

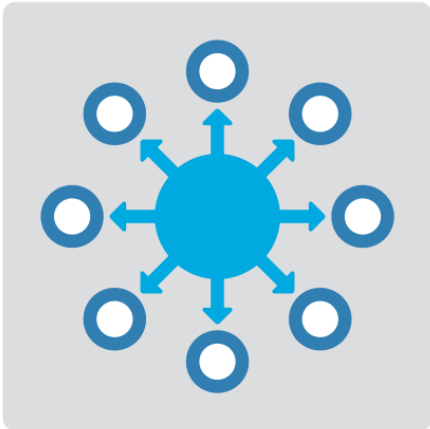
MATLAB EXPO 2017

What's New in MATLAB and Simulink

R2017a **R2016b**

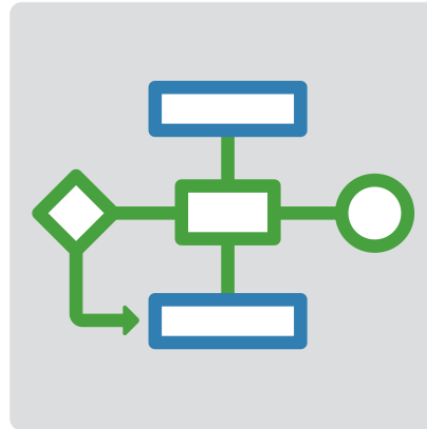
François Guérin and Daniel Martins, MathWorks

Application Breadth



**Products for the
work you do**

Workflow Depth



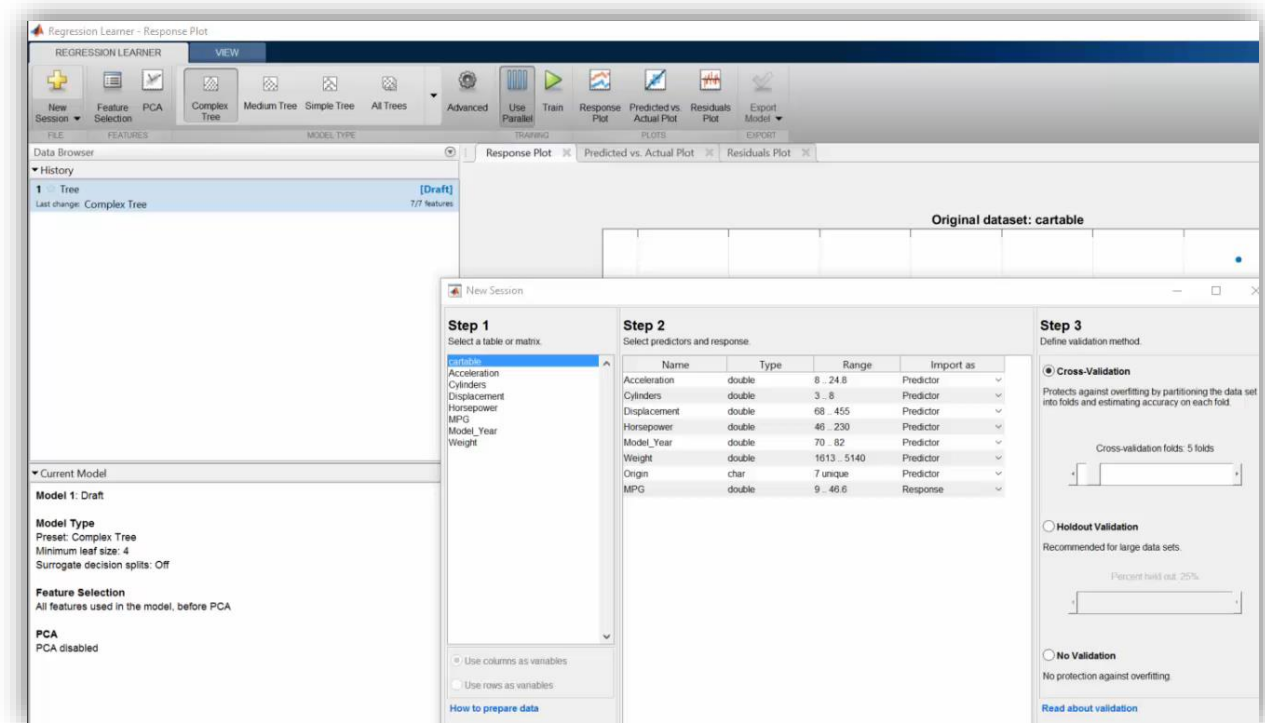
Platform Productivity



Machine Learning

“Learn” information directly from data without assuming a predetermined equation as a model

- Regression Learner app
 - Choose from multiple algorithms
 - Train and validate multiple models
 - Assess model performance, compare results, and choose the best model
- Code generation
 - Generate C code for predictive models that can be deployed directly to hardware devices

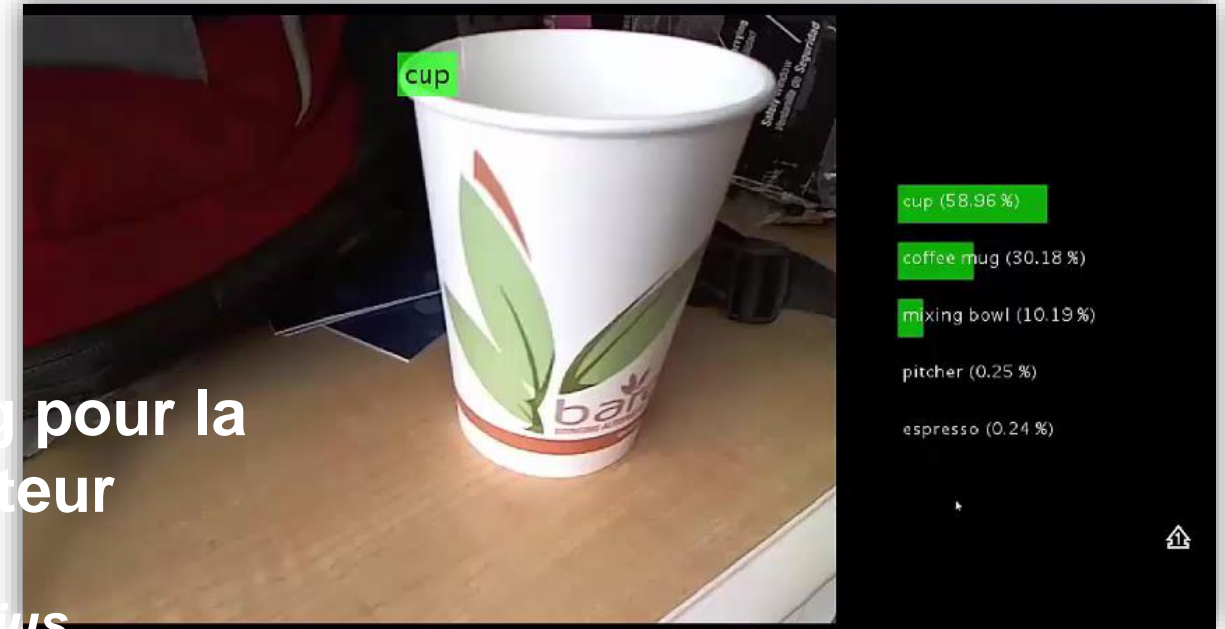


Deep Learning

R2016b R2017a

Apply deep learning to computer vision problems

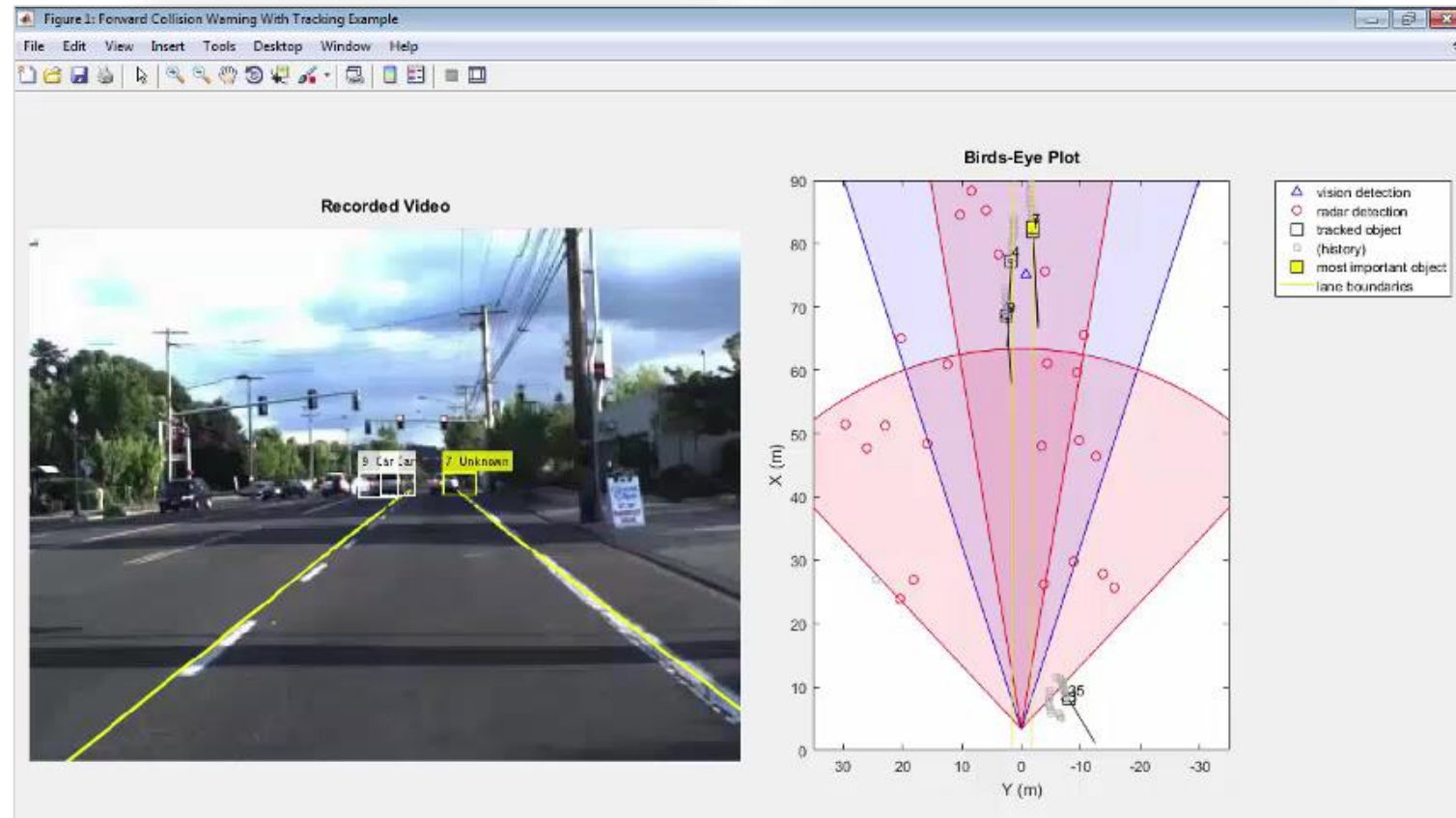
- Configure and train models using object detection algorithms (R-CNN, Fast R-CNN, Faster R-CNN)
- Leverage pretrained models for transfer learning (AlexNet, VGG-16, VGG-19)
- Import models from Caffe
- Train networks using multiple GPUs (including on Amazon EC2)



Autonomous Driving Systems

Design, simulate, and test ADAS and autonomous driving systems

- Algorithm development
 - Sensor Fusion
 - Computer Vision
 - Deep learning
- Visualization tools
- Testing and verification
 - Ground Truth Labeling App
 - Traffic scenario generation



More Modeling Domains Available

Faster Simulation

Model and simulation automotive powertrain systems

R2017a

- Model gasoline, diesel, hybrid, and electric systems
- Simulate engine subsystems, transmission assemblies, battery packs

Slower Simulation

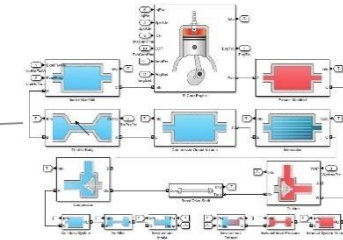
Design and test audio processing systems

R2016b

- Low-latency signal streaming
- Interactive parameter tuning
- Automatic generation of audio plugins

Powertrain Blockset
Audio System Toolbox

Powertrain Blockset



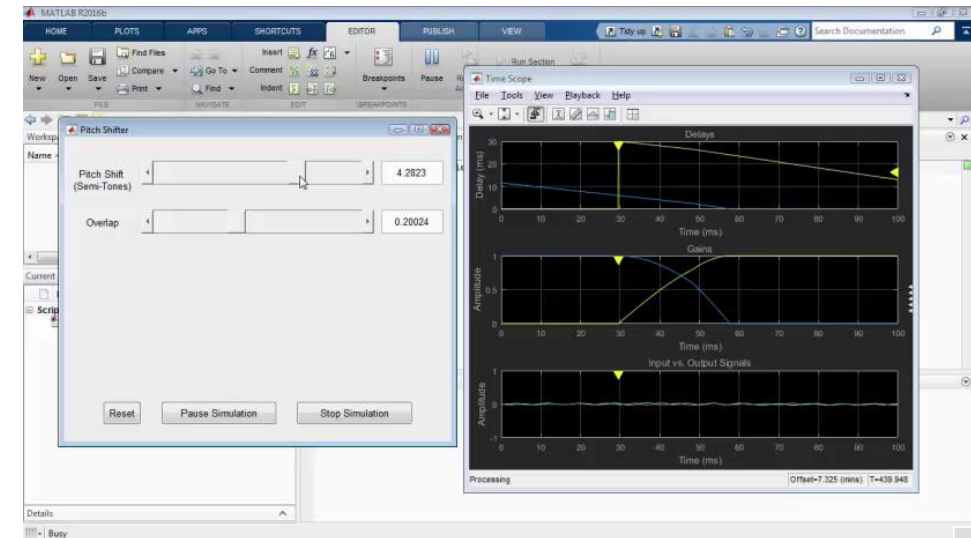
Simulink (ODE's)

Simscape (DAE's)

CAD Tools (PDE's)

Less Detail

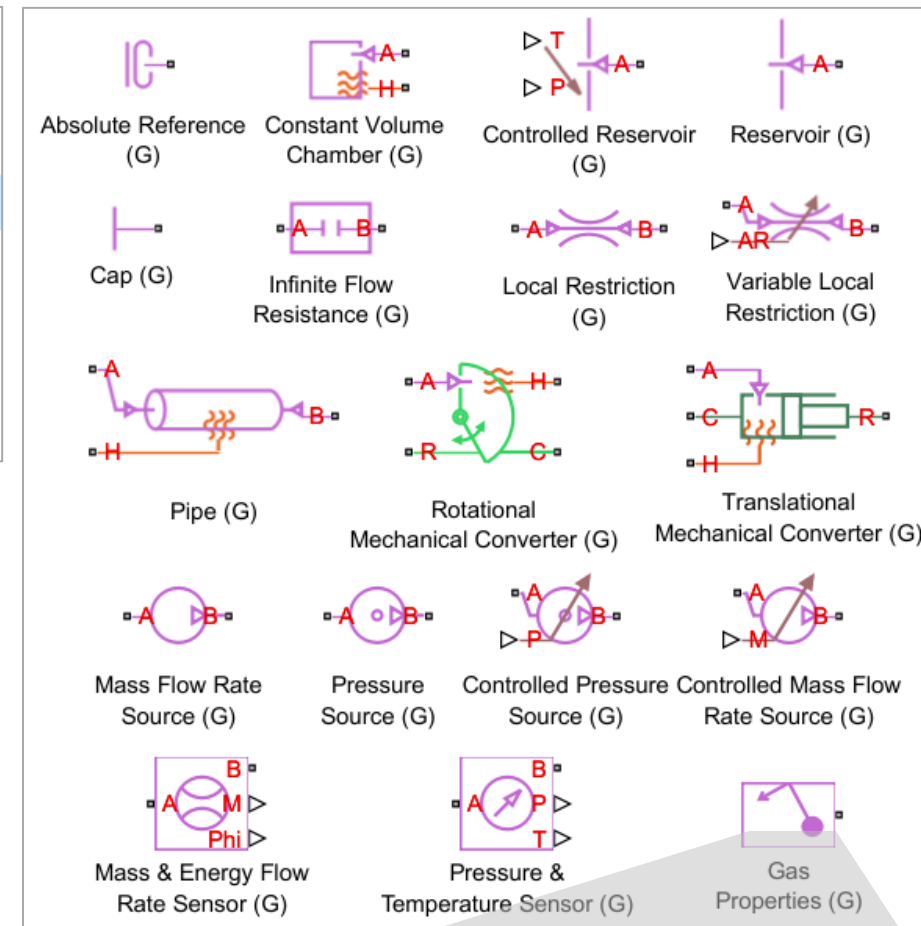
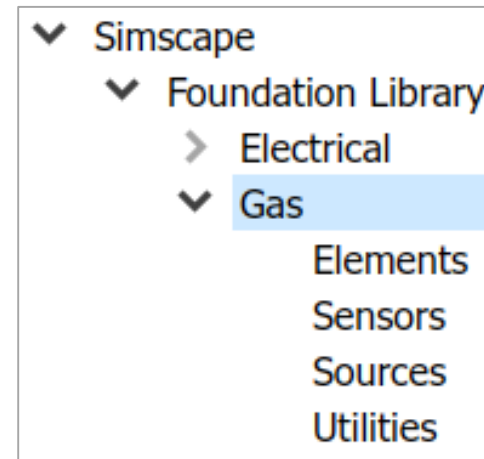
More Detail



Gas Domain and Block Library

Model gas systems with various levels of idealization

- Pneumatic actuation
- Gas transport in pipe networks
- Gas turbines for power generation
- Air cooling of thermal components
- Perfect gas, semiperfect gas, or real gas



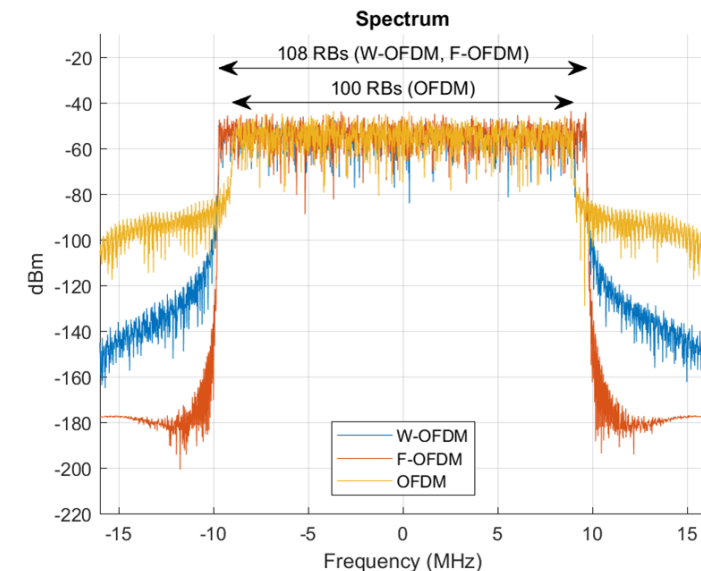
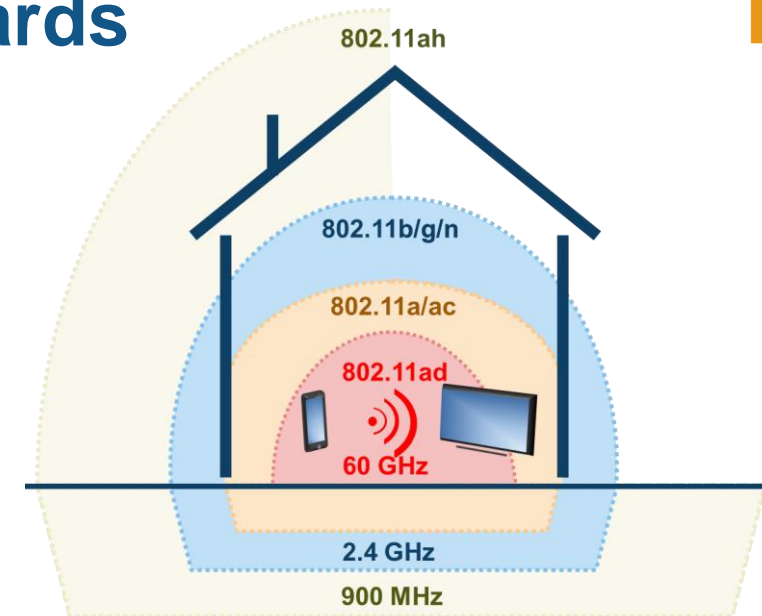
Gas specification:

- Perfect
- Semiperfect
- Real**

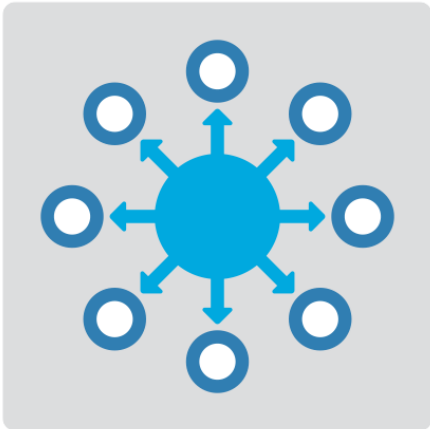
Support for the Latest Wireless Standards

Generate IEEE 802.11ad compliant waveforms and simulate 3GPP 5G radio technologies

- IEEE 802.11ad is a new Wi-Fi standard intended for high data rate short range communication
 - e.g., streaming video between a phone and a TV
- A new 5G library is available to explore the behavior and performance of new proposed 5G radio technologies

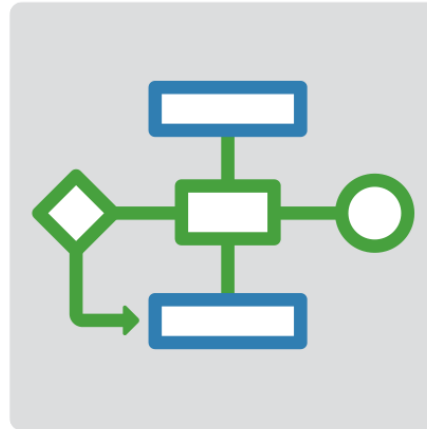


Application Breadth



**Products for the
work you do**

Workflow Depth



**Support for your
entire workflow**

Platform Productivity



Integrate MATLAB Analytics into Enterprise Applications

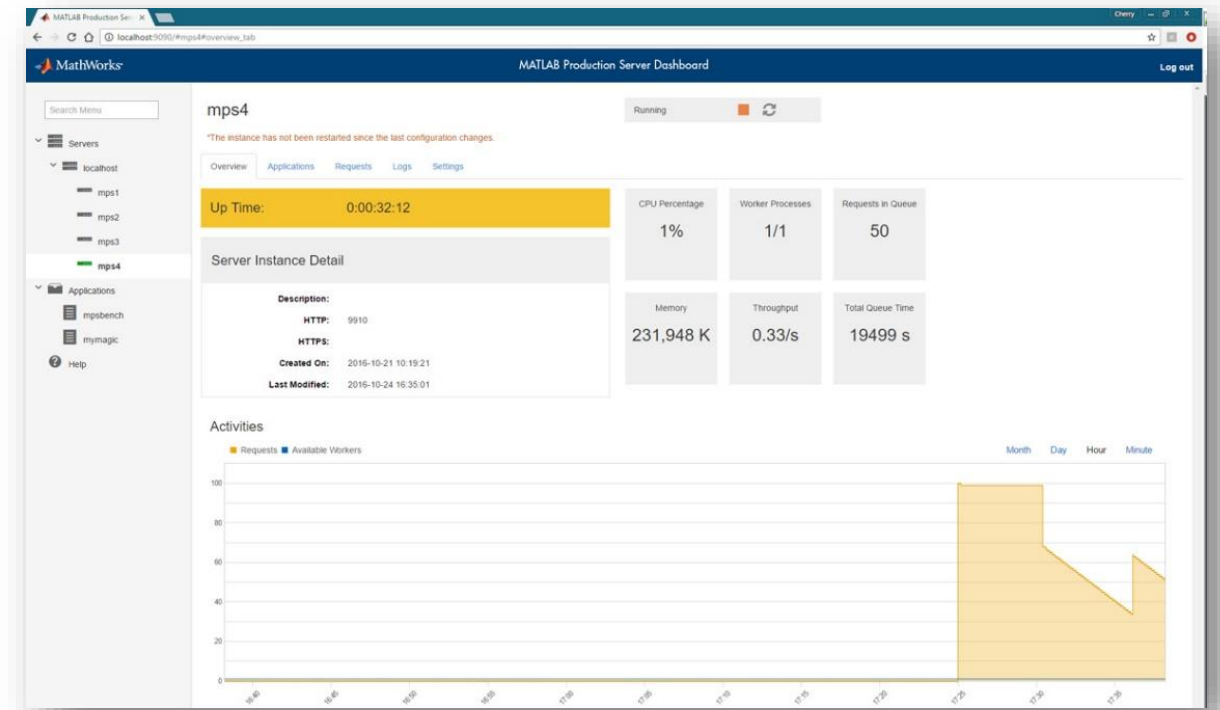
Deploy MATLAB algorithms without recoding or creating custom infrastructure

- Develop clients for MATLAB Production Server in any programming language that supports HTTP using RESTful API and JSON

R2016a

- Configure and manage multiple server instances using a web-based interface

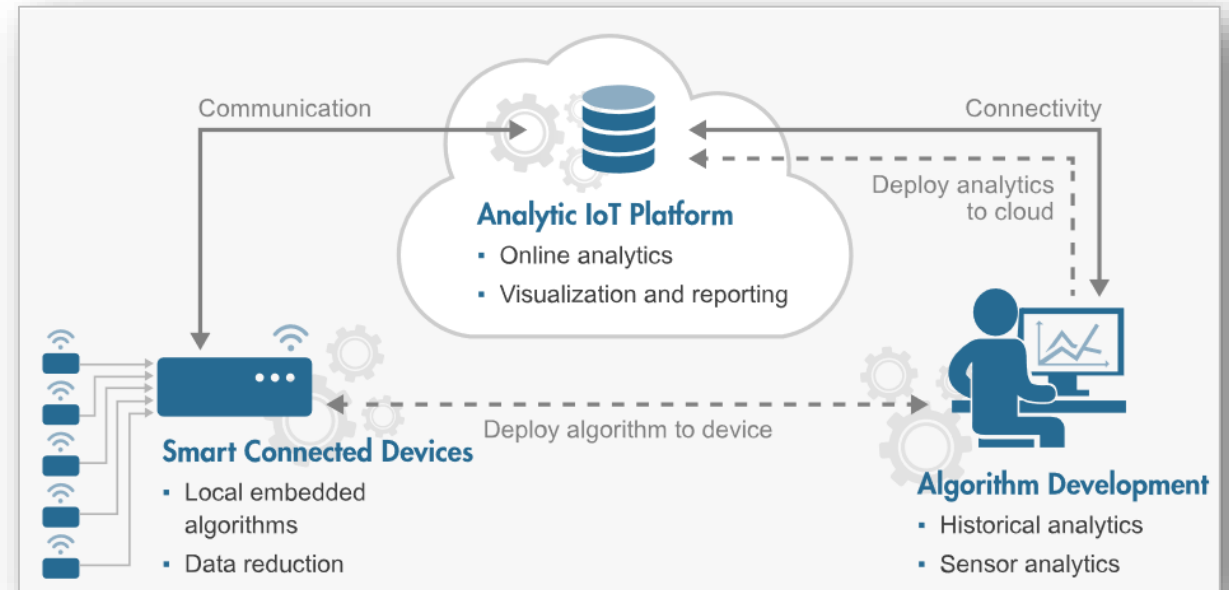
R2017a



Connecting MATLAB Analytics to IoT Systems

Develop analytics and deploy IoT systems

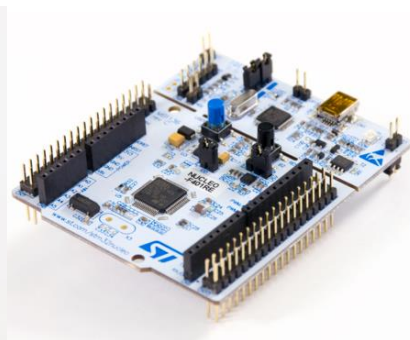
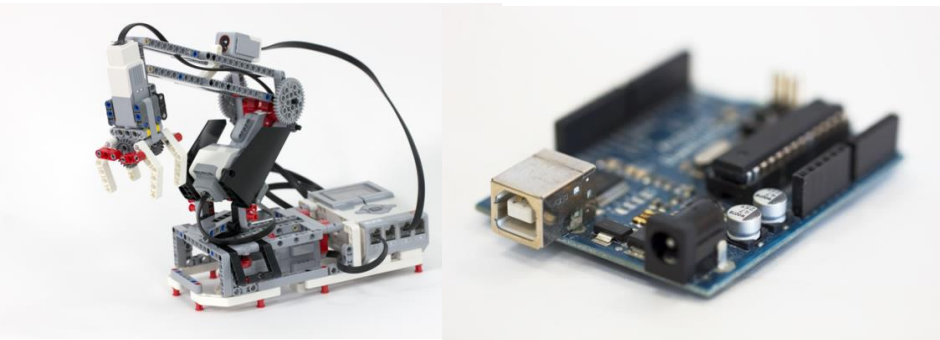
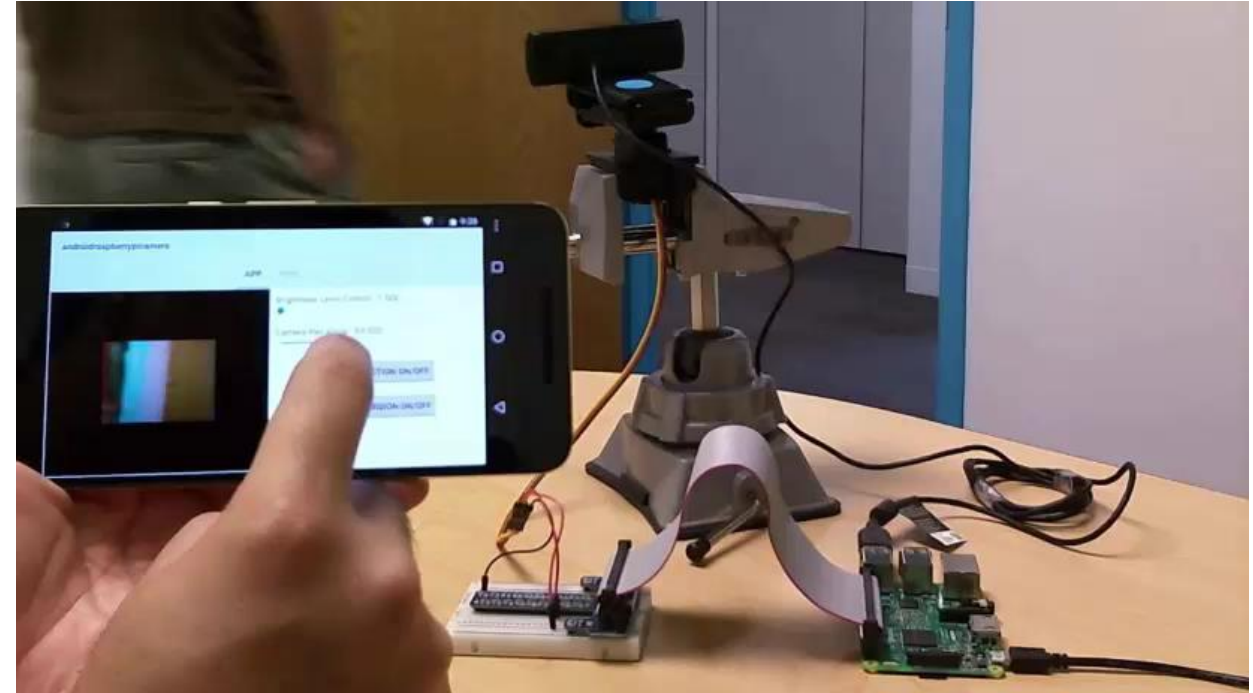
- Quickly collect and analyze IoT data with ThingSpeak and MATLAB
- Develop analytics algorithms using MATLAB and toolboxes
- Deploy on smart devices using code generation and embedded target support
- Deploy at scale on cloud using ThingSpeak and MATLAB Production Server



New Hardware Support

Run Simulink models on low-cost hardware devices

- Run Simulink models on Raspberry Pi 3 and Google Nexus devices
- Adds to existing hardware support, including LEGO, Arduino, iPhone, and Android devices



More Connections to 3rd Party Tools

Connect your models to Onshape and DOORS Next Generation

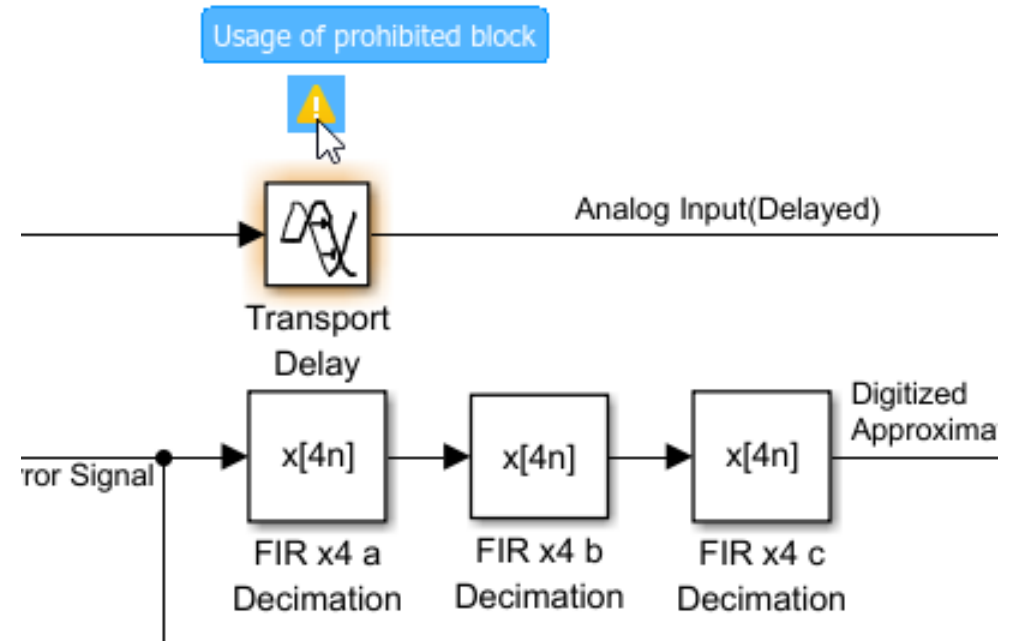
- Convert an Onshape CAD assembly into a Simscape Multibody model
- Link and trace model elements to requirements in DOORS Next Generation

The image displays a workflow for connecting CAD models to requirements. The top window shows the Onshape CAD interface with a robotic arm assembly. The bottom window shows the DOORS Next Generation interface with a requirements tree. A red circle highlights the 'Implemented By' section, which lists 'sf_car_DNG/Vehicle (SubSystem)'. A blue arrow points from this section to a context menu in the DOORS interface, where 'Link to Current Item in DNG' is highlighted with a red circle.

Complying with Safety-Critical Standards

Detect and fix standards compliance issues at design time with edit-time checking

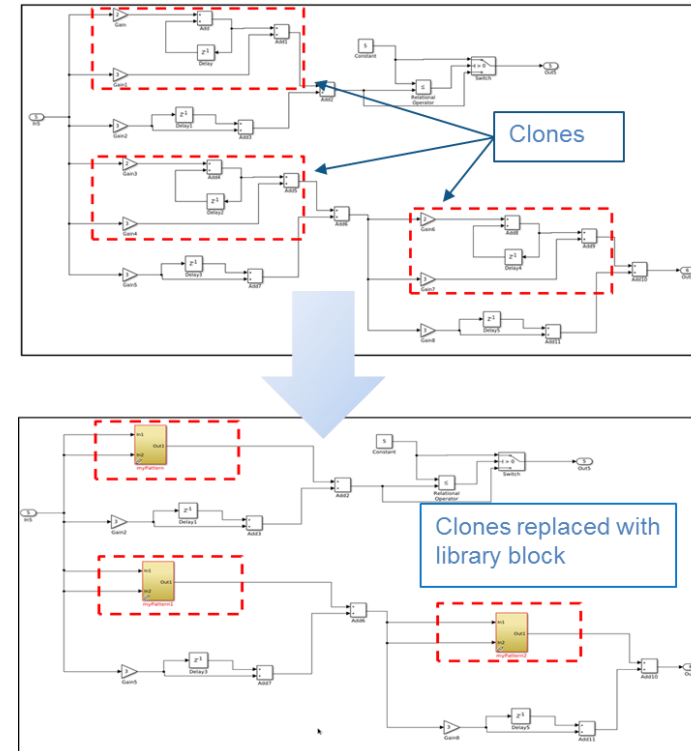
- Quickly address compliance and modeling standards issues before running the model
- For example, check for prohibited blocks or block names
- Especially useful for applications that require compliance to standards such as DO-178, ISO 26262, IEC 62304...



Efficient Code Generation

Improve code quality with clone detection and dynamic memory allocation

- Refactor repeating library patterns and subsystem clones
 - Reduces redundancy
 - Improves reusability
- Generate C code that uses dynamic memory allocation from MATLAB Function blocks
 - Allocate memory as needed at runtime



```

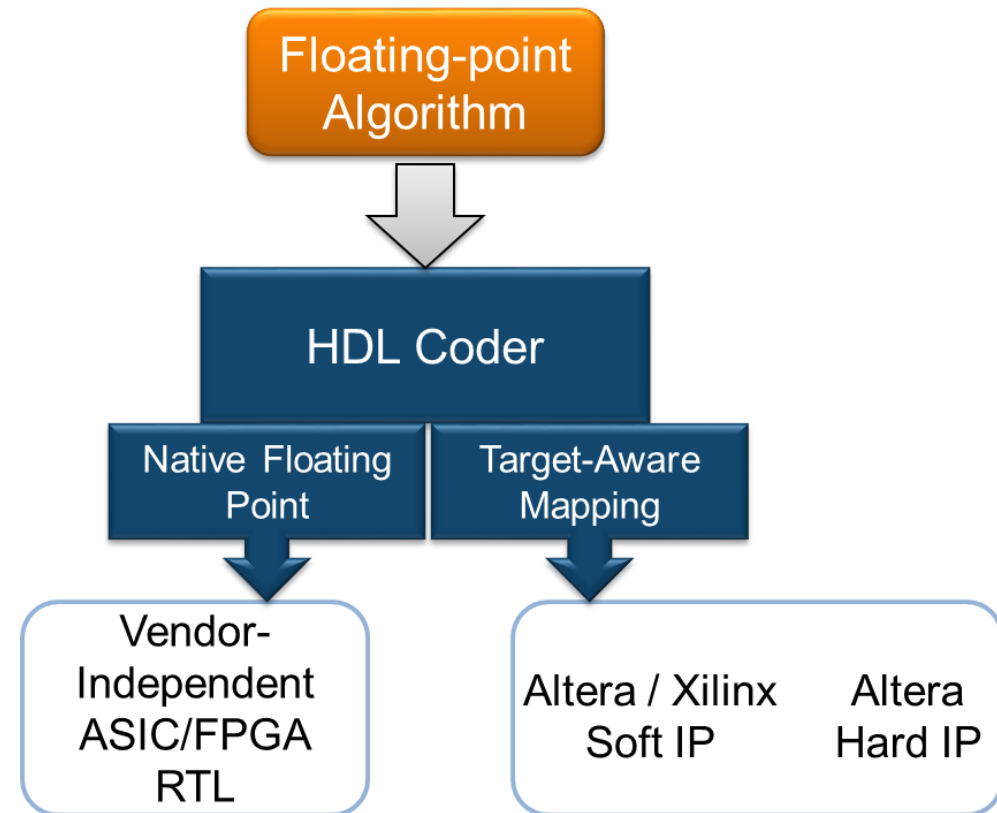
118  /* MATLAB Function: '<Root>/MATLAB Function' */
119  /* MATLAB Function 'MATLAB Function': '<S1>:1' */
120  if (!mymdl_DW.p_not_empty) {
121      /* '<S1>:1:4' */
122      /* '<S1>:1:5' */
123      k = mymdl_DW.p->size[0] * mymdl_DW.p->size[1];
124      mymdl_DW.p->size[0] = 1;
125      mymdl_DW.p->size[1] = 0;
126      mymdl_emxEnsureCapacity((emxArray_common_mymdl_T *)mymdl_DW.p, k, (int
127          sizeof(real_T));
128      mymdl_DW.p_not_empty = false;
129  }

```

Floating Point HDL Code Generation

Generate HDL code directly from single-precision floating point Simulink models

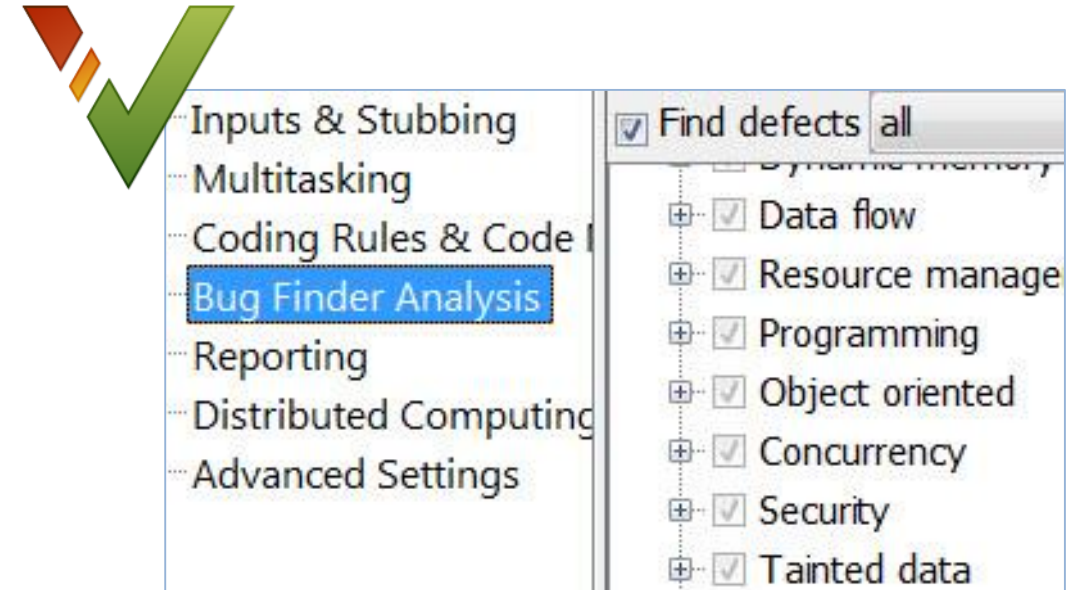
- No need to manually convert from floating point to fixed point
- Mix integer, fixed-point, and floating point operations to balance numerical accuracy versus hardware resource usage



Code Verification

Detect and prove the absence of run-time errors in your source code using static analysis

- Identify CERT C violations using defect checkers and coding rules
- Detect security vulnerabilities highlighted by the CERT C standard
- Addresses growing concern over software security with the rise in system connectivity



```

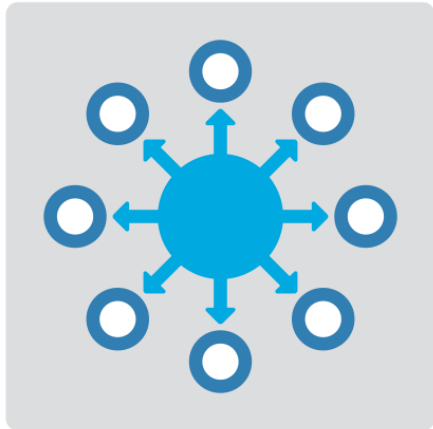
if (output v7 >= 0) {
    saved_values[output v7] = s8_ret;
    return s8_ret;
}
return reset_temp;

```

Assignment to element of static array (int 16): [-32 .. 112]
array size: 127
array index value: [0 .. 555]

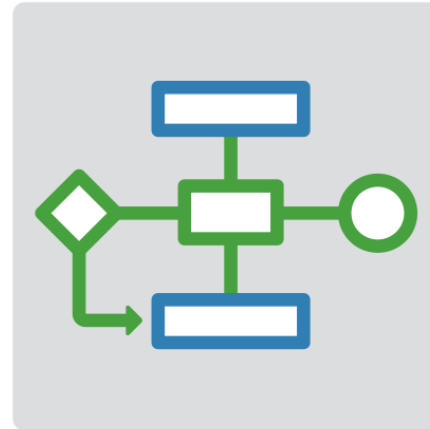
CERT C	Description	Polyspace Code Prover
ARR30-C	Do not form or use out-of-bounds pointers or array subscripts	Array access out of bounds

Application Breadth



**Products for the
work you do**

Workflow Depth



**Support for your
entire workflow**

Platform Productivity



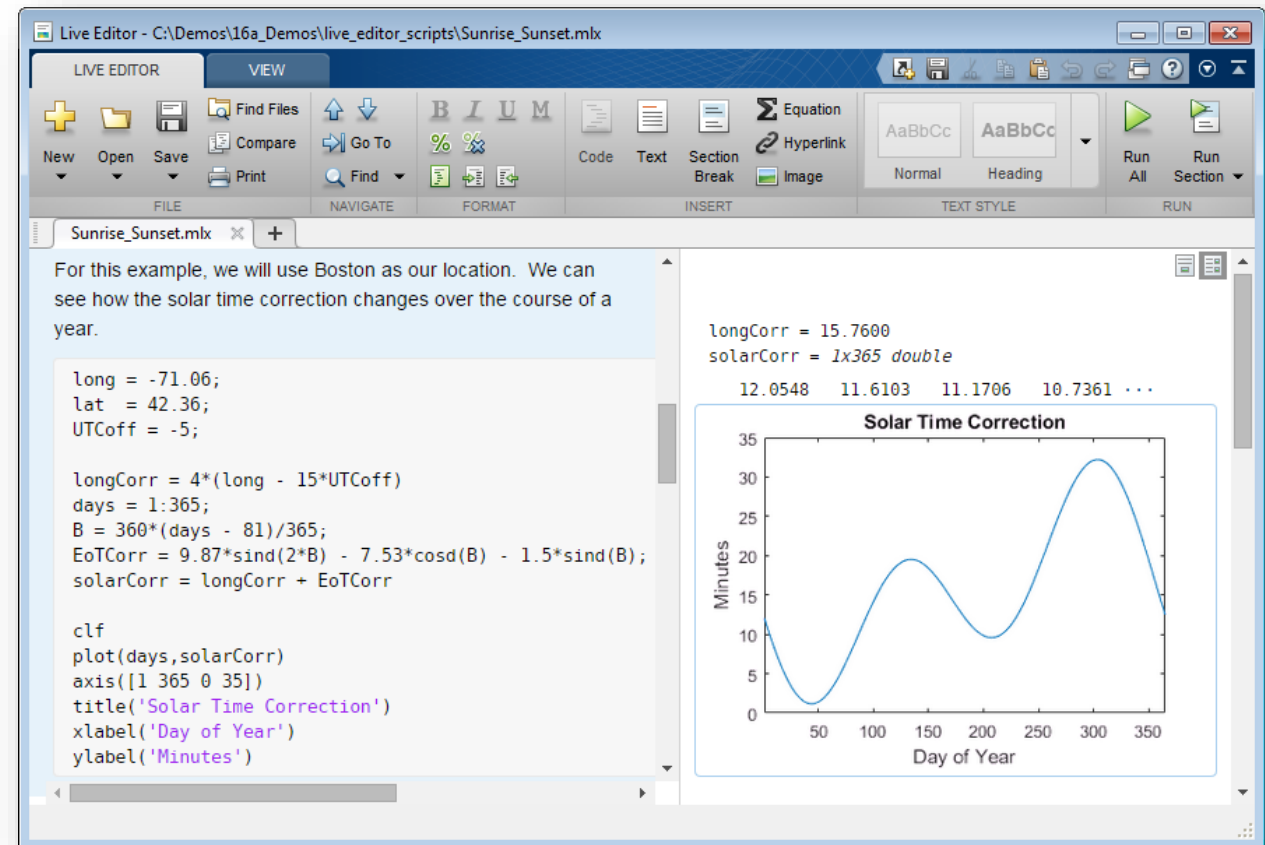
**Getting your work
done faster**

Change the Way You Work in MATLAB

R2016b R2017a

See results together with your MATLAB code in the Live Editor (introduced in R2016a)

- Add equations, images, hyperlinks, and formatted text
- Present, share, and collaborate using interactive documents
- Interactive figure updates
 - Pan, zoom, and rotate axes
 - Interactive plot customization, with MATLAB code generation to automate work
- Interactive equation editor

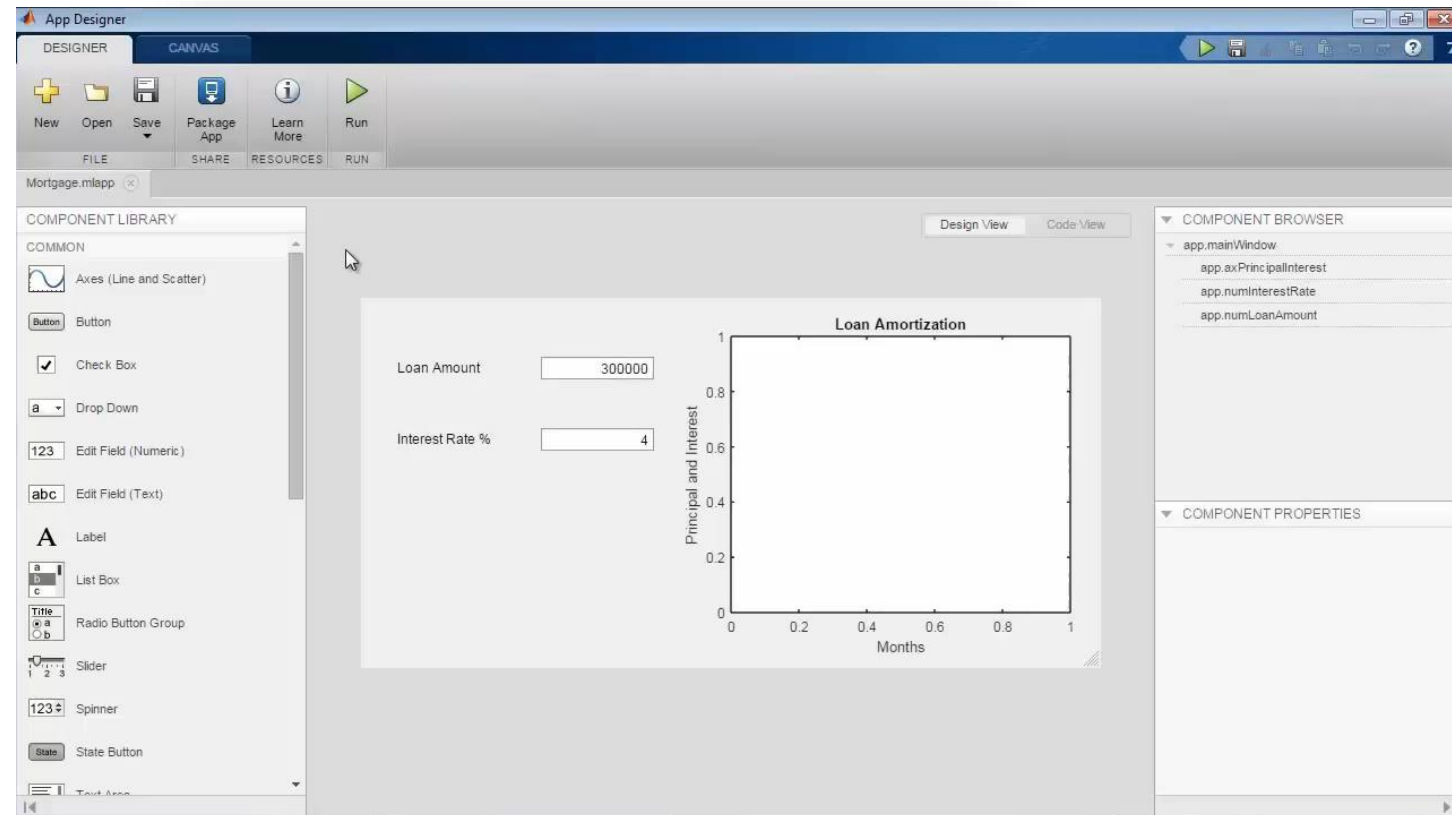


App Designer

R2016b R2017a

Environment for building MATLAB apps (introduced in R2016a)

- Full set of standard user interface components, as well as gauges, knobs, switches, and lamps
- Rich design environment for laying out apps
- Object-based code format for easily sharing data between parts of the app
- Enhancements include:
 - Majority of 2-D plots supported
 - Embed tabular displays using `uitable`
 - Zoom and pan plots in apps

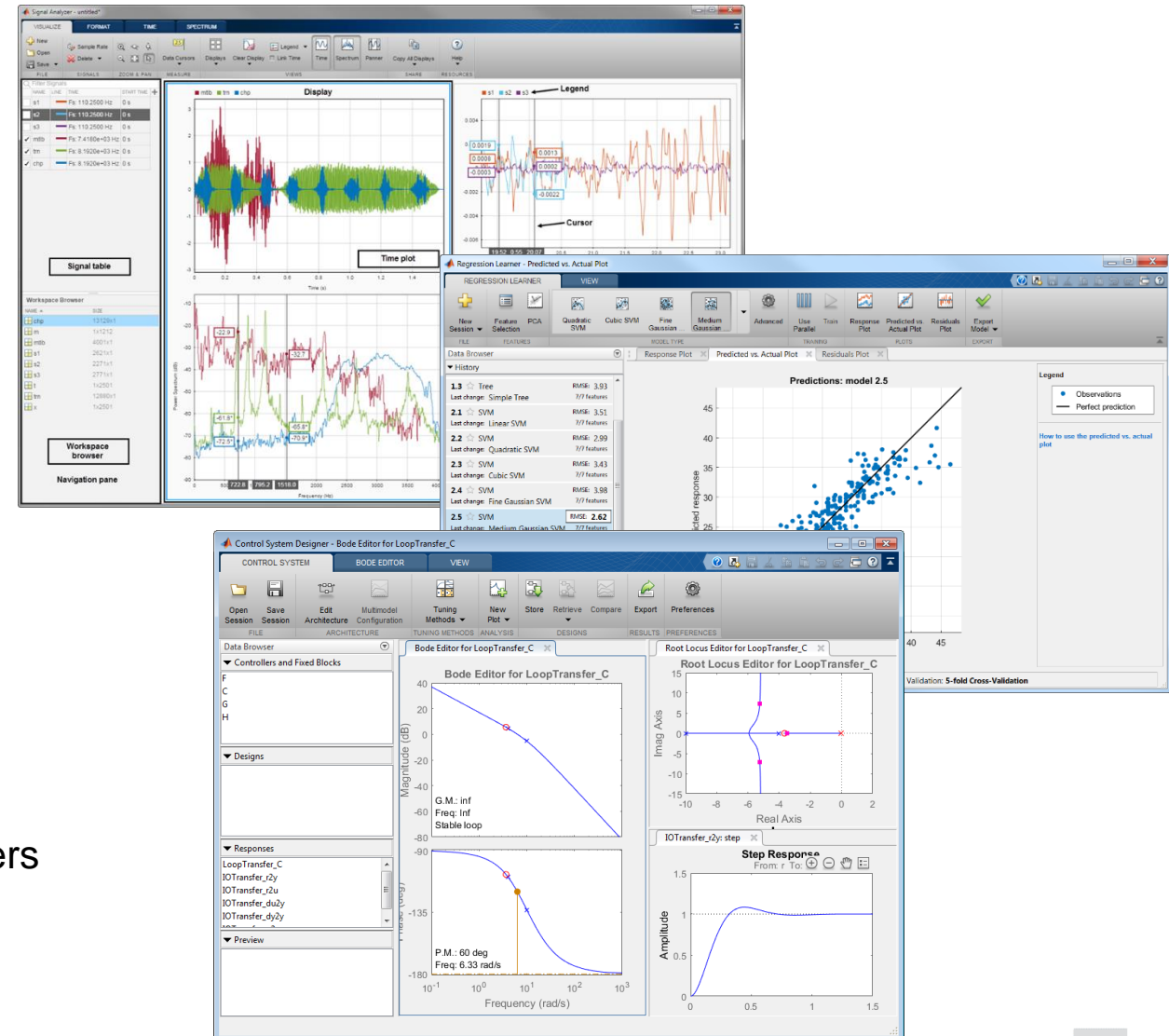


Apps Simplify Modeling and Simulation

R2016b R2017a

These interactive applications automate common technical computing tasks

- Signal Analyzer app
 - Perform time- and frequency-domain analysis of multiple time series
- Regression Learner app
 - Train regression models using supervised machine learning
- Control System Designer app
 - Design single-input, single-output (SISO) controllers



Signal Processing Toolbox

Statistics and Machine Learning Toolbox

Control System Toolbox

Working with Data Just Got Easier

R2016b R2017a

New data types and functionality for more efficient storage and managing of data

- **timetable** data container (introduced in R2016b)
 - Store time-stamped tabular data
 - Reorganize, evenly space, and align data
- **string** arrays (introduced in R2016b)
 - Memory efficient, faster string operations
 - New functions for common string manipulation
- New capabilities for **preprocessing** data
 - Find, fill, and remove missing data
 - Detect and replace outliers
 - Smooth noisy data

Time	Day	Total	Westbound	Eastbound
06/24/2015 00:00:00	Wednesday	13	9	4
06/24/2015 01:00:00	Wednesday	3	3	0
06/24/2015 02:00:00	Wednesday	1	1	0
06/24/2015 03:00:00	Wednesday	1	1	0
06/24/2015 04:00:00	Wednesday	1	1	0
06/24/2015 05:00:00	Wednesday	7	3	4

```

Command Window
>> s = [string('Square Circle Triangle'); string('Red Blue Green')]
s =
    2x1 string array
    "Square Circle Triangle"
    "Red Blue Green"
>>
>>
>> replace(s, 'Square', 'Rectangle')
ans =
    2x1 string array
    "Rectangle Circle Triangle"
    "Red Blue Green"
fx >>

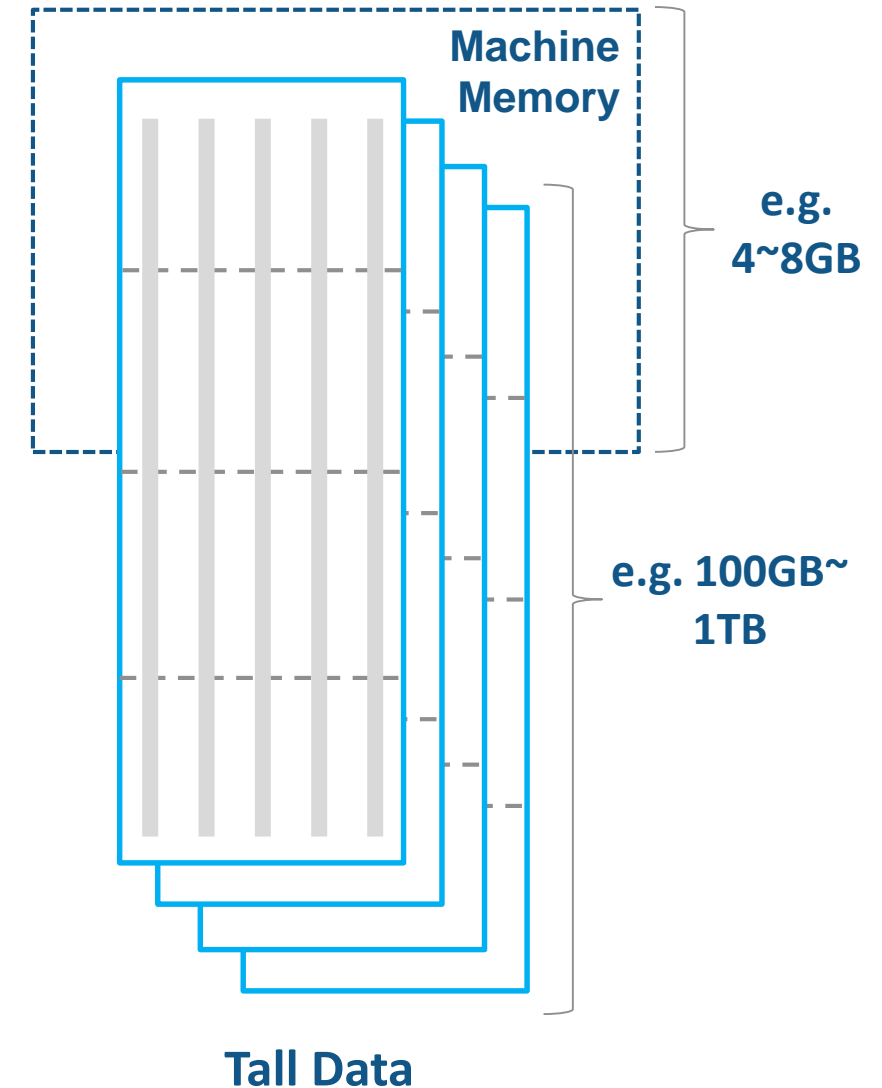
```

Working with Big Data Just Got Easier

R2016b **R2017a**

Use tall arrays to manipulate and analyze data that is too big to fit in memory

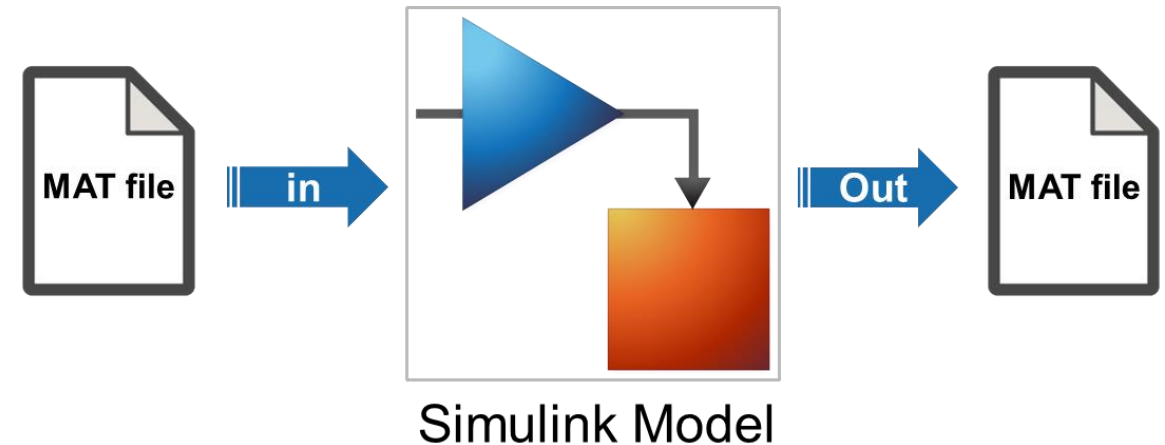
- Tall arrays let you use familiar MATLAB functions and syntax to work with big datasets, even if they don't fit in memory
- Support for hundreds of functions in MATLAB and Statistics and Machine Learning Toolbox
- Works with Spark + Hadoop Clusters



Working with Big Data Just Got Easier in Simulink Too

Stream large input signals from MAT-files without loading the data into memory

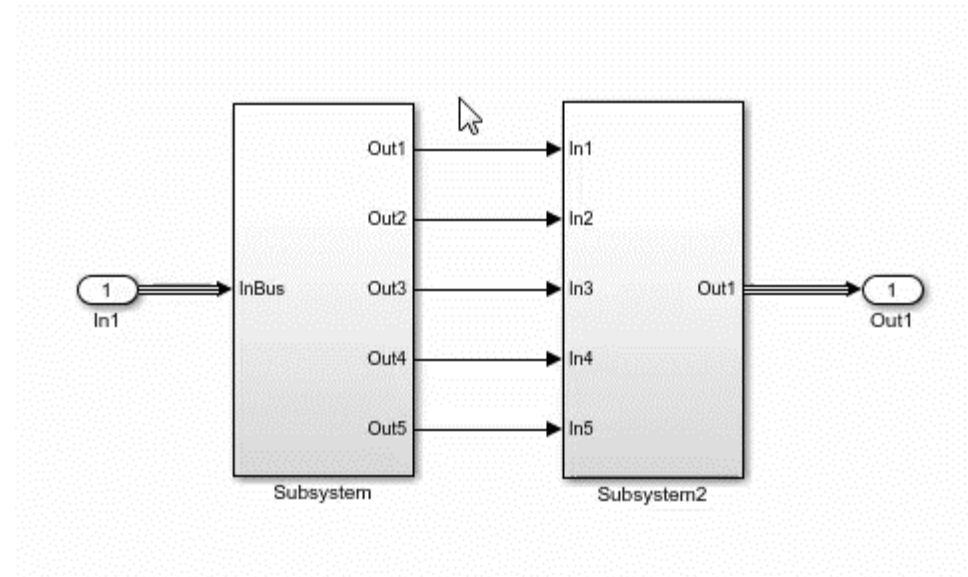
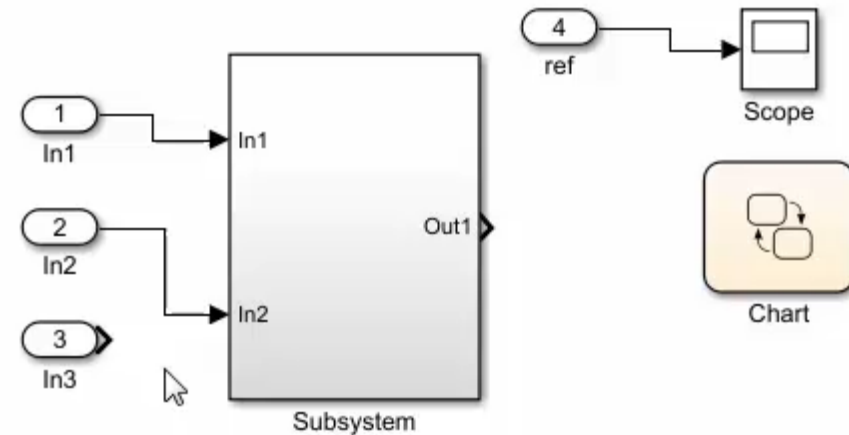
- Provides a big data workflow for Simulink simulations
- Use big data in Simulink logging and loading
- Especially useful when running many simulations where data retrieved is too large to fit into memory



Create Your Models Faster

Use automatic port creation and reduced bus wiring

- Add inports and outports to blocks when routing signals
- Quickly group signals as buses and automatically create bus element ports for fewer signal lines



Define your Data Faster

Reduces the need to open separate dialog boxes

- Model and block parameter data is now accessible within the main editor window
- Accessing and defining Stateflow data is also much easier

The screenshot displays the Simulink Stateflow editor for a fuel rate control system. The main workspace shows a Stateflow chart with several states and transitions. The 'Model Data Editor' window is open at the bottom, showing a table of data stores. The 'Property Inspector' and 'Symbol Manager' windows are also visible on the right side of the interface.

Block	Name	Test Point	Stream	Log Data	Path
Pressure Estimation					sidemo_fuelsys/fuel_rate_control/control_logic/Pressure_map_estimate
Throttle					sidemo_fuelsys/fuel_rate_control/control_logic/Pressure_map_estimate
Speed					sidemo_fuelsys/fuel_rate_control/control_logic/Pressure_map_estimate
Throttle Estimation					sidemo_fuelsys/fuel_rate_control/control_logic/Pressure_map_estimate
map					sidemo_fuelsys/fuel_rate_control/control_logic/Throttle_throttle_estimate
Speed					sidemo_fuelsys/fuel_rate_control/control_logic/Throttle_throttle_estimate

Property Inspector

Symbol Manager

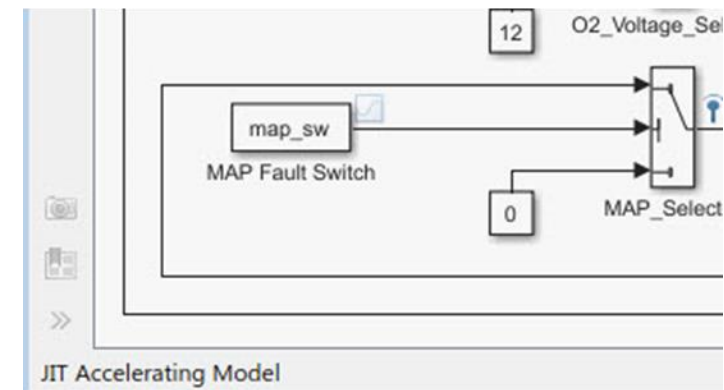
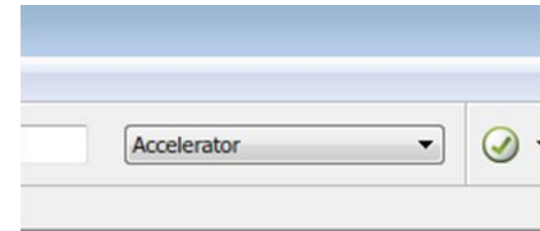
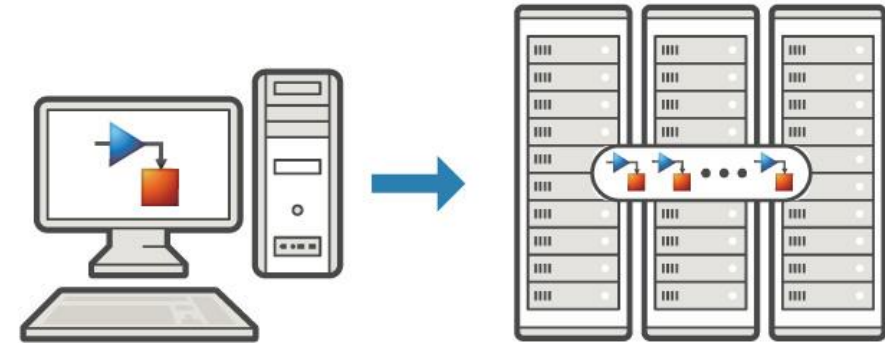
TYPE	NAME	PORT
	O2	
	Pressure	
	Throttle	
	Speed	
	Fail	
	es_o	1
	O2_normal	2
	max_ego	
	max_press	
	max_throt	
	hys	
	es_i	1
	fuel_mode	3

Model Data Editor

Simulate your Model Faster

Use the new `parsim` command and JIT acceleration to speed up your simulations

- Directly run multiple parallel simulations from the `parsim` command
- Quickly build the top-level model for improved performance when running simulations in Accelerator mode
- Especially use for Monte Carlo simulations and Design of Experiments

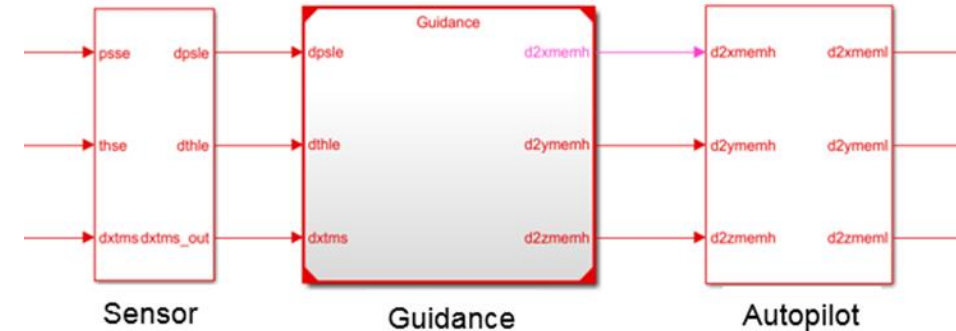


Cross-Release Code Integration

Reuse code generated from previous releases

- Reuse code that you generated from previous releases (R2010a and later)
- Avoid reverification cost due to the reuse of unmodified code

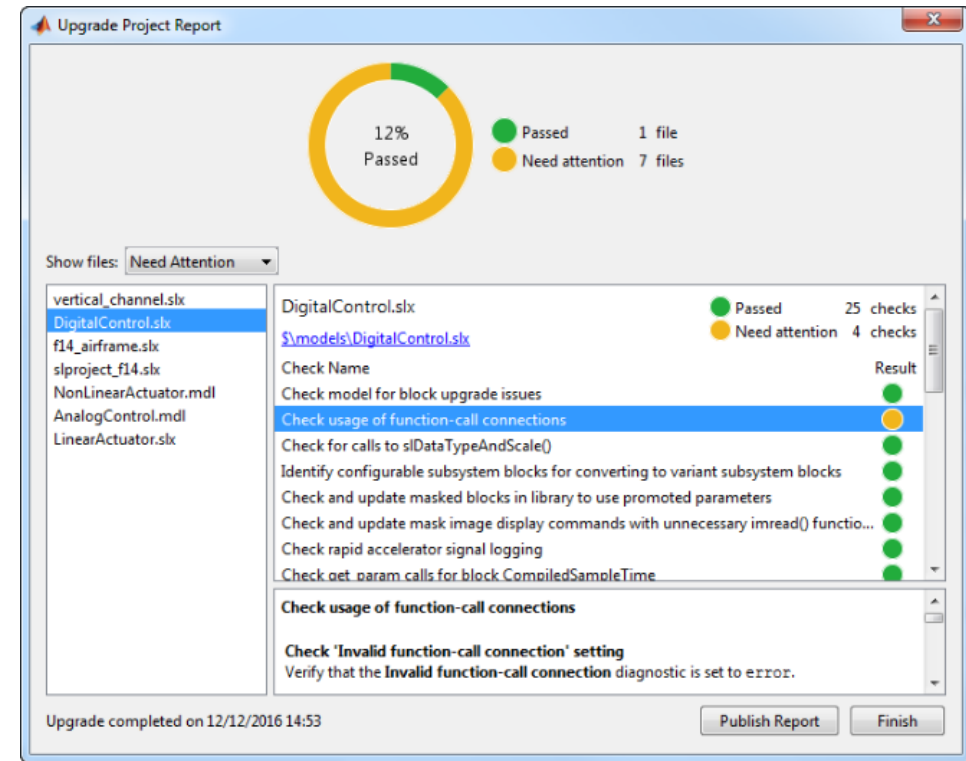
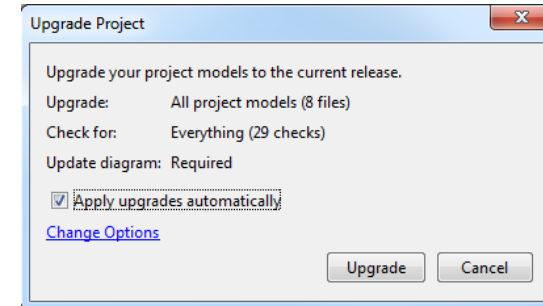
```
void AutonomousSystem_step(void)  
{  
  Sensor_SFcn( ... ) /* R2015b */  
  Guidance( ... ) /* R2016b */  
  Autopilot_SFcn( ... ) /* R2013a */  
}
```



Simulink Project Upgrade

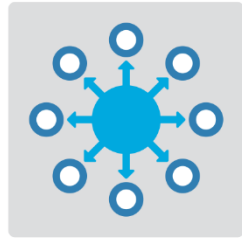
Easily update all the models in your Simulink Project to the latest release




- Avoid the manual process of upgrading one model at a time
- Simulink Project upgrade is an easy to use UI to automate the upgrade process of all the models in a Simulink project
- Fixes are automatically applied and a report gets generated



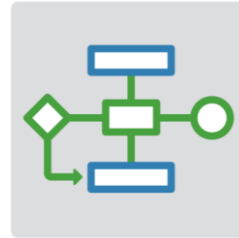
What's New in MATLAB and Simulink?

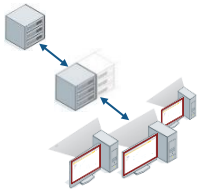


Application Breadth



- Machine learning 
- Deep learning
- Autonomous driving 
- New modeling domains
- New wireless standards 

Workflow Depth



- Enterprise applications 
- IoT systems
- 3rd party tool integration 
- Standards compliance
- Code generation and verification 

Platform Productivity



- Live Editor 
- MATLAB Apps 
- New (big) data types
- Modeling enhancements
- Release adoption 

Learn more and try R2017a ...

**Play
zone**

MATLAB EXPO 2017

Thank You

