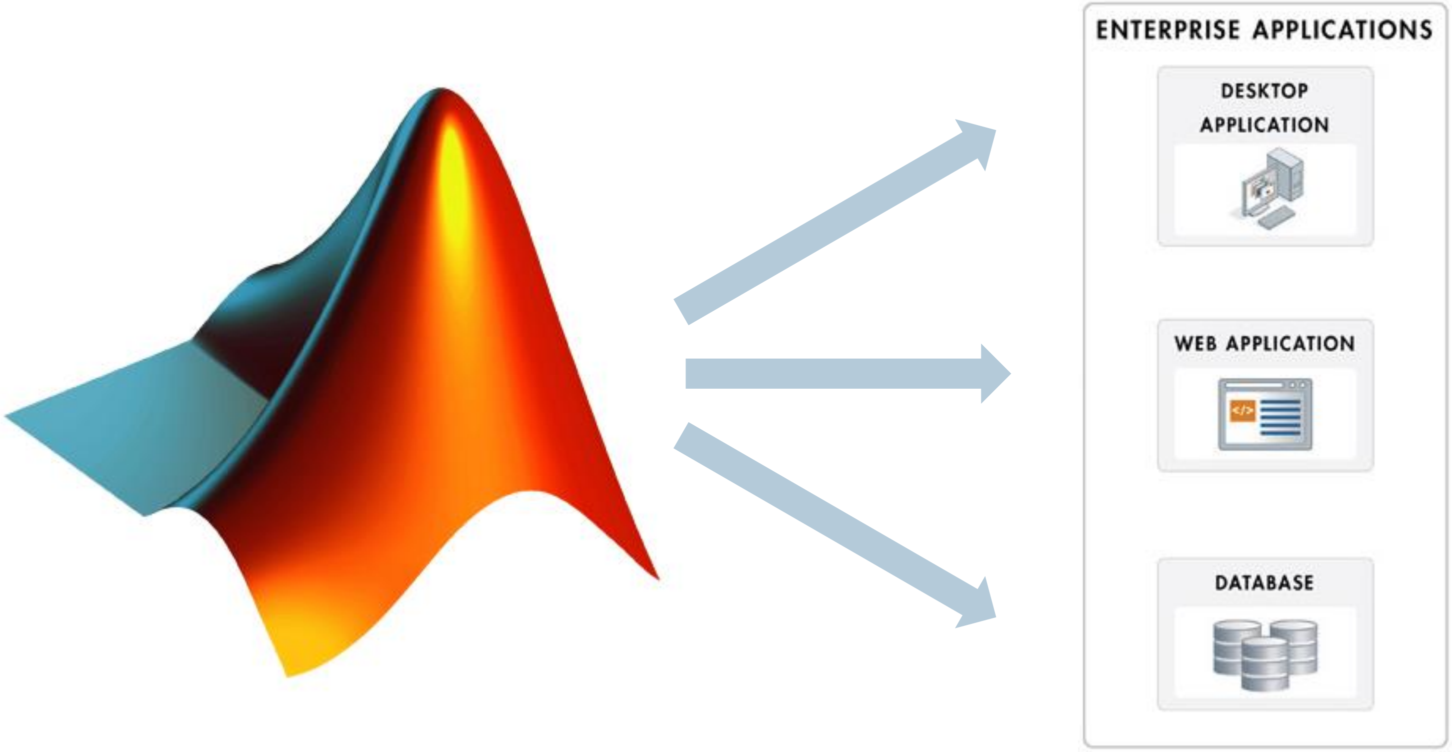
Develop algorithms



Analyze data

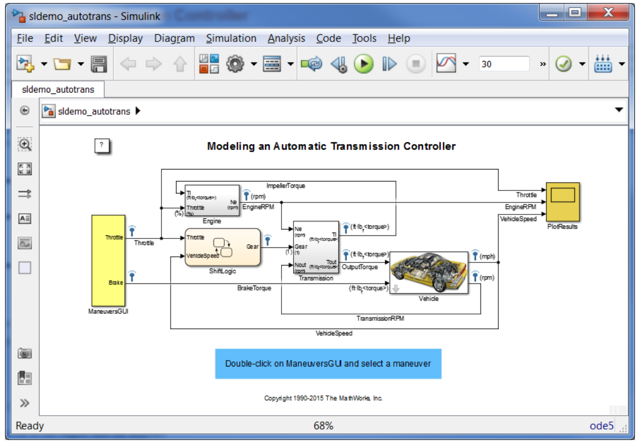
write MATLAB code.

Engineers and scientists…



deploy algorithms and applications within web, enterprise, and production systems.

Engineers and scientists…

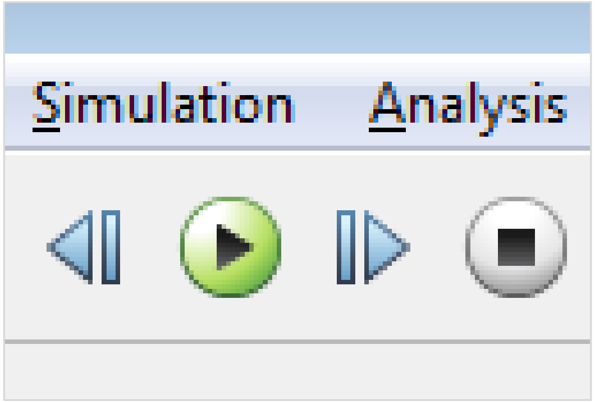


Modeling an Automatic Transmission Controller

Double-click on ManeuversGUI and select a maneuver

Copyright 1990-2015 The MathWorks, Inc.

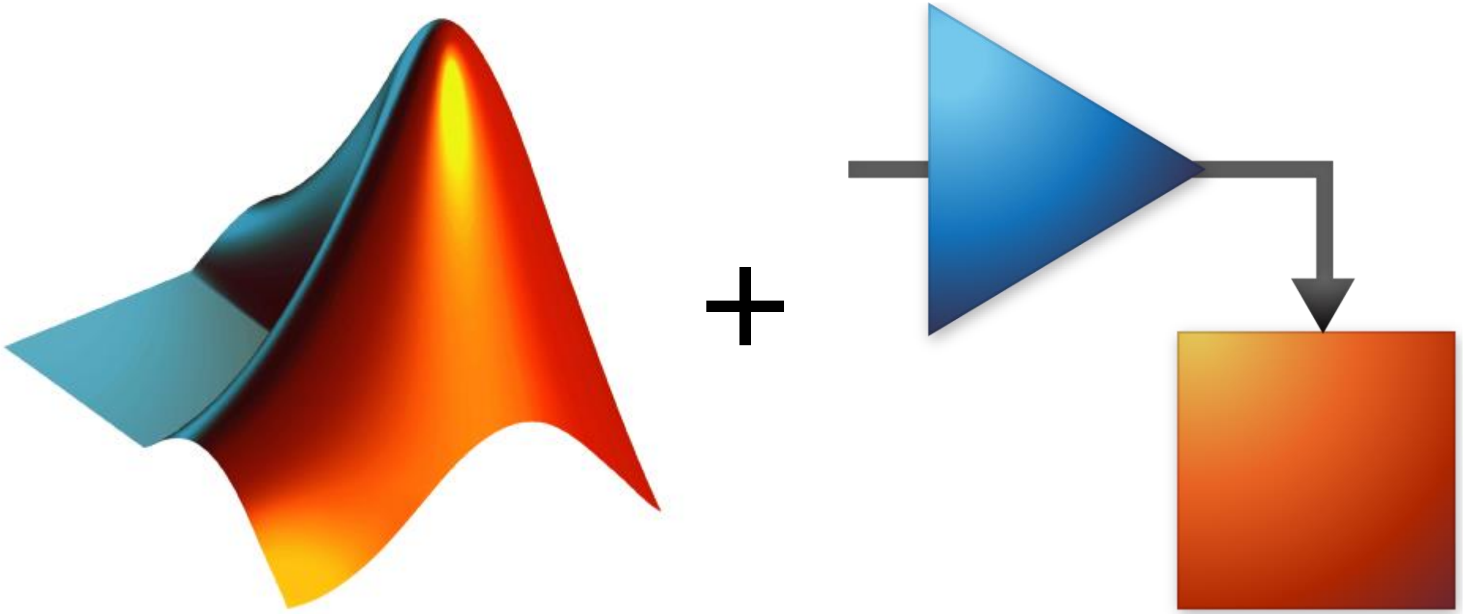
Model systems



Run simulations

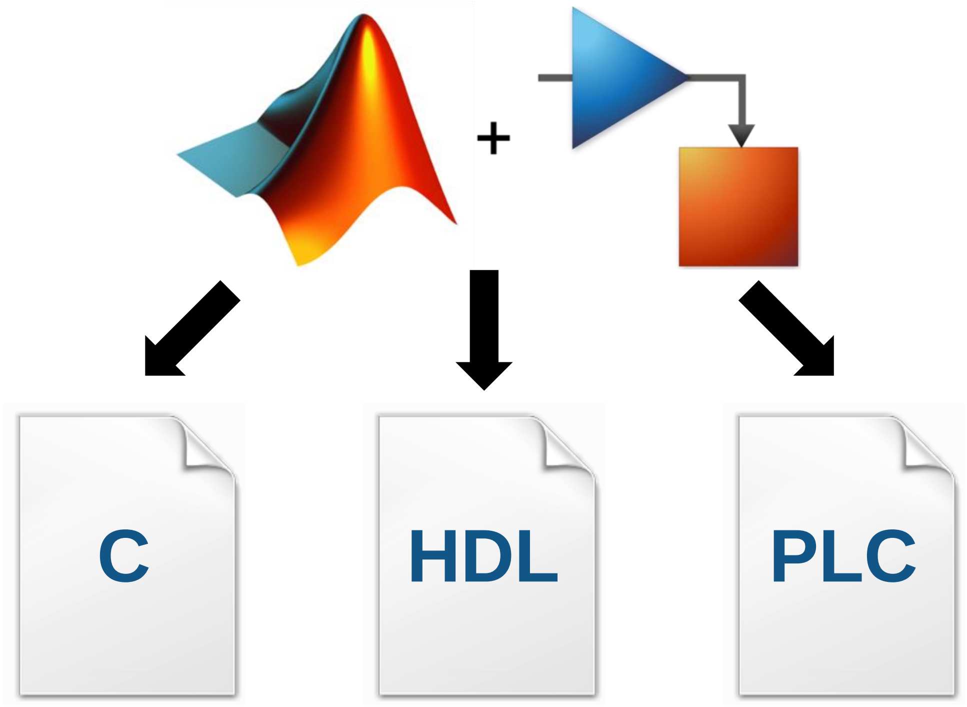
build Simulink models.

Engineers and scientists…



combine MATLAB code and Simulink models together.

Engineers and scientists…



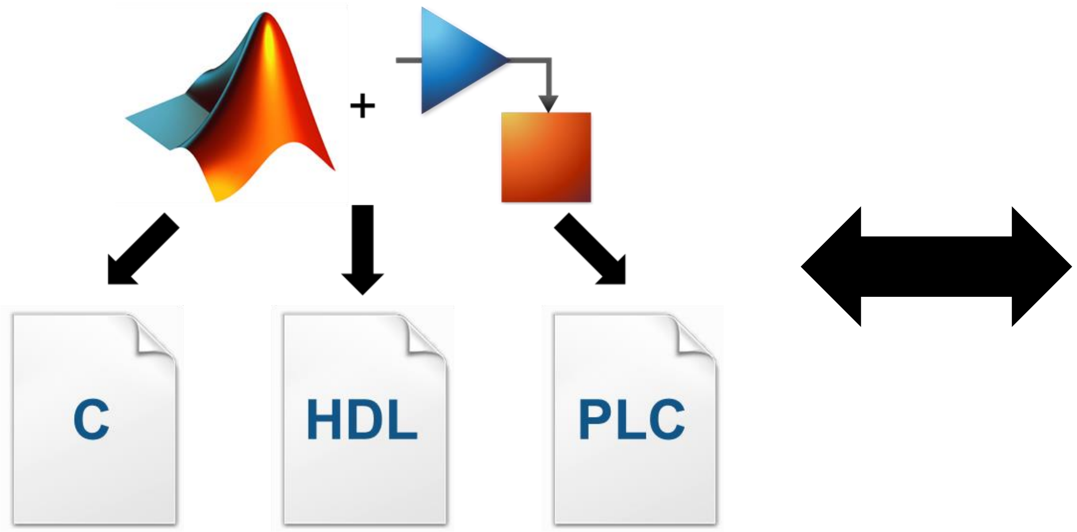
C

HDL

PLC

generate code.

Engineers and scientists…



connect software to hardware.

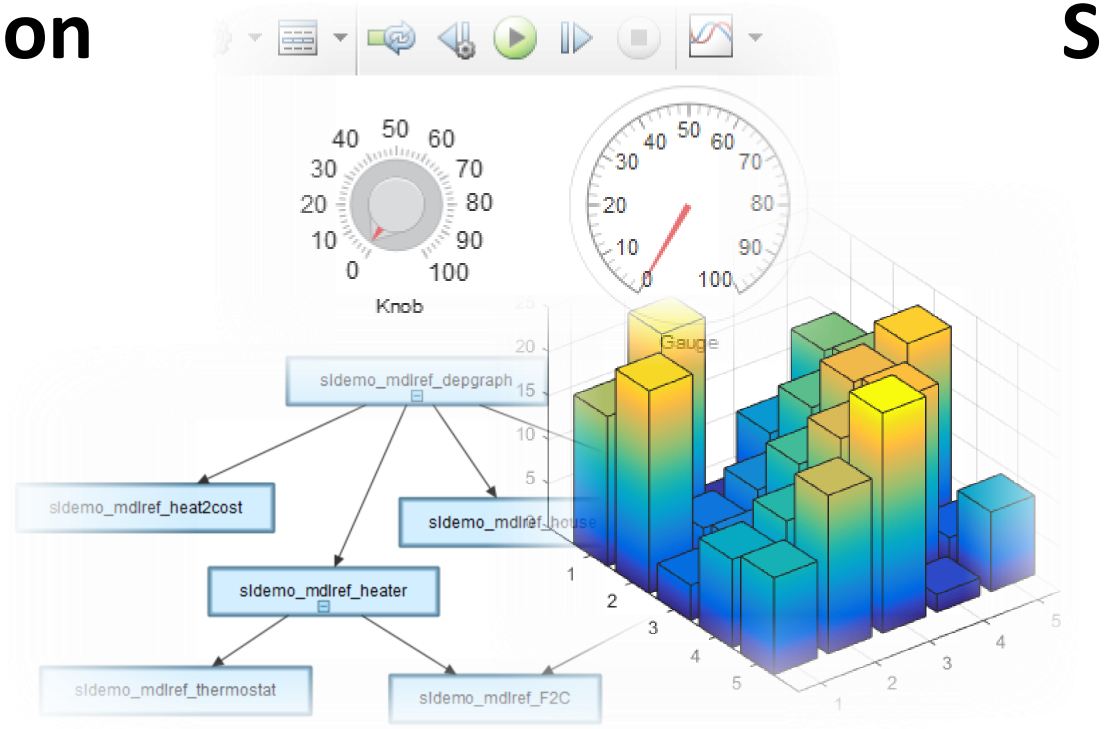
And it’s all easier to do in the latest releases.

R2015b

R2016a

Analysis and Visualization

Modeling and Simulation



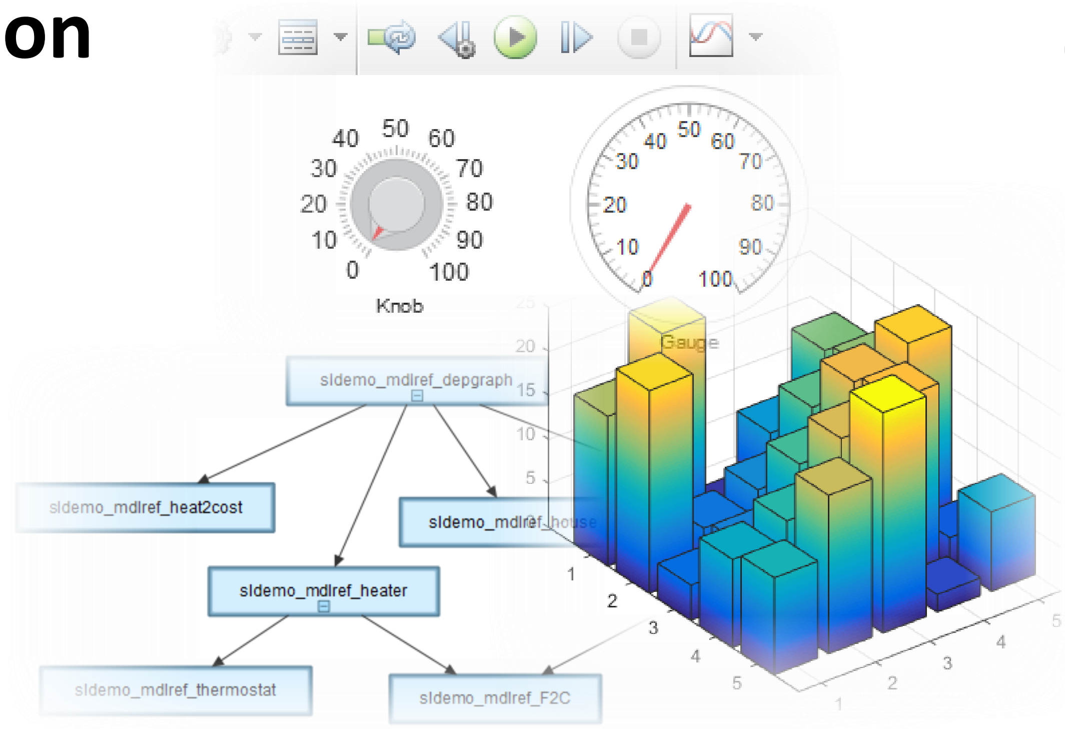
Testing and Verification

Sharing and Collaboration

Performance

Analysis and Visualization

Modeling and Simulation



Testing and Verification

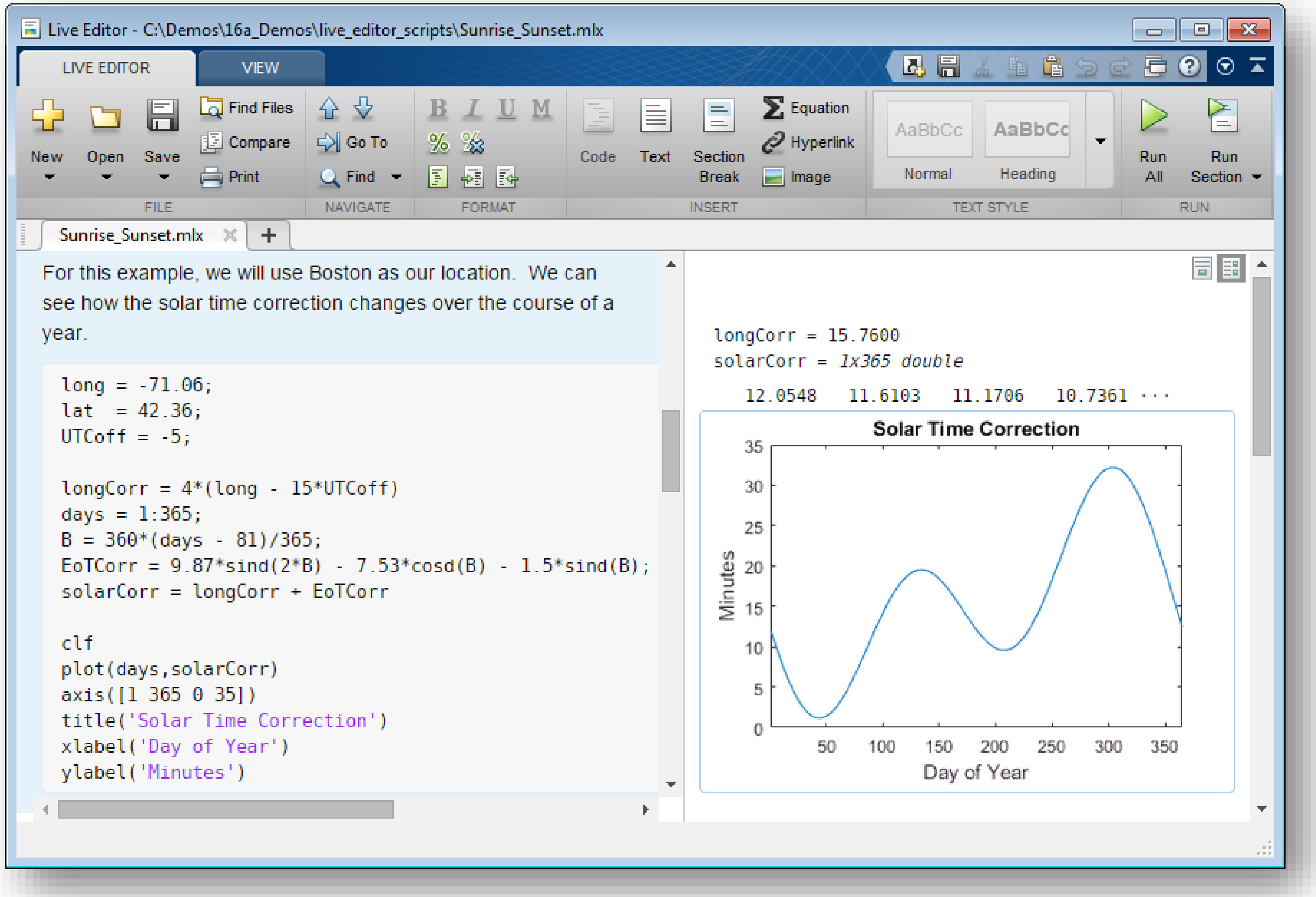
Sharing and Collaboration

Performance

MATLAB Live Editor

Change the way you work in MATLAB

- See results together with the code that produced them, accelerating exploratory programming and analysis
- Add equations, images, hyperlinks, and formatted text to create interactive narratives
- Create lectures that combine explanatory text, mathematical equations, code and results



For this example, we will use Boston as our location. We can see how the solar time correction changes over the course of a year.

```
long = -71.06;
lat  = 42.36;
UTCoff = -5;

longCorr = 4*(long - 15*UTCoff)
days = 1:365;
B = 360*(days - 81)/365;
EoTCorr = 9.87*sind(2*B) - 7.53*cosd(B) - 1.5*sind(B);
solarCorr = longCorr + EoTCorr

clf
plot(days,solarCorr)
axis([1 365 0 35])
title('Solar Time Correction')
xlabel('Day of Year')
ylabel('Minutes')
```

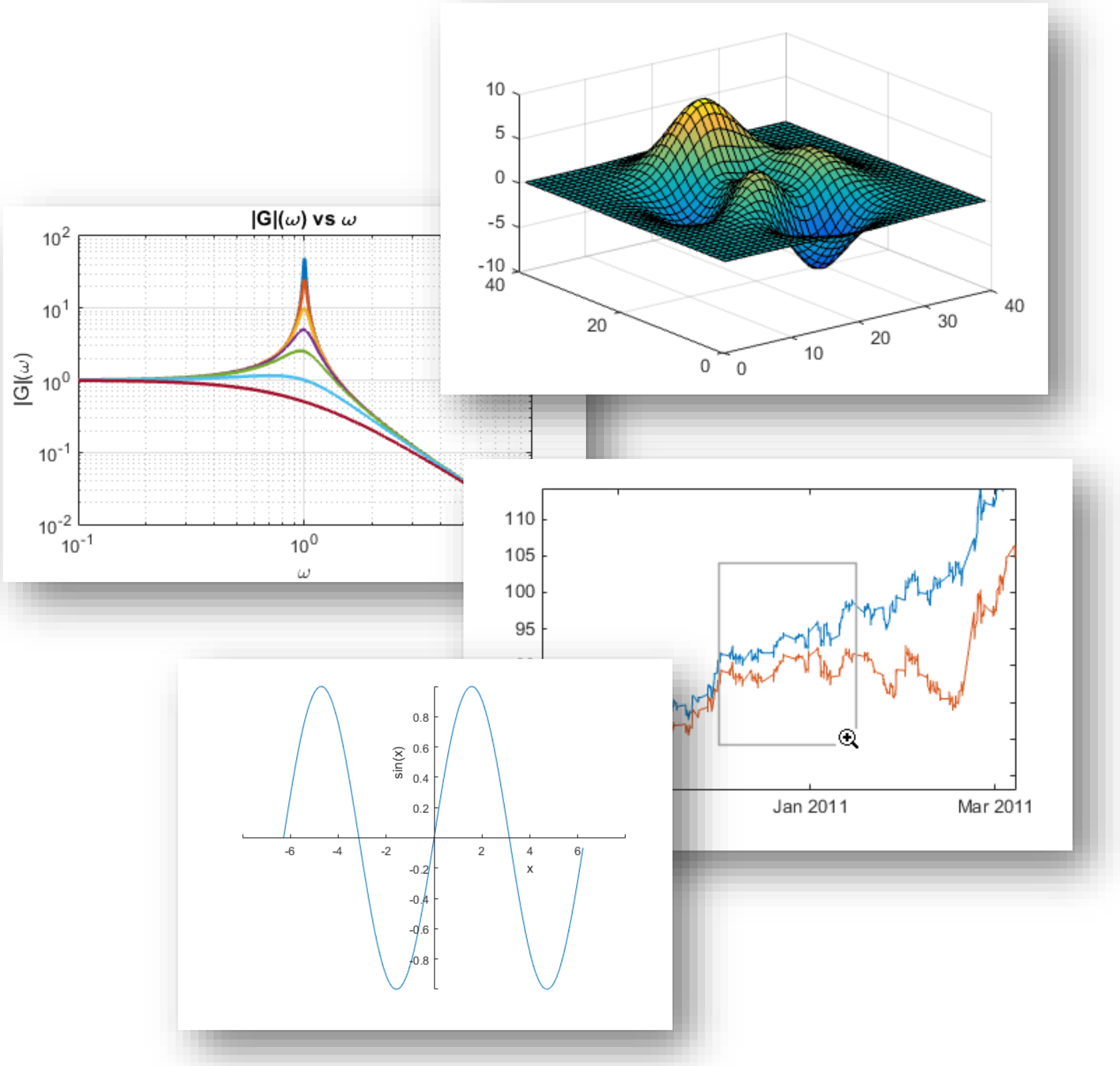
```
longCorr = 15.7600
solarCorr = 1x365 double

   12.0548    11.6103    11.1706    10.7361 ···
```

MATLAB Graphics

R2014b … R2016a

New look makes data easier to interpret and graphics objects are easier to customize

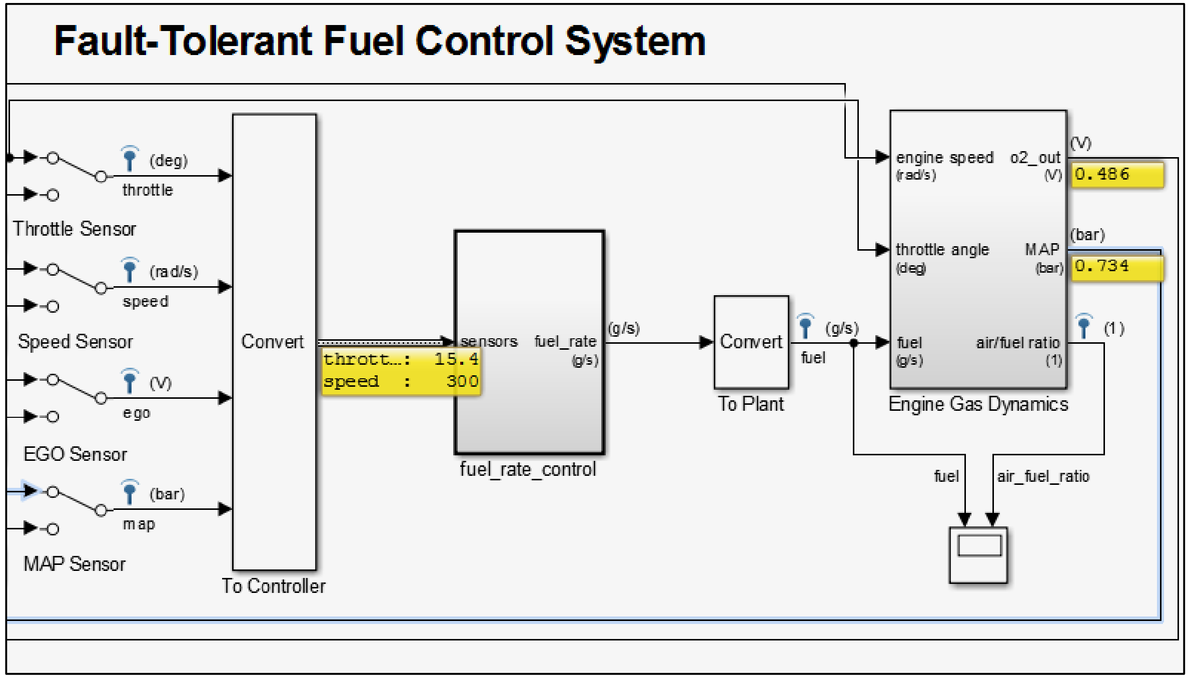
- New default line colors, fonts, and styles with anti-aliased graphics and fonts improve the clarity and aesthetics of MATLAB visualizations
- Steady stream of new features released
  - **R2014b** – rotatable tick labels, automatic updating of `datetime` tick labels, and new visualization functions (`histogram`, `animatedline`)
  - **R2015b** – increased control for customizing plot axes
  - **R2016a** – new functions for polar plots, multiple y-axis plots, and for plotting mathematical expressions and equations

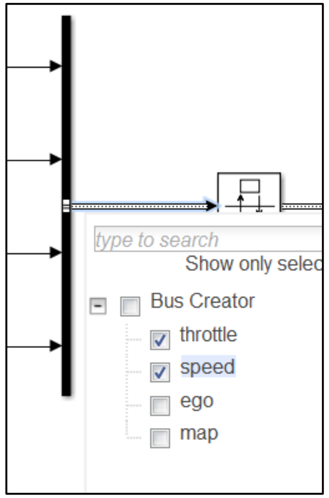


One-Click Display

Click a signal line when the simulation is running to view the current value

- Display port value for a signal by clicking it during simulation for easy debugging
- For bus signals, select the signals of interest before simulation

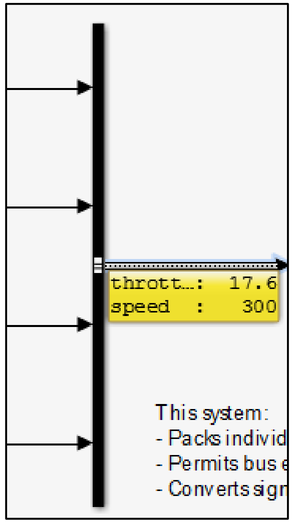



type to search

Show only selec

□ Bus Creator

- ☑ throttle
- ☑ speed
- □ ego
- □ map



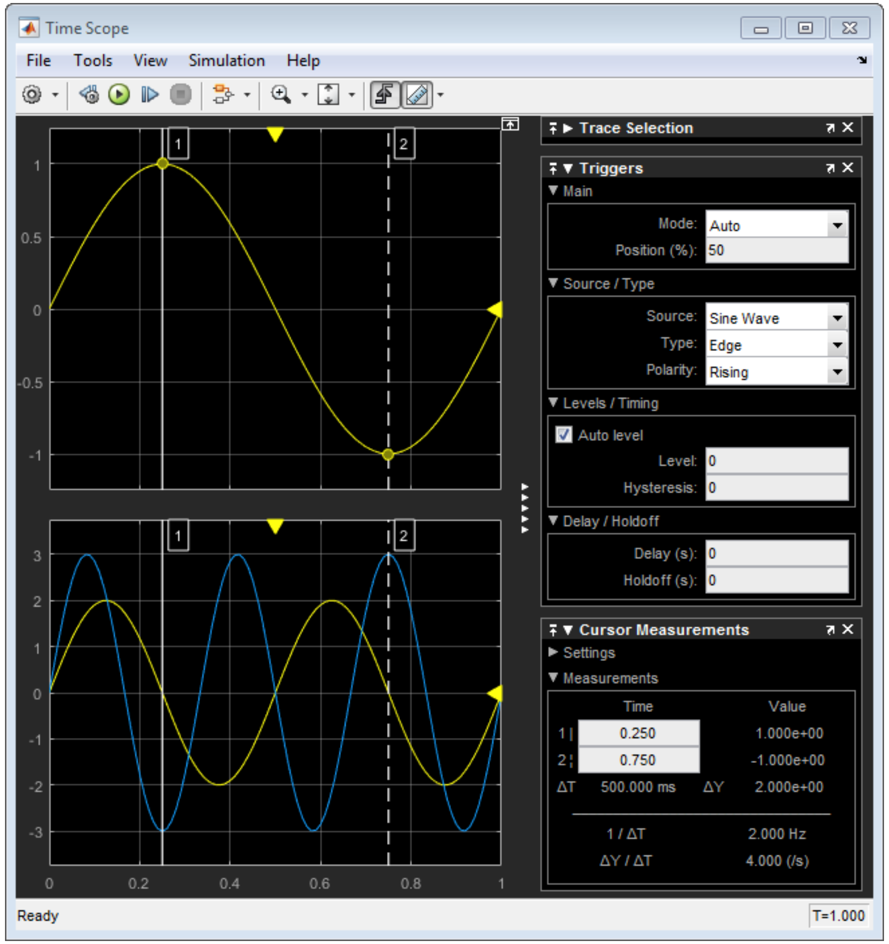
This system:
- Packs individ
- Permits bus e
- Converts sign

New Interface for Scopes

R2015b

View and debug signals with cursors and measurements

- Scope, Floating Scope, and Viewers all upgraded with new UI
- Includes simulation data analysis and debugging tools
  - Cursors
  - Measurements
  - Triggers



Trace Selection

Triggers

Main

Mode: Auto

Position (%): 50

Source / Type

Source: Sine Wave

Type: Edge

Polarity: Rising

Levels / Timing

☑ Auto level

Level: 0

Hysteresis: 0

Delay / Holdoff

Delay (s): 0

Holdoff (s): 0

Cursor Measurements

Settings

Measurements

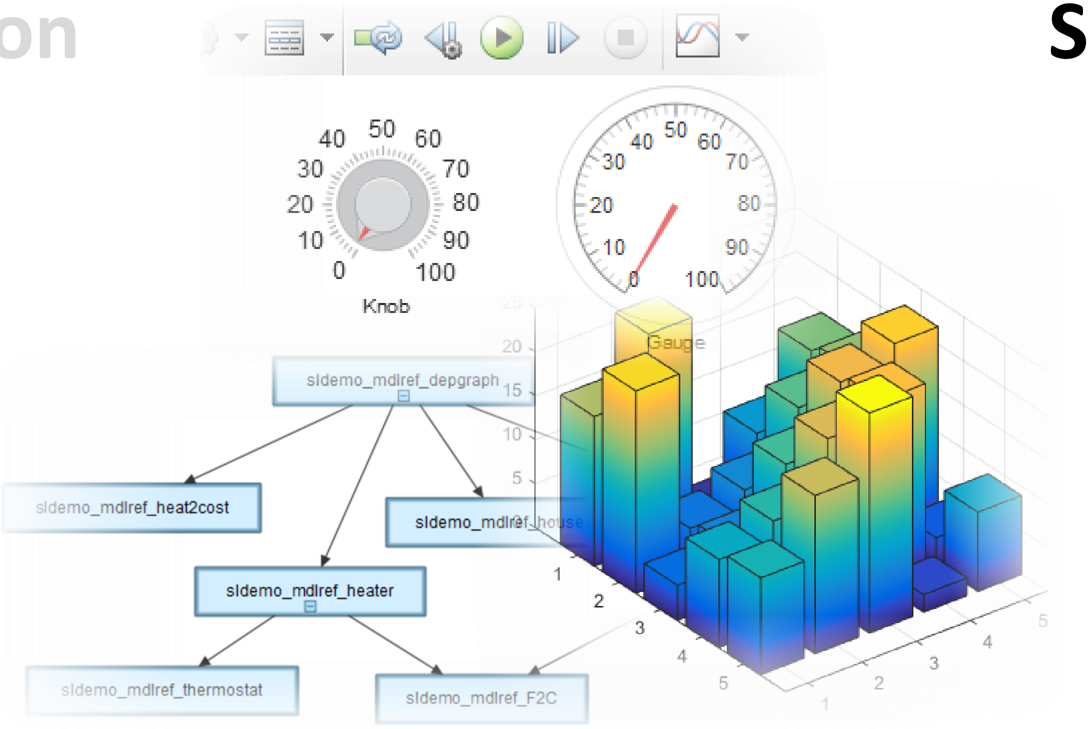
	Time		Value
1	0.250		1.000e+00
2	0.750		-1.000e+00
ΔT	500.000 ms	ΔY	2.000e+00
	1 / ΔT		2.000 Hz
	ΔY / ΔT		4.000 (/s)

Ready

T=1.000

Analysis and Visualization

Modeling and Simulation



Testing and Verification

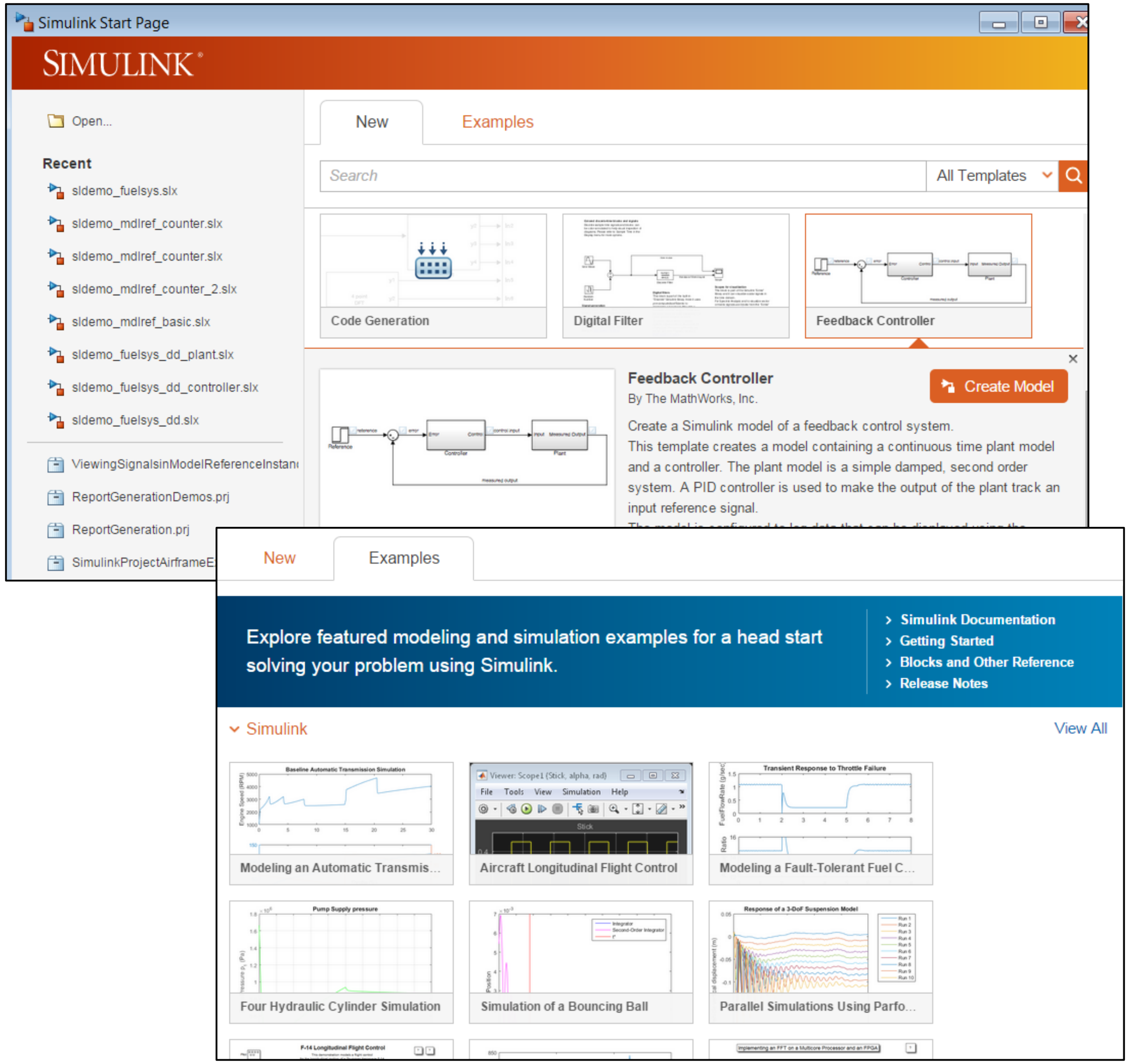
Sharing and Collaboration

Performance

Start Page

Get started or resume work faster by accessing templates, recent models, and featured examples

- Create new Simulink models using templates as starting points to common modeling approaches
- Define your own templates for standardization
- Use fully developed example models as a reference as you set out to build your own models
- Access most recent Simulink models right from the start page


Simulink Start Page
SIMULINK®
Open...
Recent
sldemo_fuelsys.slx
sldemo_mdlref_counter.slx
sldemo_mdlref_counter.slx
sldemo_mdlref_counter_2.slx
sldemo_mdlref_basic.slx
sldemo_fuelsys_dd_plant.slx
sldemo_fuelsys_dd_controller.slx
sldemo_fuelsys_dd.slx
ViewingSignalsinModelReferenceInstan
ReportGenerationDemos.prj
ReportGeneration.prj
SimulinkProjectAirframeE
New
Examples
Search
All Templates
Code Generation
Digital Filter
Feedback Controller
Create Model
Simulink
View All
Modeling an Automatic Transmis...
Aircraft Longitudinal Flight Control
Modeling a Fault-Tolerant Fuel C...
Four Hydraulic Cylinder Simulation
Simulation of a Bouncing Ball
Parallel Simulations Using Parfo...
Simulink Documentation
Getting Started
Blocks and Other Reference
Release Notes

Feedback Controller

By The MathWorks, Inc.

Create a Simulink model of a feedback control system. This template creates a model containing a continuous time plant model and a controller. The plant model is a simple damped, second order system. A PID controller is used to make the output of the plant track an input reference signal.

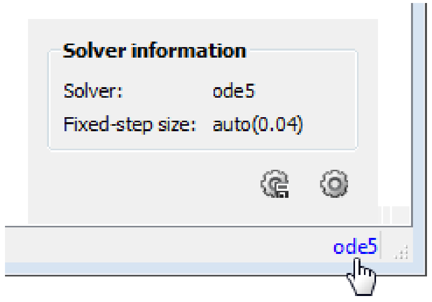
Explore featured modeling and simulation examples for a head start solving your problem using Simulink.

Automatic Solver Option

R2015b

Set up and simulate your model more quickly with automatically selected solver settings

- Simulink will select a solver and step size that is optimized for your specific model
- Considers factors such as model stiffness and simulation performance
- All new Simulink models use the automatic solver option
- Can optionally lock down solver so that it does not change from one simulation to another

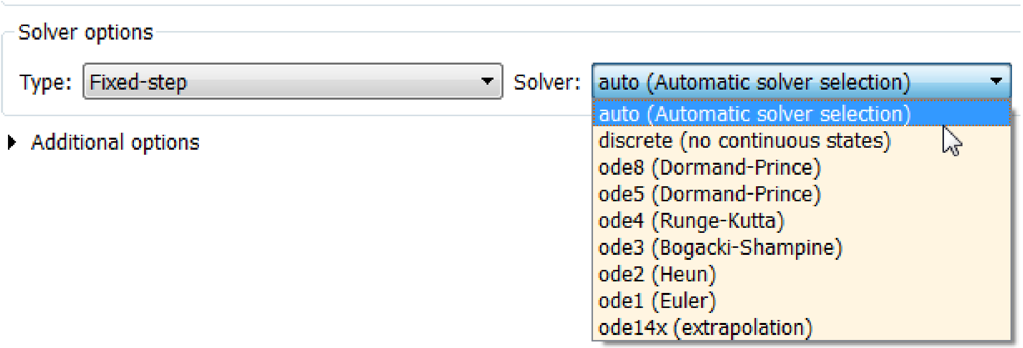


Solver information

Solver: ode5

Fixed-step size: auto(0.04)

ode5



Solver options

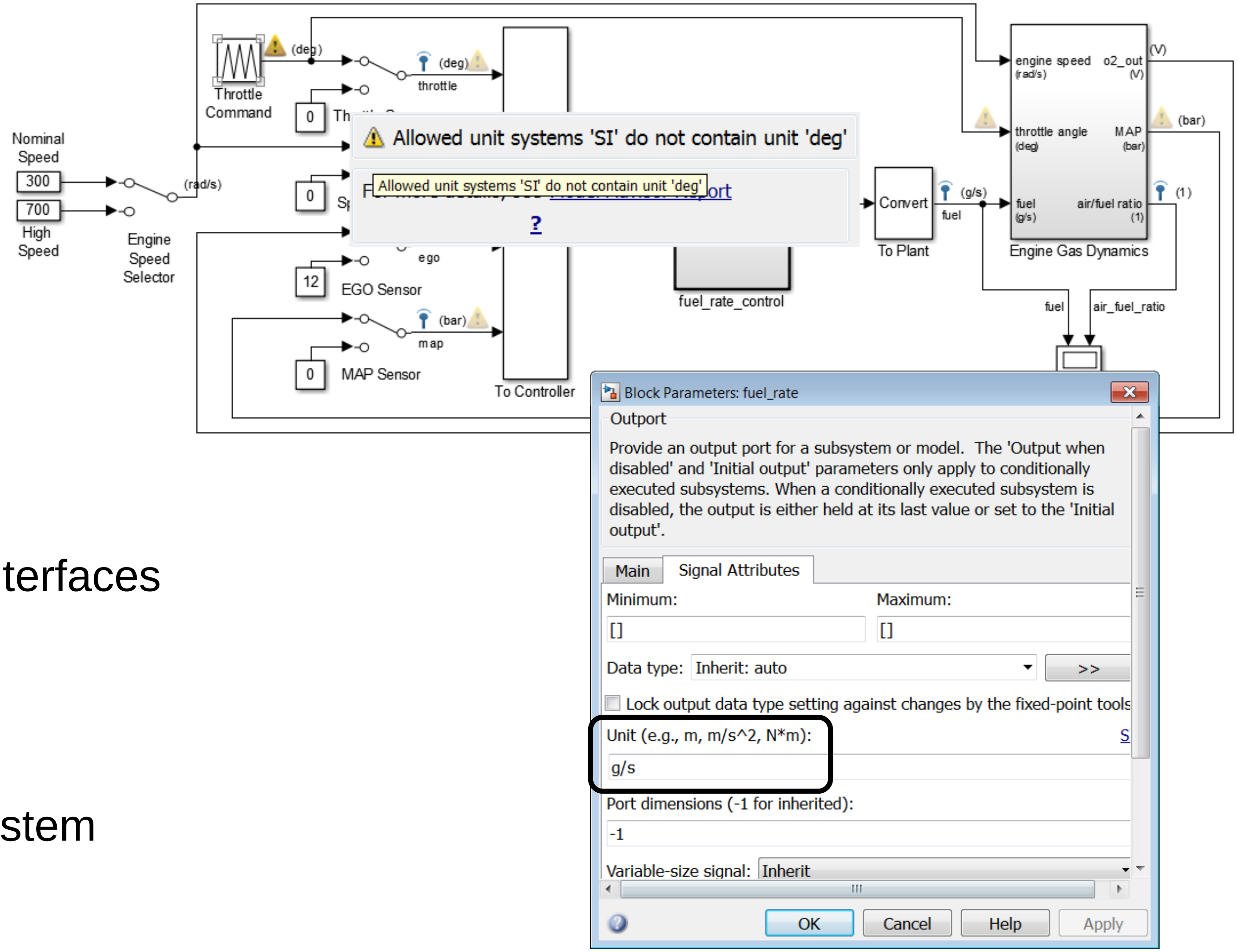
Type: Fixed-step

Solver: auto (Automatic solver selection)

Simulink Units

R2016a

Specify, visualize, and check consistency of units on interfaces



⚠ Allowed unit systems 'SI' do not contain unit 'deg'

Allowed unit systems 'SI' do not contain unit 'deg'

Block Parameters: fuel_rate

Outport

Provide an output port for a subsystem or model. The 'Output when disabled' and 'Initial output' parameters only apply to conditionally executed subsystems. When a conditionally executed subsystem is disabled, the output is either held at its last value or set to the 'Initial output'.

Minimum: []

Maximum: []

Data type: Inherit: auto

□ Lock output data type setting against changes by the fixed-point tools

Unit (e.g., m, m/s^2, N*m): g/s

Port dimensions (-1 for inherited): -1

Variable-size signal: Inherit

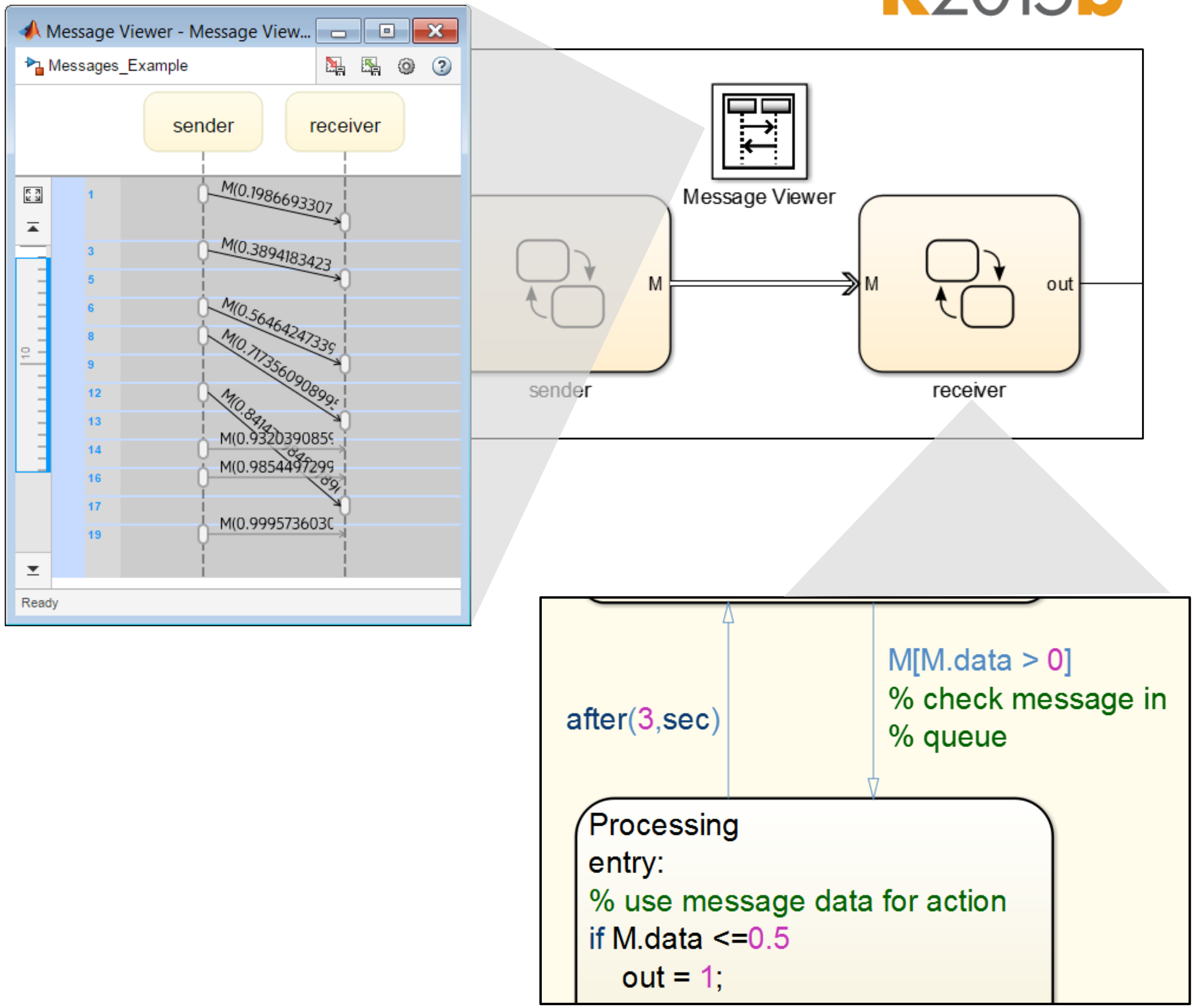
- Specify physical units for Simulink signals and bus elements
- Identify unit mismatches at the component interfaces
- Automatically convert units
- Enforce consistency by restricting the unit system

Messages

R2015b

Model asynchronous operations in state charts using objects that carry data and can be queued

- New message object and queue
- Message Viewer block to visualize lifetime of a message
- Signal lines in Simulink to transfer messages between charts

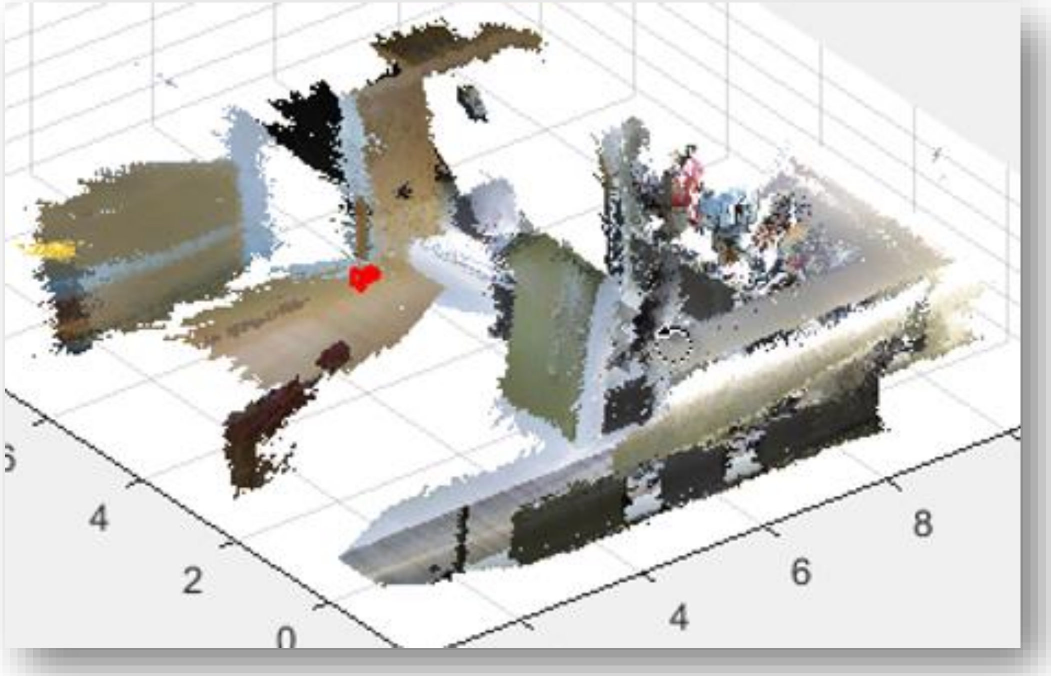


```
after(3,sec)
M[M.data > 0]
% check message in
% queue
Processing
entry:
% use message data for action
if M.data <=0.5
    out = 1;
```

3D Vision

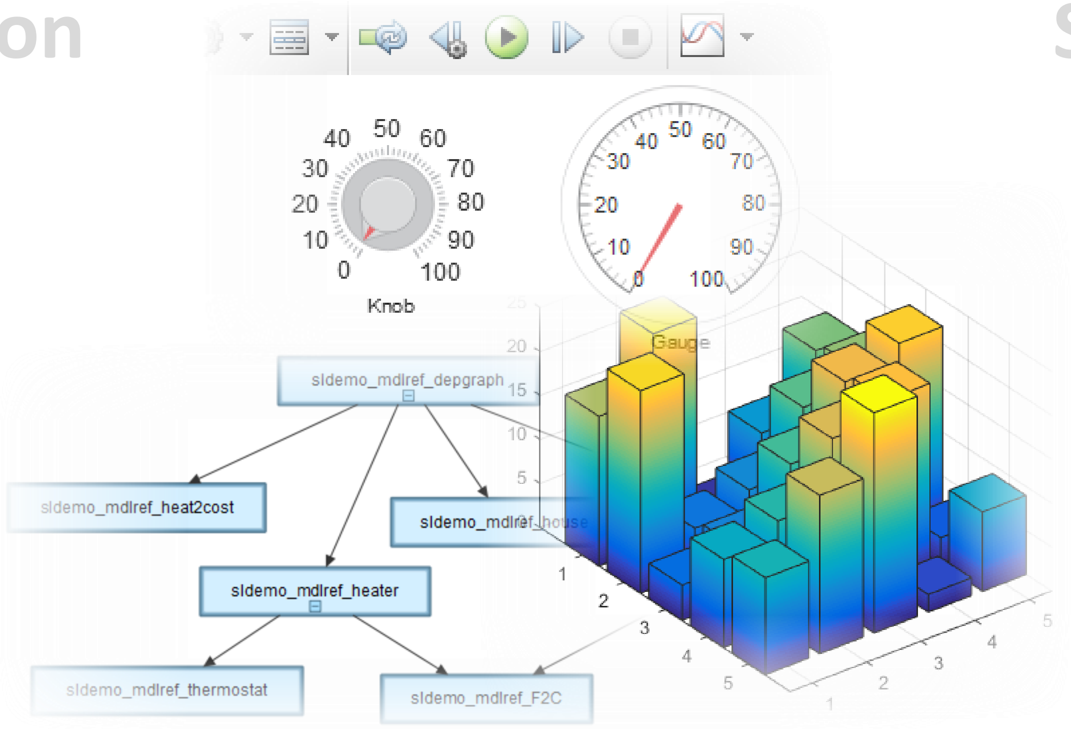
Enables autonomous systems to map and measure the world

- Supports workflows for ADAS, autonomous driving, and robotics
- New functionality to support:
  - 3D point cloud processing
  - Structure from motion



Analysis and Visualization

Modeling and Simulation



Testing and Verification

Sharing and Collaboration

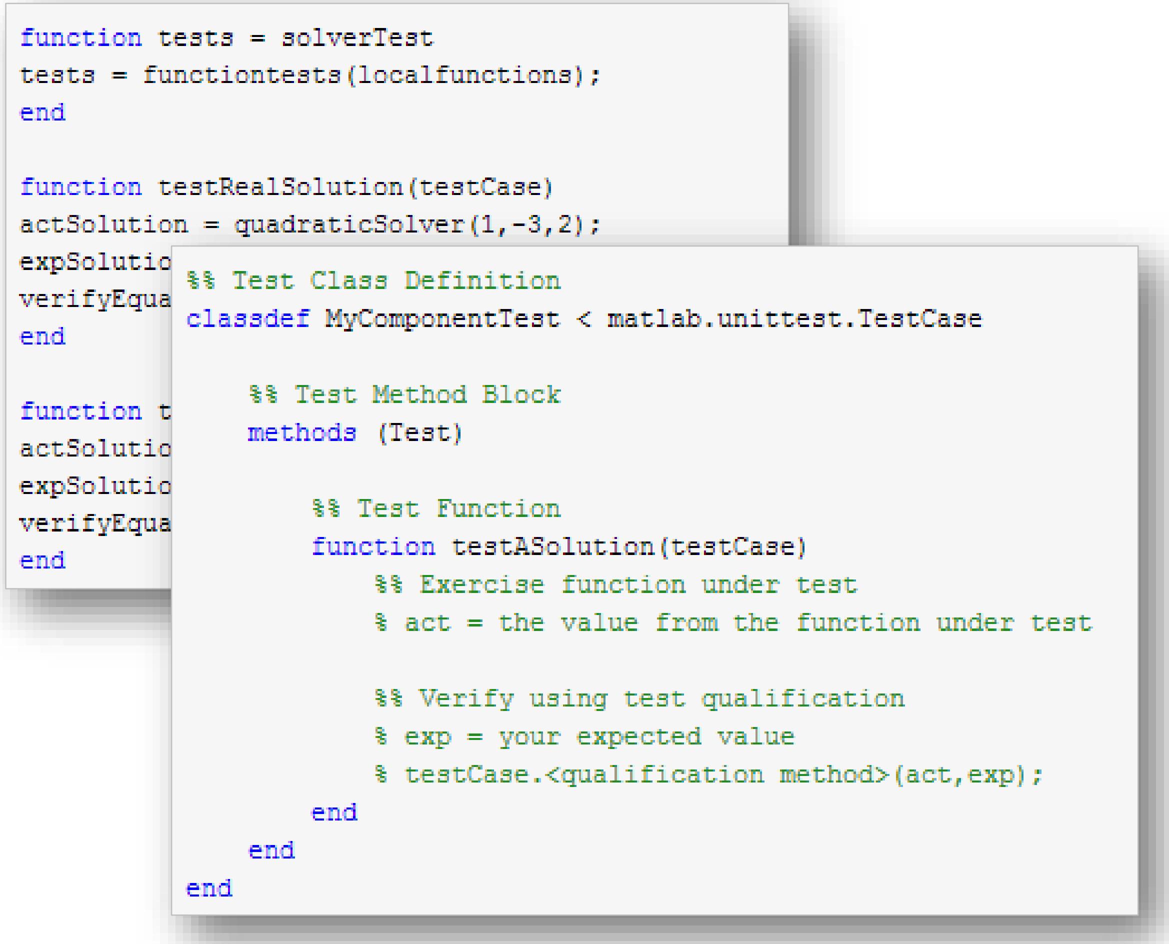
Performance

MATLAB Unit Testing Framework

R2013a … R2016a

Write and run unit tests, and analyze test results

- xUnit-style testing framework for the MATLAB language
- Includes a set of readily available qualification methods, and supports automation, providing easy reuse of test-cases
- Includes script-based, function-based, and object-oriented interfaces



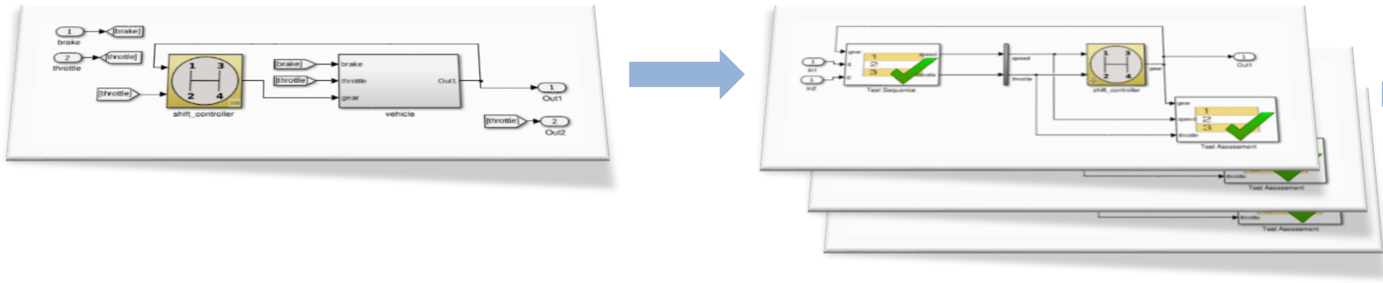
```
function tests = solverTest
tests = functiontests(localfunctions);
end

function testRealSolution(testCase)
actSolution = quadraticSolver(1,-3,2);
expSolutio
verifyEqua
end

function t
actSolutio
expSolutio
verifyEqua
end
```

```
%% Test Class Definition
classdef MyComponentTest < matlab.unittest.TestCase

    %% Test Method Block
    methods (Test)

        %% Test Function
        function testASolution(testCase)
            %% Exercise function under test
            % act = the value from the function under test

            %% Verify using test qualification
            % exp = your expected value
            % testCase.<qualification method>(act,exp);
        end
    end
end
```

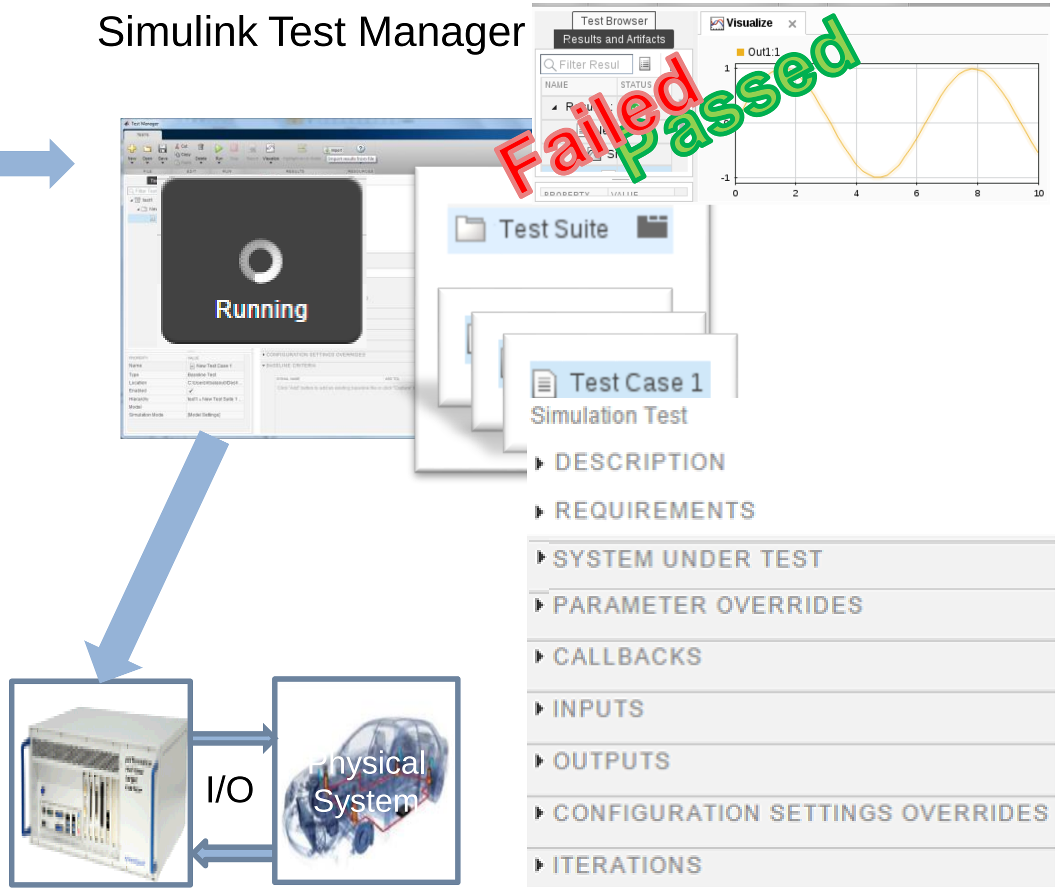
Simulink Test

R2015a

Simulink Test Harnesses



Simulink Test Manager



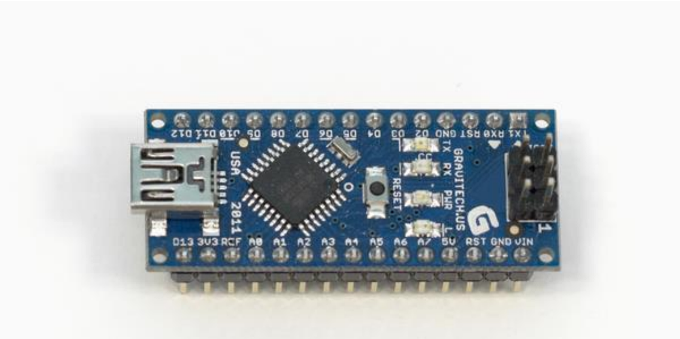
Author, execute and manage simulation-based testing

- Build synchronized executable test environments
- Create inputs and assessments based on logic or temporal conditions
- Integration with Real-Time Testing R2016a

Deploying to Hardware

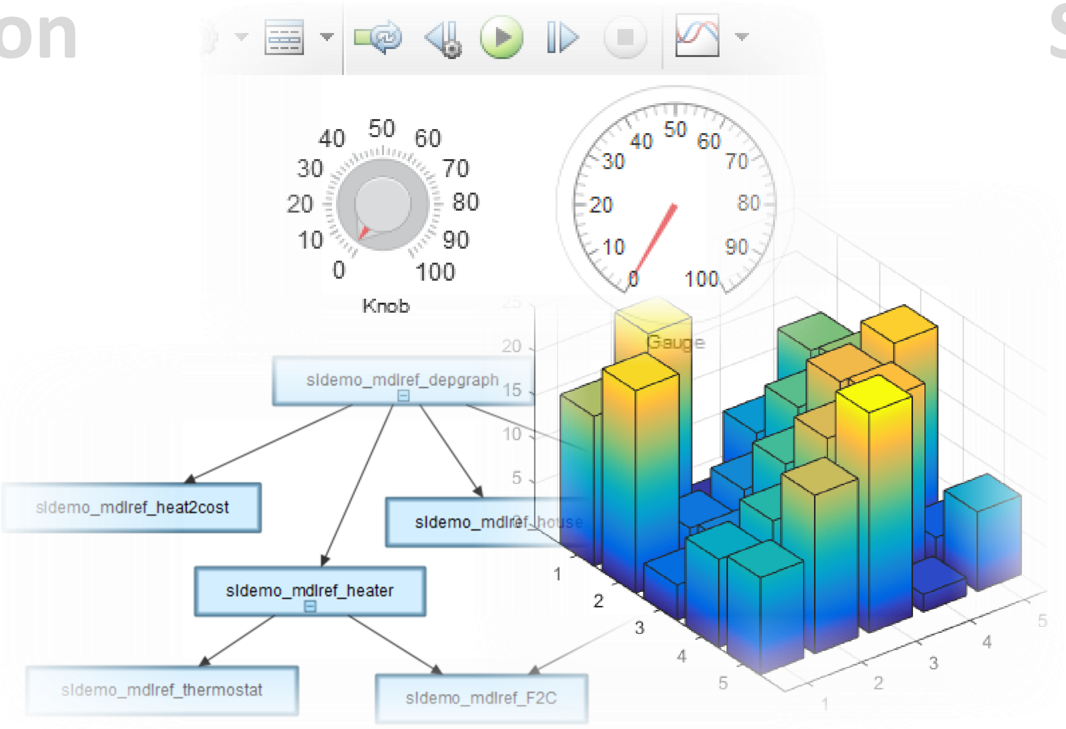
Run your models on low-cost hardware and stream data into MATLAB

- Acquire images from Raspberry Pi and Kinect V2 into MATLAB and Simulink
- Run Simulink models on Lego EV3, Raspberry Pi 3, Raspberry Pi 3, and Arduino Yun
- Adds to existing support for Arduino, Lego, and Raspberry Pi platforms



Analysis
and
Visualization

Modeling
and
Simulation



Testing
and
Verification

**Sharing
and
Collaboration**

Performance

App Designer

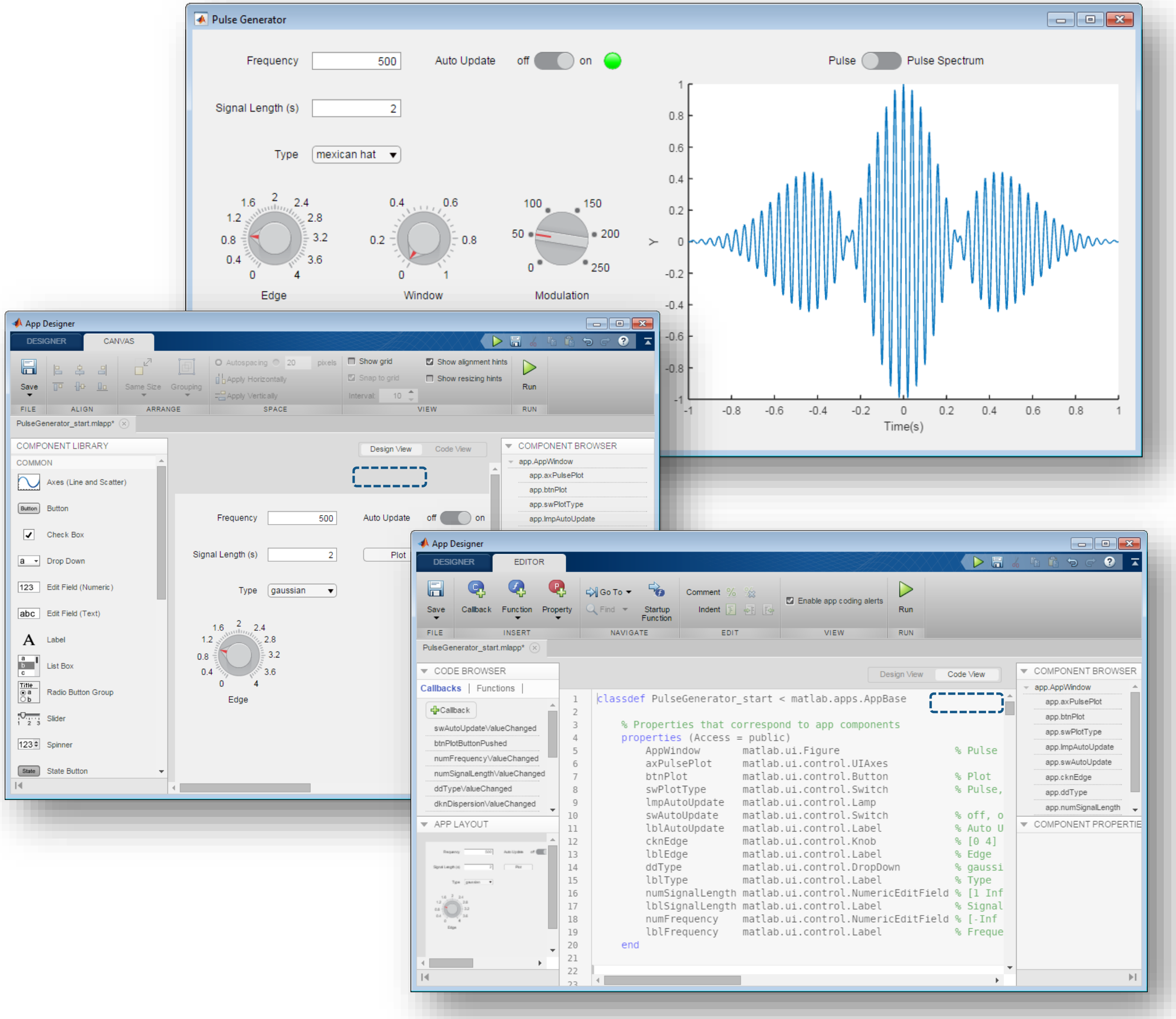
R2016a

Develop MATLAB applications with an enhanced design environment and expanded UI component set

- Choose from standard components (buttons, check boxes, panels, etc.), as well as gauges, lamps, knobs and switches

- Quickly move between visual design and code development

- New object-based code format makes it easier to share data between parts of the app



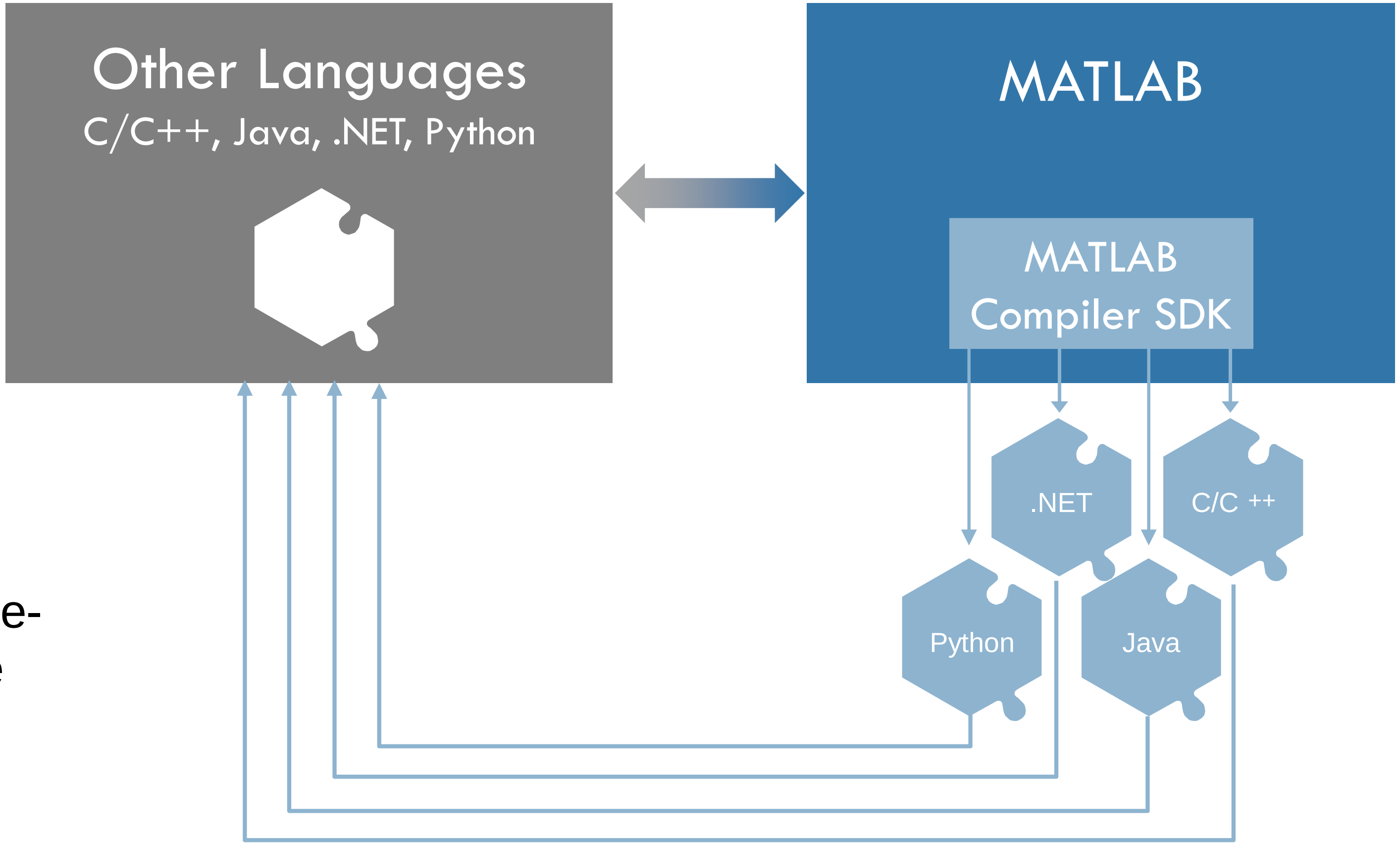
```
classdef PulseGenerator_start < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        AppWindow        matlab.ui.Figure                 % Pulse
        axPulsePlot      matlab.ui.control.UIAxes
        btnPlot          matlab.ui.control.Button         % Plot
        swPlotType       matlab.ui.control.Switch         % Pulse,
        lmpAutoUpdate    matlab.ui.control.Lamp
        swAutoUpdate     matlab.ui.control.Switch         % off, o
        lblAutoUpdate    matlab.ui.control.Label          % Auto U
        cknEdge          matlab.ui.control.Knob           % [0 4]
        lblEdge          matlab.ui.control.Label          % Edge
        ddType           matlab.ui.control.DropDown       % gaussi
        lblType          matlab.ui.control.Label          % Type
        numSignalLength  matlab.ui.control.NumericEditField % [1 Inf
        lblSignalLength  matlab.ui.control.Label          % Signal
        numFrequency     matlab.ui.control.NumericEditField % [-Inf
        lblFrequency     matlab.ui.control.Label          % Freque
    end
```

Using MATLAB with Other Languages

R2015b

Integrate MATLAB with other programming languages, including C/C++, Java, .NET, and Python

- Call MATLAB from another language
- Reuse legacy code written in another programming language within MATLAB
- Package MATLAB programs into language-specific software components to integrate with other programming languages
  - Python support added in **R2015b**

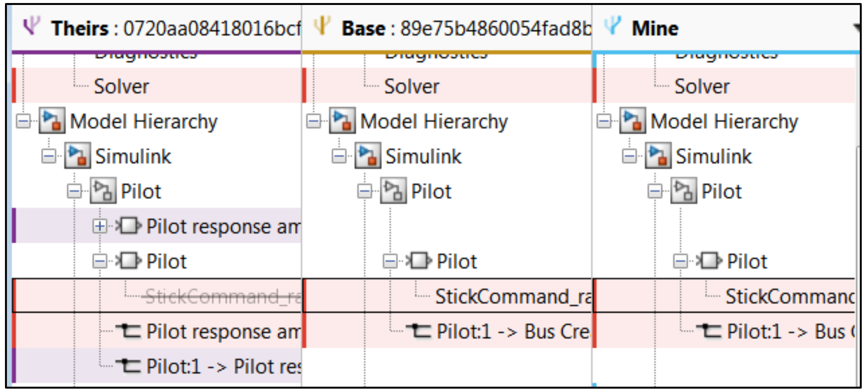


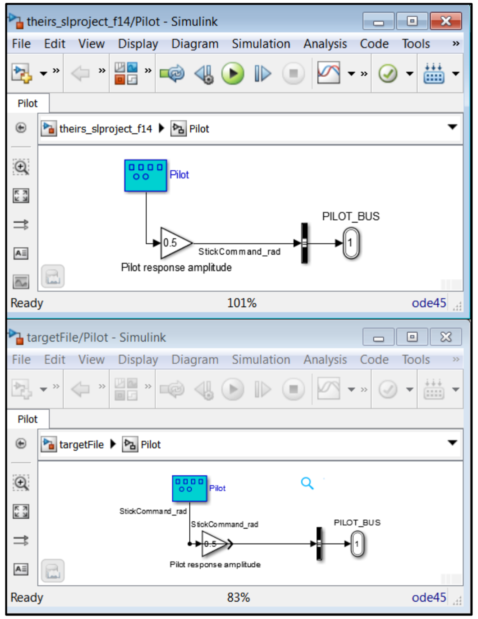
Three-Way Model Merge

R2016a

Graphically resolve conflicts between revisions within a Simulink project

- Resolve conflicts in model files under source control
- Provides an interactive comparison report with the two conflicting designs along with the original base model

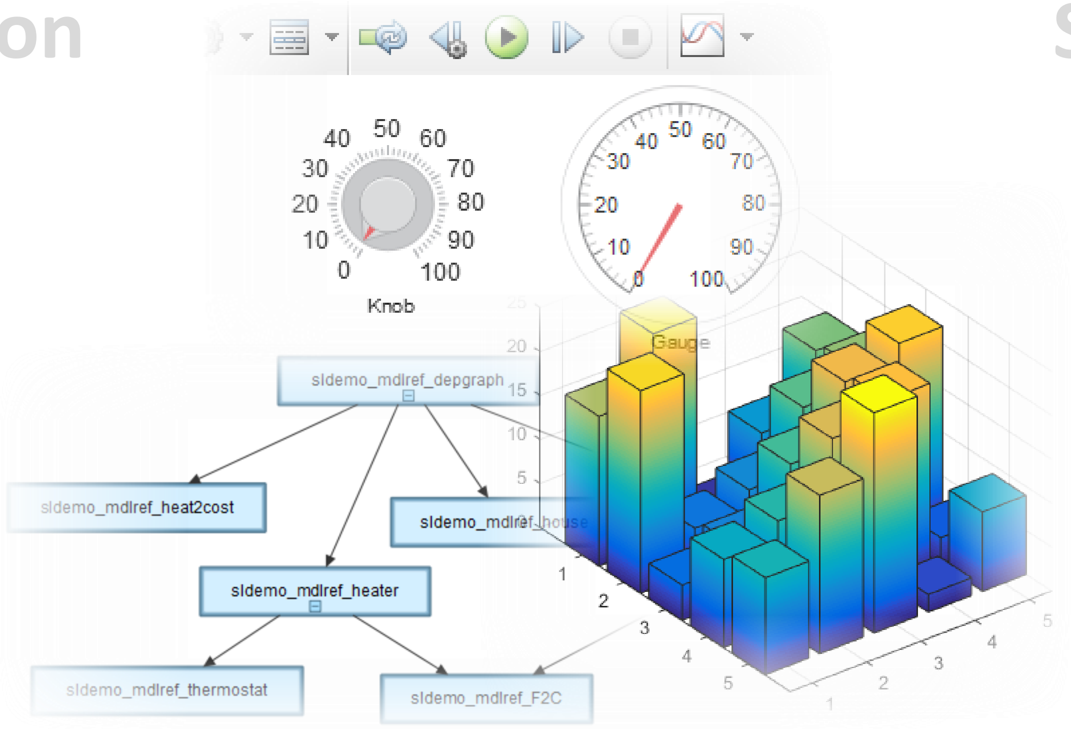




theirs_slproject_f14/Pilot - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools
Pilot
theirs_slproject_f14 ▸ Pilot
Pilot
PILOT_BUS
0.5
StickCommand_rad
1
Pilot response amplitude
Ready
101%
ode45
targetFile/Pilot - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools
Pilot
targetFile ▸ Pilot
Pilot
StickCommand_rad
StickCommand_rad
PILOT_BUS
0.5
1
Pilot response amplitude
Ready
83%
ode45

Analysis
and
Visualization

Modeling
and
Simulation



Testing
and
Verification

Sharing
and
Collaboration

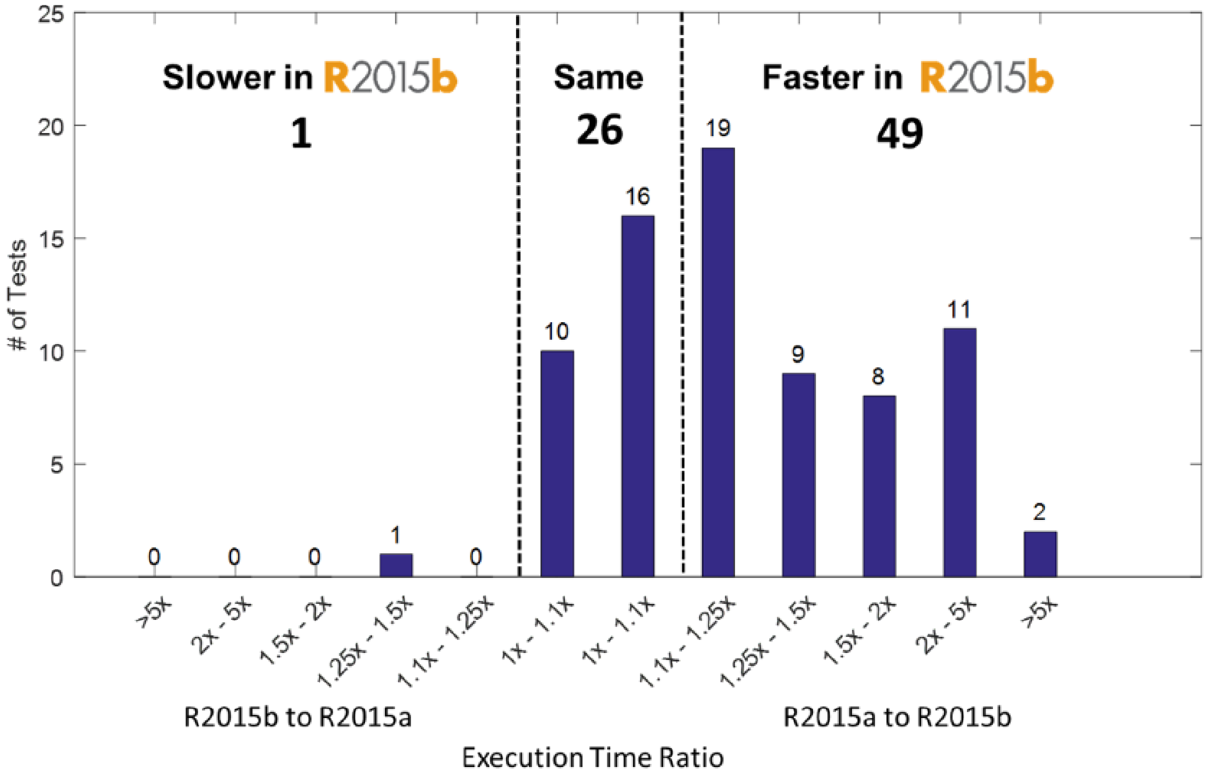
Performance

MATLAB Execution Engine

R2015b

Redesigned execution engine runs MATLAB code faster

- All MATLAB code can now be JIT compiled
- Average performance improvement of 40% on 76 performance-sensitive user applications
- A platform for future improvements
- Performance testing framework R2016a
  - Measure MATLAB code performance
  - Interface leverages the unit testing framework



GPU Acceleration and Parallel Computing

R2015b

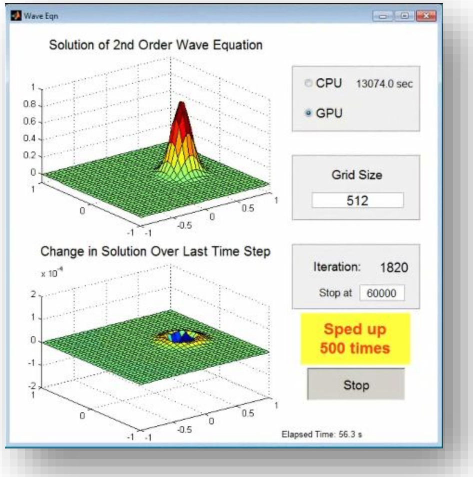
Perform parallel computations using GPUs

- Accelerate applications using GPU-enabled functions
  - > 300 in MATLAB
  - > 90 in Statistics and Machine Learning Toolbox
  - > 50 in Image Processing Toolbox

- Use enhanced gpuArray functions for sparse matrices on GPUs

```
Transfer data to GPU
>> GX = gpuArray(X);

GPGPU Computation
>> GY = fft2(GX);

Gather data to CPU
>> Y = gather(GY);
```

Simple GPU code in MATLAB


Wave Eqn

Solution of 2nd Order Wave Equation

○ CPU 13074.0 sec

● GPU

Grid Size: 512

Change in Solution Over Last Time Step

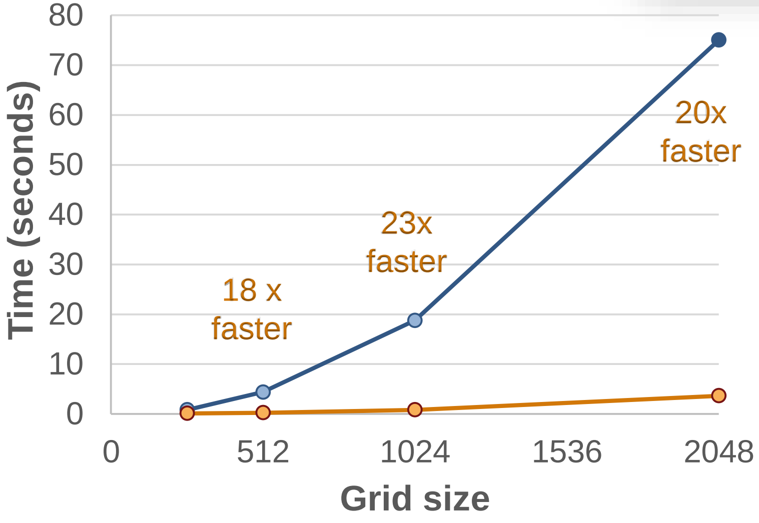
Iteration: 1820

Stop at: 60000

Sped up 500 times

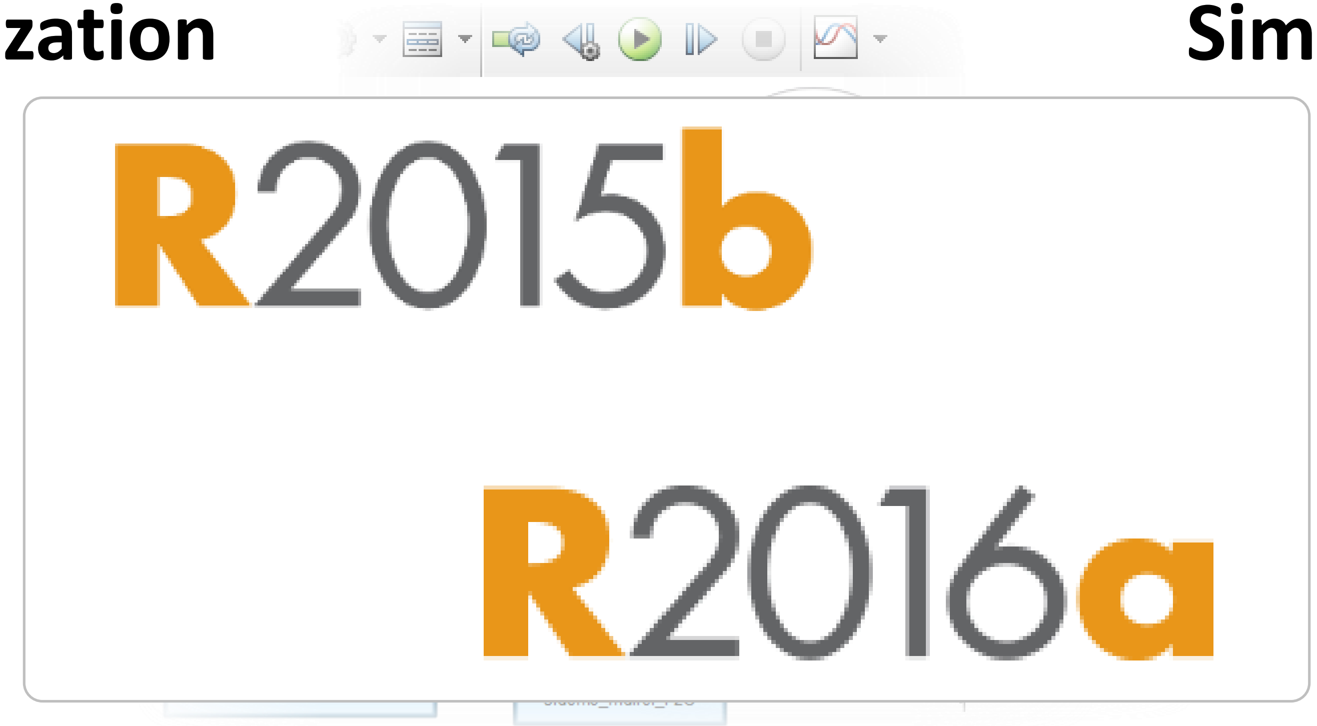
Stop

Elapsed Time: 56.3 s



Analysis and Visualization

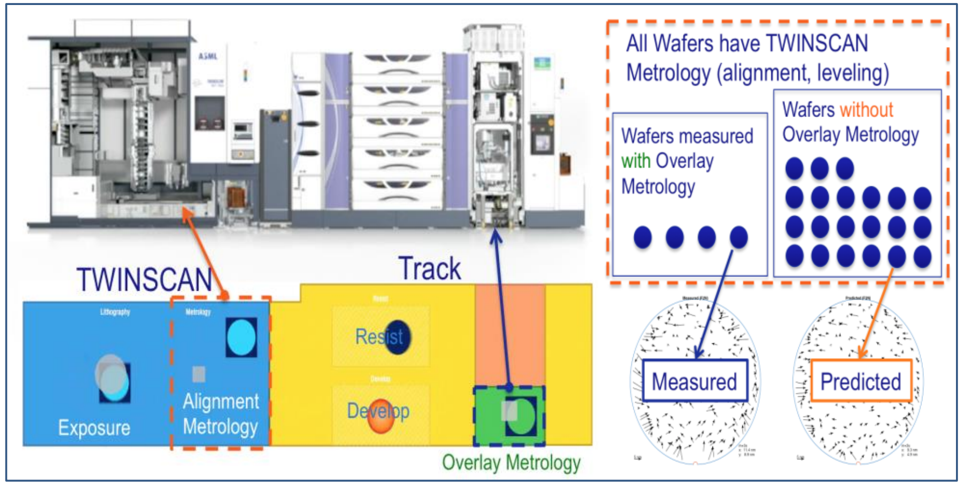
Modeling and Simulation



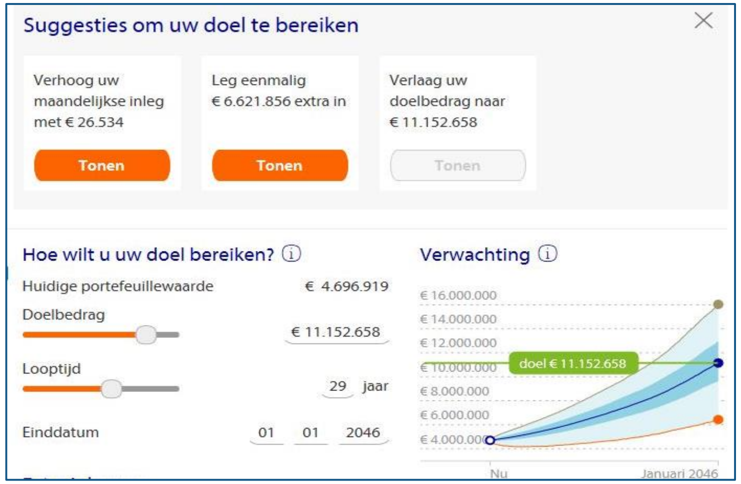
Testing and Verification

Sharing and Collaboration

Performance


All Wafers have TWINSCAN Metrology (alignment, leveling)
Wafers measured with Overlay Metrology
Wafers without Overlay Metrology
TWINSCAN
Track
Lithography
Metrology
Exposure
Alignment Metrology
Resist
Develop
Overlay Metrology
Measured
Predicted

ASML Develops Virtual Metrology Technology for Semiconductor Manufacturing with Machine Learning


Suggesties om uw doel te bereiken

Verhoog uw maandelijkse inleg met € 26.534

Tonen

Leg eenmalig € 6.621.856 extra in

Tonen

Verlaag uw doelbedrag naar € 11.152.658

Tonen

Hoe wilt u uw doel bereiken?

Huidige portefeuillewaarde € 4.696.919

Doelbedrag € 11.152.658

Looptijd 29 jaar

Einddatum 01 01 2046

Verwachting

Rabobank Develops Goal Monitor to Optimize Portfolio of their Customers

Vintecc Develops PLC System for Multi-Axle Harvesting Machine Using Model-Based Design

Punch Powertrain Develops Faster and more Accurate Control for Switched Reluctance Motors using Zynq SoC

InMotion Student Team Develops the Racing Car of the Future

MATLAB EXPO 2016