

# MATLAB EXPO 2018

What's New in MATLAB  
and Simulink **R2017b** **R2018a**

Giuseppe Ridinò

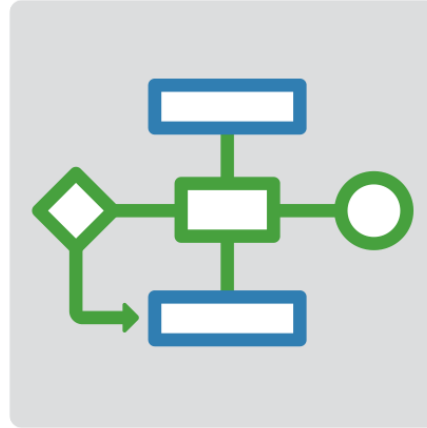


## Platform Productivity



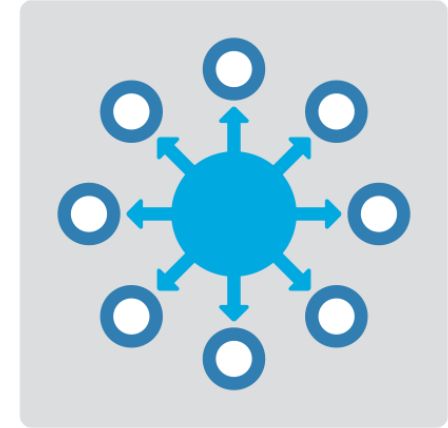
**Getting your work  
done faster**

## Workflow Depth



**Support for your  
entire workflow**

## Application Breadth

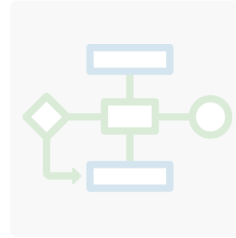


**Products for the  
work you do**

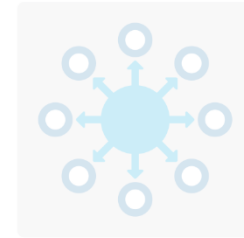
## Platform Productivity



## Workflow Depth



## Application Breadth



- **Create Your Designs Faster**
- **Simplify Analysis**
- **Simulate Faster and Scale Your Work**
- **Collaborate**

# Create Your Designs Faster

The screenshot shows the MATLAB Live Editor interface. The main window displays a script titled "Explore and Analyze Storm Events" with two sections: "Frequency of Events" and "Visualize with a Heatmap".

**Frequency of Events**  
 Explore the frequency of various storm events and locations and the associated damage costs.

```
clear
load prepEvents
data = timetable2table(data);
head(data)
```

**Visualize with a Heatmap**  
 This is helpful in exploring patterns across categories like the events and locations.

```
bigFigure;
heatmap(data, 'state', 'weathercats');
xlabel('State')
ylabel('Storm Event')
title('Frequency of Events by Location')
```

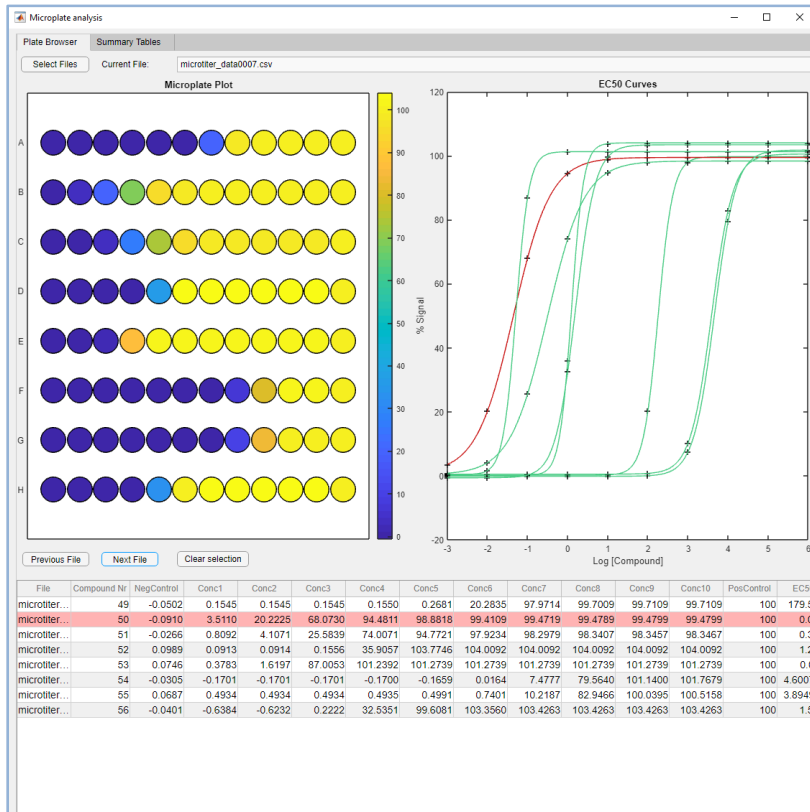
The output area shows the result of the `head` function: an 8x18 table with columns for index and Time.

	Time
1	22-Jul-2016
2	15-Jul-2016
3	15-Jul-2016
4	16-Jul-2016
5	15-Jul-2016
6	15-Jul-2016
7	15-Jul-2016
8	15-Jul-2016

Below the table is a heatmap visualization. The y-axis is labeled "Storm Event" and lists 20 categories: Avalanche, Blizzard, Coastal Weather, Dense Fog, Dense Fog, Drought, Dust Devil, Dust Storm, Extreme Heat, Flood, Freezing Fog, Frost/Freeze, Funnel Cloud, Hail, Heat, Heavy Rain, Hurricane, Ice Storm, Lightning, Sandstorm, Snow, Thunderstorm Wind, Tornado, Tropical Storm, and Waterspout. The x-axis represents the 18 data points from the table above. Blue squares indicate the presence of a storm event at a specific time.

**MATLAB**

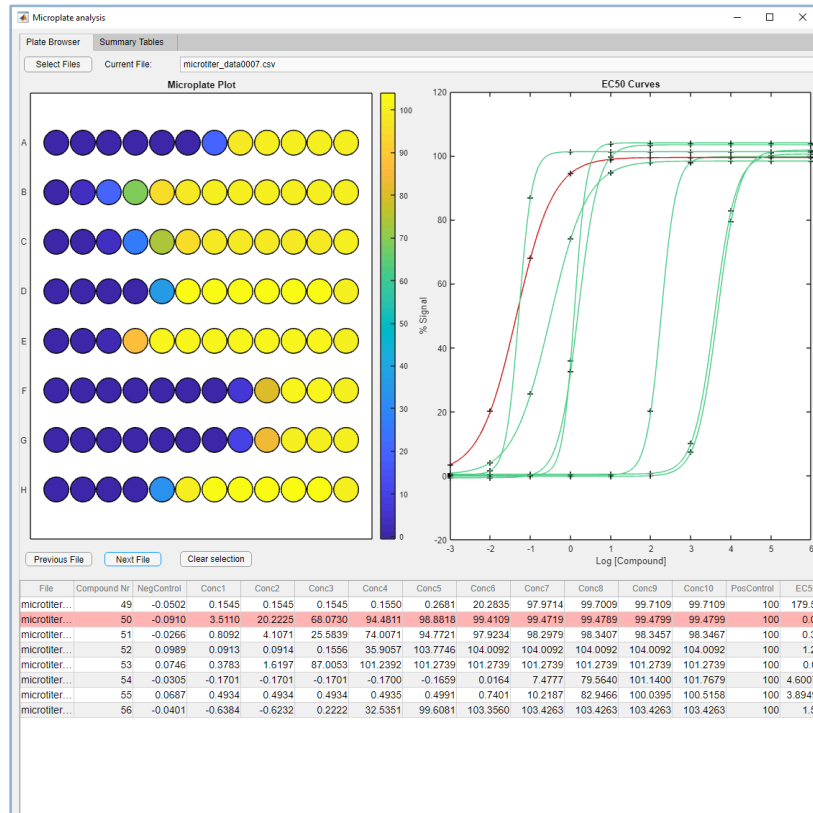
# Create Your Designs Faster



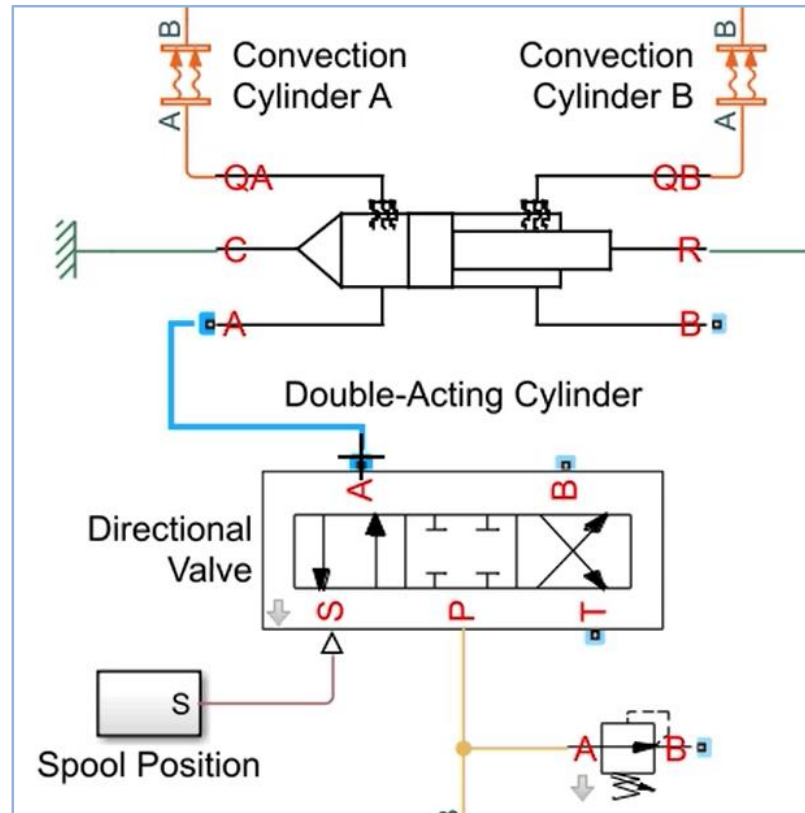
**MATLAB**

The screenshot shows the MathWorks File Exchange page for the 'GUIDE to App Designer Migration Tool for MATLAB'. The page title is 'GUIDE to App Designer Migration Tool for MATLAB' and it is version 1.0 (15.1 KB) by the MathWorks App Designer Team. The description states: 'Use the GUIDE to App Designer Migration tool to help transition your GUIDE apps to App Designer.'

# Create Your Designs Faster

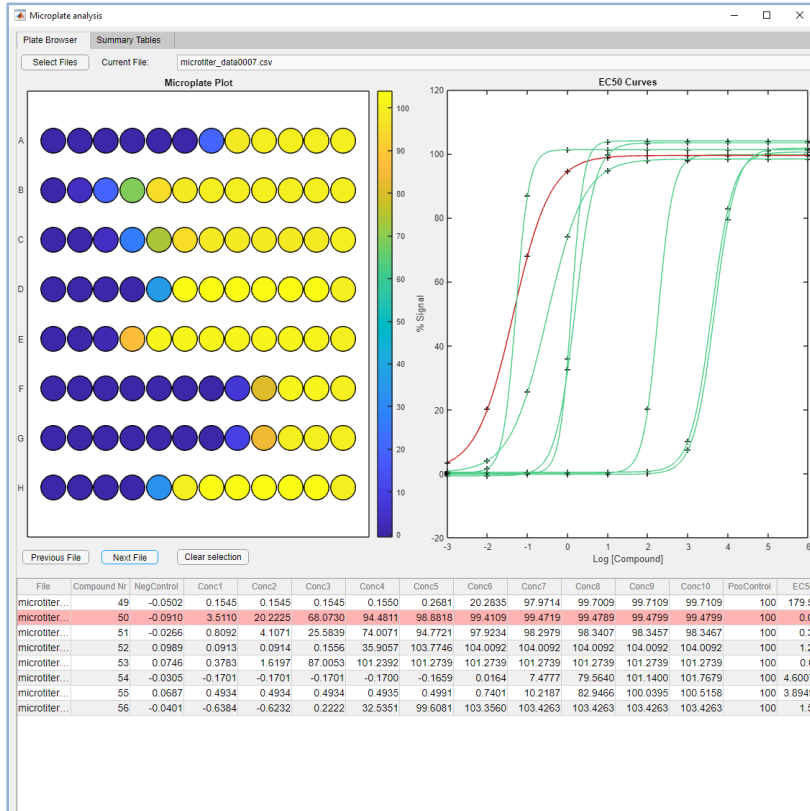


**MATLAB**

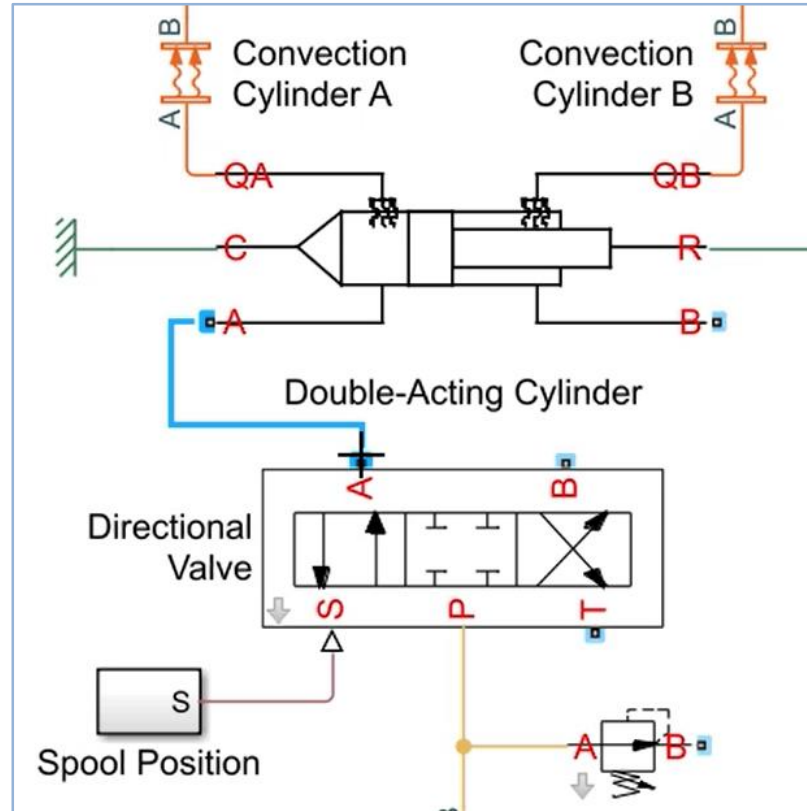


**Simulink (Simscape)**

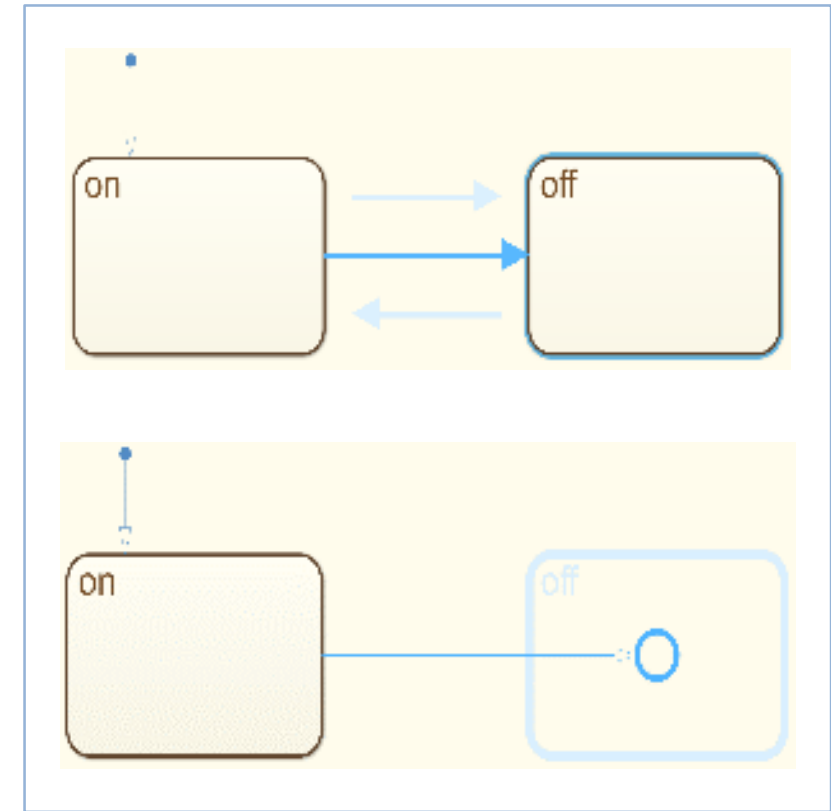
# Create Your Designs Faster



**MATLAB**



**Simulink (Simscape)**

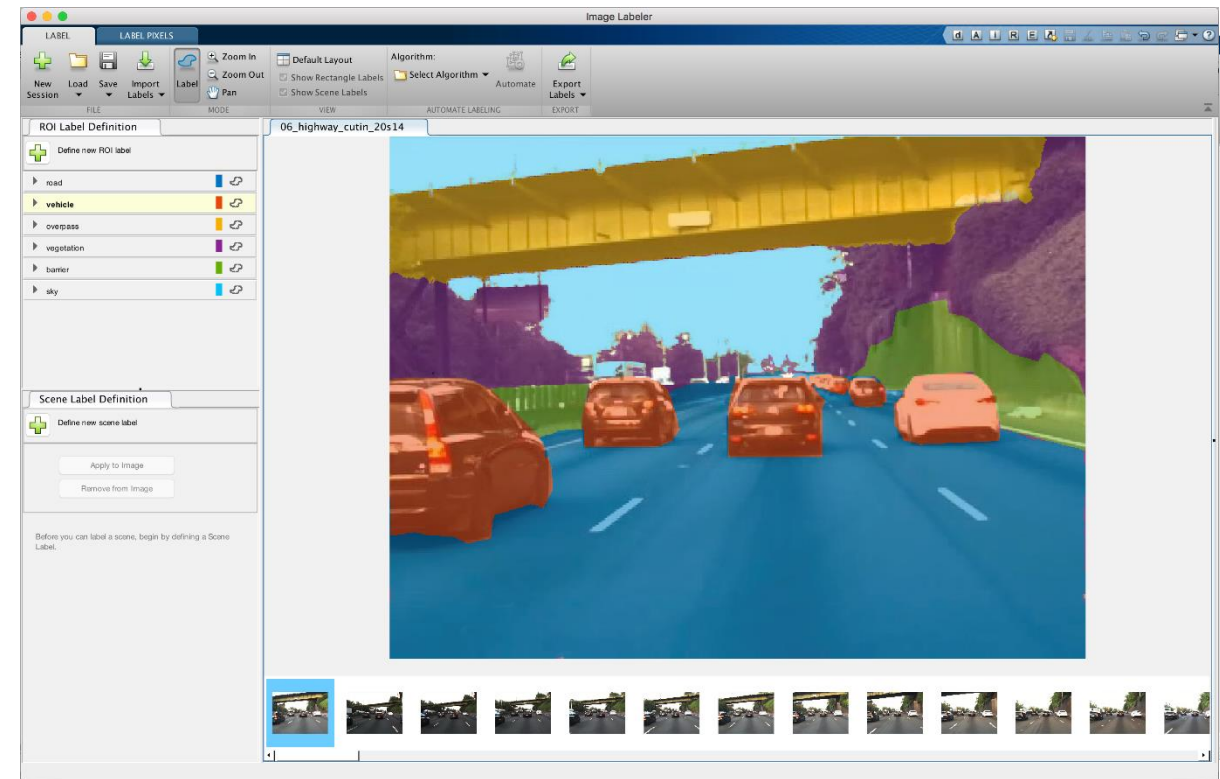


**Stateflow**

# Simplify Analysis with Apps

These interactive applications automate common technical computing tasks

- Econometric Modeler app
  - Perform time series analysis, specification testing, modeling, and diagnostics
- Analog Input Recorder app
  - Acquire and visualize analog input signals
- Image Labeling app
  - Label ground truth in a collection of images

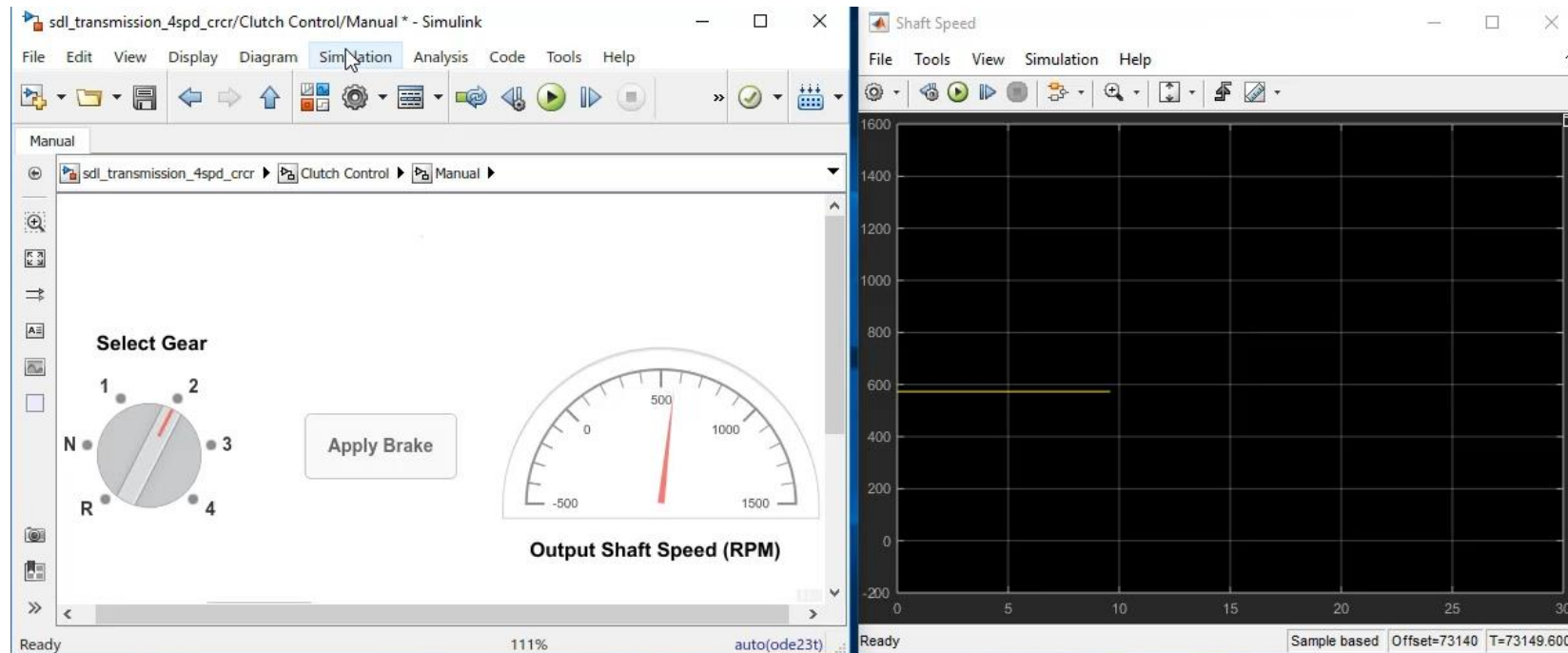




# Simplify Analysis by Simulating at Wall Clock Speed

**Slow down the simulation for easier model interactivity**

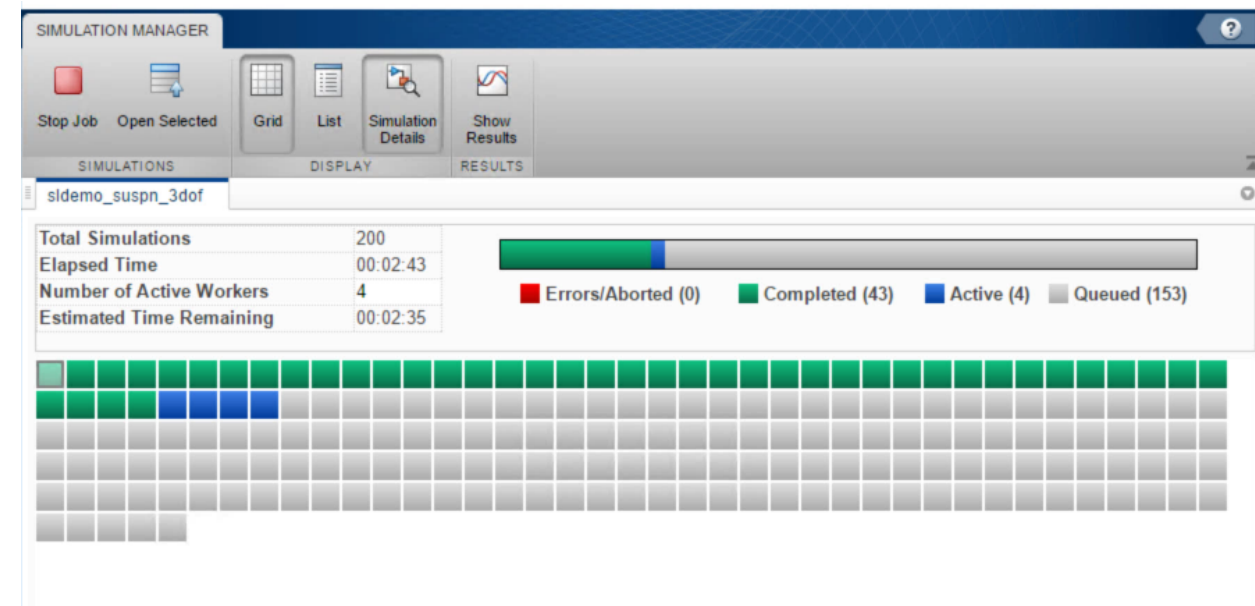
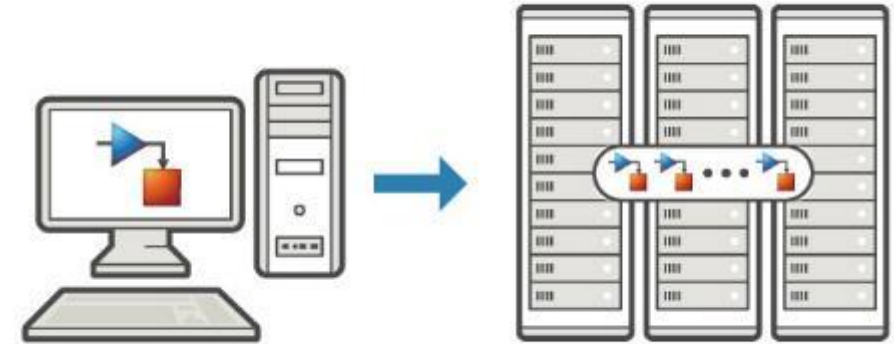
- Especially for models controlled and monitored via Dashboard blocks and other displays
- Useful when model is connected to hardware



# Scale Your Work

## Use parallel computing to run multiple simulations faster

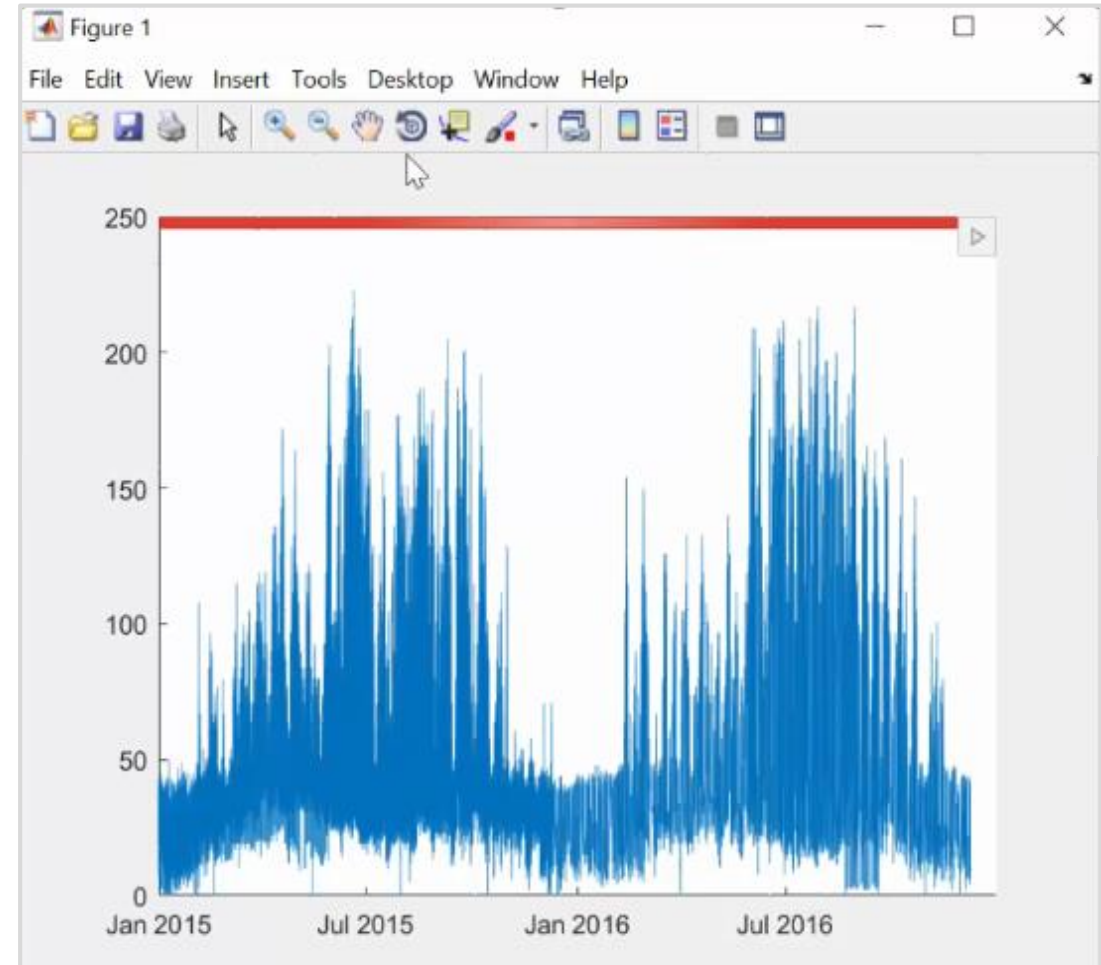
- Run multiple parallel simulations with `parsim`
- Monitor simulation status and progress in the Simulation Manager



# Scale Your Work

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

- Use familiar MATLAB functions and syntax
- Support for hundreds of functions
- Works with Spark + Hadoop clusters

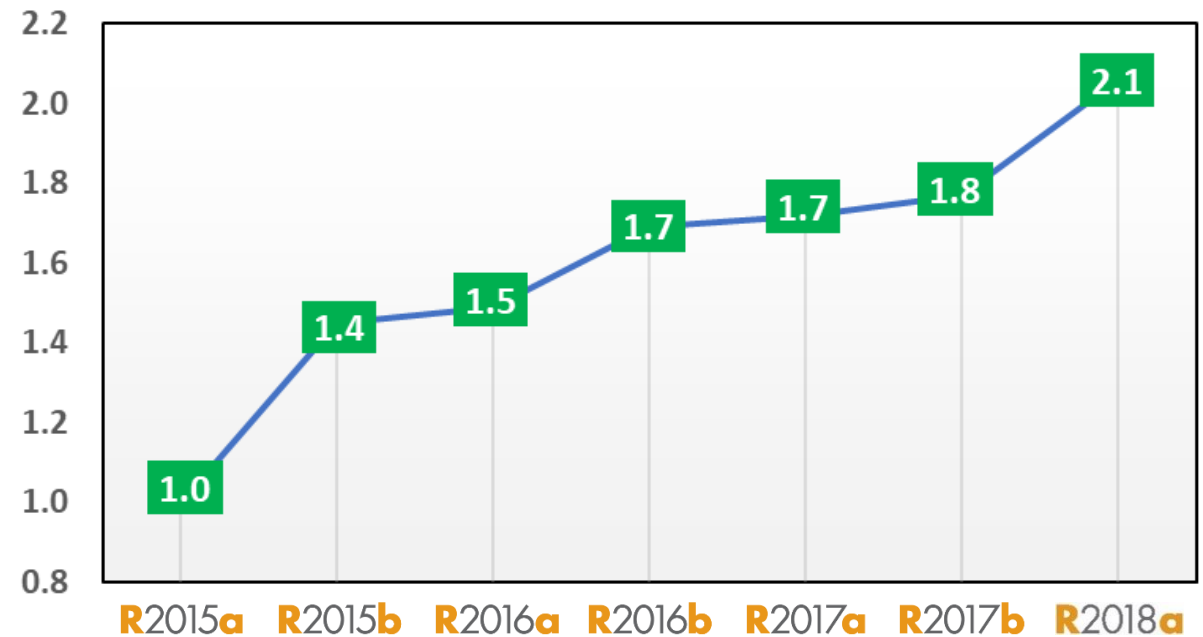


# Simulate Faster

## Redesigned execution engine runs MATLAB code faster

- All MATLAB code can now be JIT compiled
- MATLAB runs your code over twice as fast as it did just three years ago
- No need to change a single line of your code
- Increased speed of MATLAB startup in R2018a

Average Speedup in Customer Workflows



# Team Collaboration

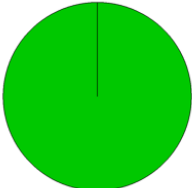
Use advanced software development features to manage, test, and integrate MATLAB code

## MATLAB® Test Report

Timestamp: 04-Jan-2017 13:28:06  
 Host: AH-SDE  
 Platform: win64  
 MATLAB Version: 9.1.0.441655 (R2016b)

Number of Tests: 17  
 Testing Time: 0.4516 seconds

Overall Result: PASSED



17 passed

### Overview

C:\Documents\MATLAB\OOP\Blip\Demos\Extensions\UnitTest\Class\

BlipTests.BlipSizeLengthTests	0.1403 seconds
BlipTests.BlipSubsasnTests	0.1542 seconds
BlipTests.BlipSubsrefTests	0.1572 seconds

### Details

C:\Documents\MATLAB\OOP\Blip\Demos\Extensions\UnitTest\Class\

BlipTests.BlipSizeLengthTests

- scalarBlipSize  
The test passed.  
Duration: 0.0863 seconds [\(Overview\)](#)
- vectorBlipSize  
The test passed.  
Duration: 0.0027 seconds [\(Overview\)](#)
- scalarBlipLength  
The test passed.  
Duration: 0.0044 seconds [\(Overview\)](#)

# Team Collaboration

Use advanced software development features to manage, test, and integrate MATLAB code

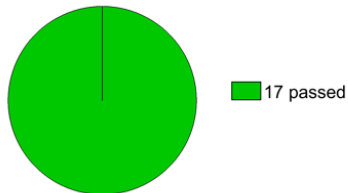
Identify differences between model elements, Stateflow charts, and MATLAB Function blocks

## MATLAB® Test Report

Timestamp: 04-Jan-2017 13:28:06  
 Host: AH-SDE  
 Platform: win64  
 MATLAB Version: 9.1.0.441655 (R2016b)

Number of Tests: 17  
 Testing Time: 0.4516 seconds

Overall Result: PASSED



**Overview**

Test Name	Duration
BlipTests.BlipSizeLengthTests	0.1403 seconds
BlipTests.BlipSubsasnTests	0.1542 seconds
BlipTests.BlipSubsrefTests	0.1572 seconds

**Details**

C:\Documents\MATLAB\OOP\Blip\Demos\Extensions\UnitTest\Class\

**BlipTests.BlipSizeLengthTests**

- scalarBlipSize**  
The test passed.  
Duration: 0.0863 seconds
- vectorBlipSize**  
The test passed.  
Duration: 0.0027 seconds
- scalarBlipLength**  
The test passed.  
Duration: 0.0044 seconds

Three-Way Merge - mine\_slproject\_f14.slx

MERGE

Previous Next Linked Scrolling Top Model Bottom Model Highlight Now Always Highlight in Models Filter Accept & Close

NAVIGATE HIGHLIGHT FILTER FINISH

Theirs: 340c64c37beb096a316e58a11358a6387d026b5f Base: e317566e2ad5f02f38f648e7d08716367a0fac Mine: mine\_slproject\_f14.slx

Simulink Pilot PilotGain Pilot:1 -> Bus Creator:1 PilotGain:1 Bus Creator:1

Model Configuration Sets Configuration Solver

Target: targetFile.slx

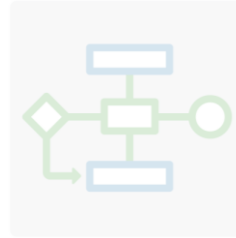
Resolve remaining 1 changes

TYPE	UNRESOLVED	RESOLVED
Conflict	1	0
Conflicted manual merge	0	0
Manual merge	0	0
Automatic	0	4
<b>Total</b>	<b>1</b>	<b>4</b>

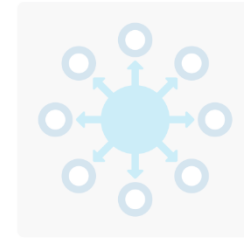
## Platform Productivity



## Workflow Depth



## Application Breadth

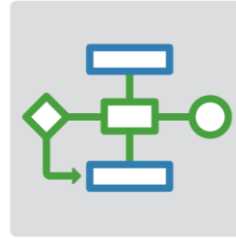


- **Create Your Designs Faster**
- **Simplify Analysis**
- **Simulate Faster and Scale Your Work**
- **Collaborate**

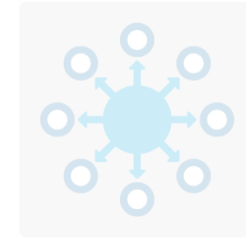
## Platform Productivity



## Workflow Depth



## Application Breadth



- **Deployment of MATLAB Algorithms and Applications**
- **Code Generation from Simulink Models**
- **Verification and Validation**



# Deploy MATLAB Algorithms and Applications

## Access Data



Sensors



Files



Databases

## Analyze Data



Data exploration



Preprocessing



Domain-specific algorithms

## Develop



AI model



Algorithm development



Modeling & simulation

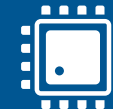
## Deploy



Desktop apps

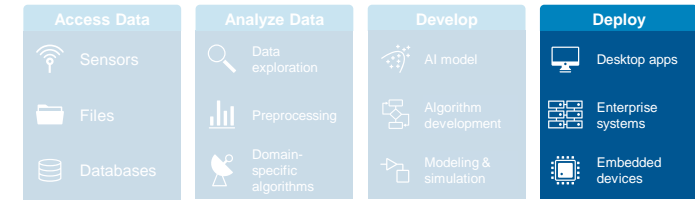


Enterprise systems



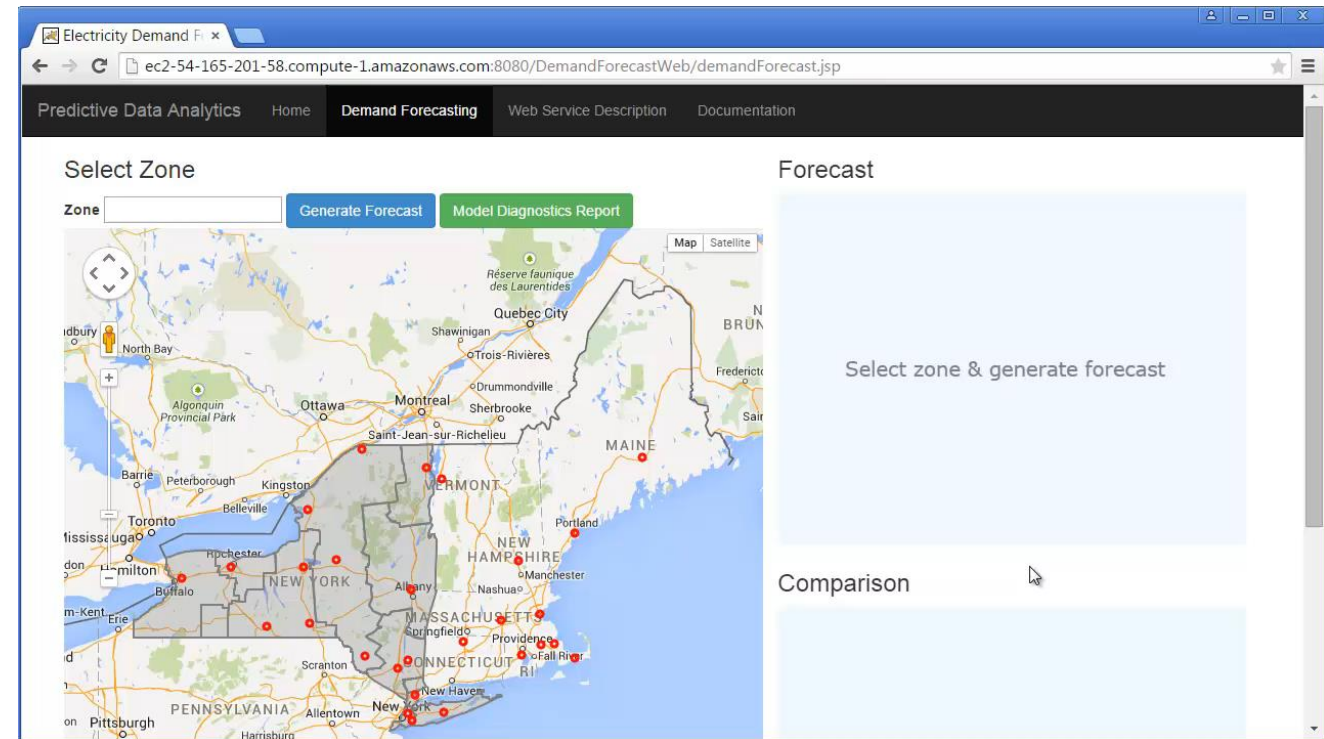
Embedded devices

# Deploy MATLAB Algorithms and Applications

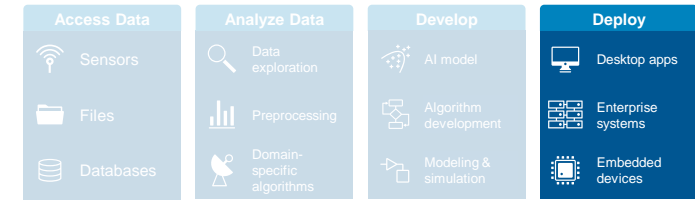


## Share your work outside of MATLAB without having to recode your algorithms

- Standalone desktop applications
- Add-ins for Microsoft Excel
- Software components to integrate with other languages (*C/C++*, *.NET*, *Python*, *Java*)
- Software components for web and enterprise applications

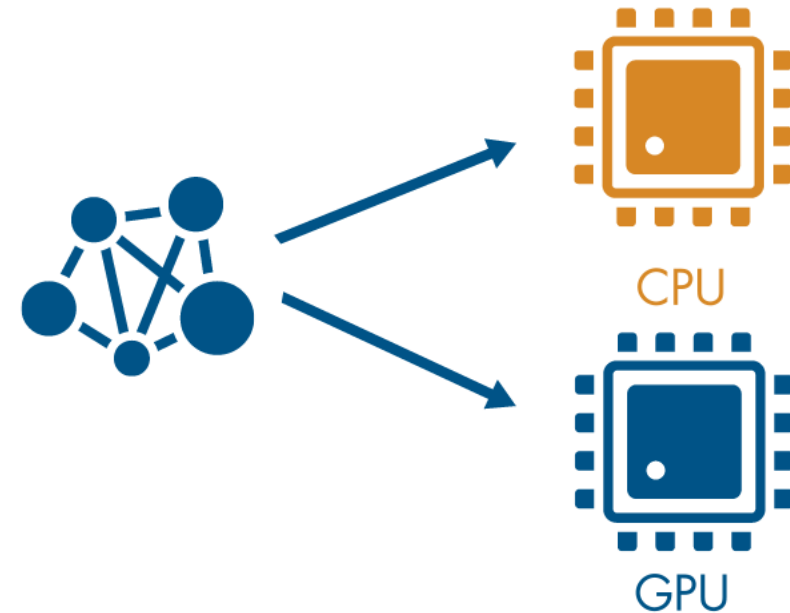


# Deploy MATLAB Algorithms



## Deploy machine learning and deep learning models using automatically generated code

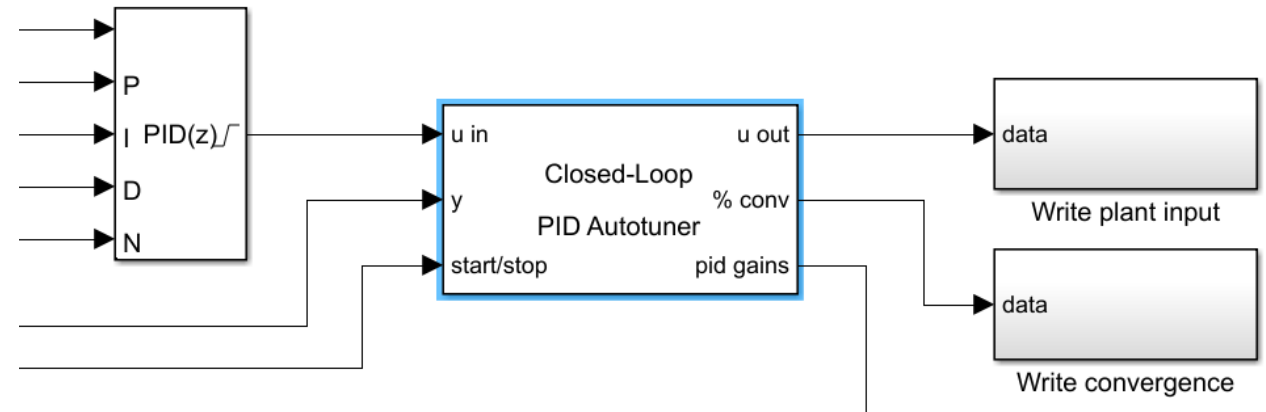
- Generate C code for predictive machine learning and deep learning models
- Generate optimized CUDA code for deep learning, embedded vision, and autonomous systems



# PID Control Tuning

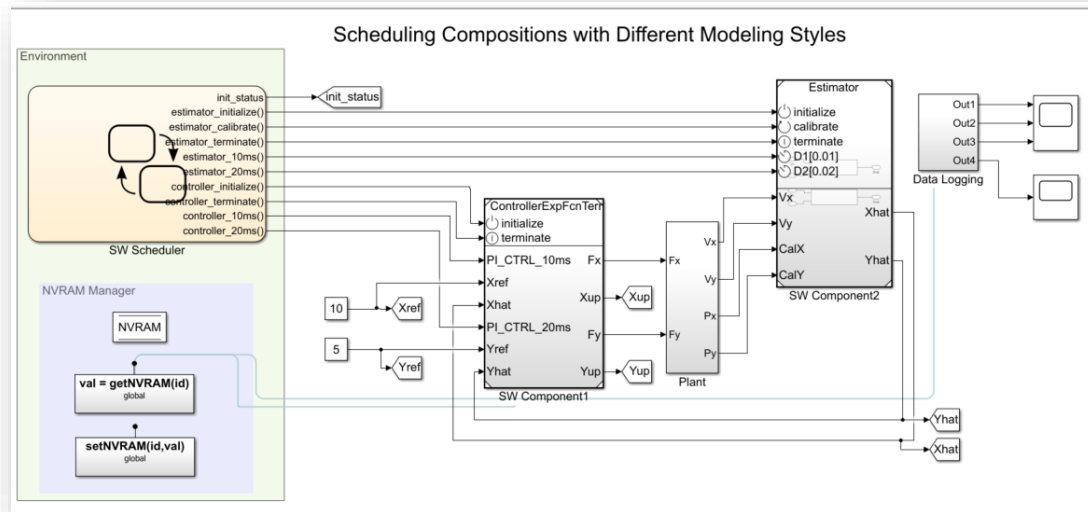
## Implement an embedded PID auto-tuning algorithm

- Automatically tune PID controller gains in real time against a physical plant
- No model of plant dynamics required
- Deploy the auto-tuning algorithm to embedded software using automatic code generation



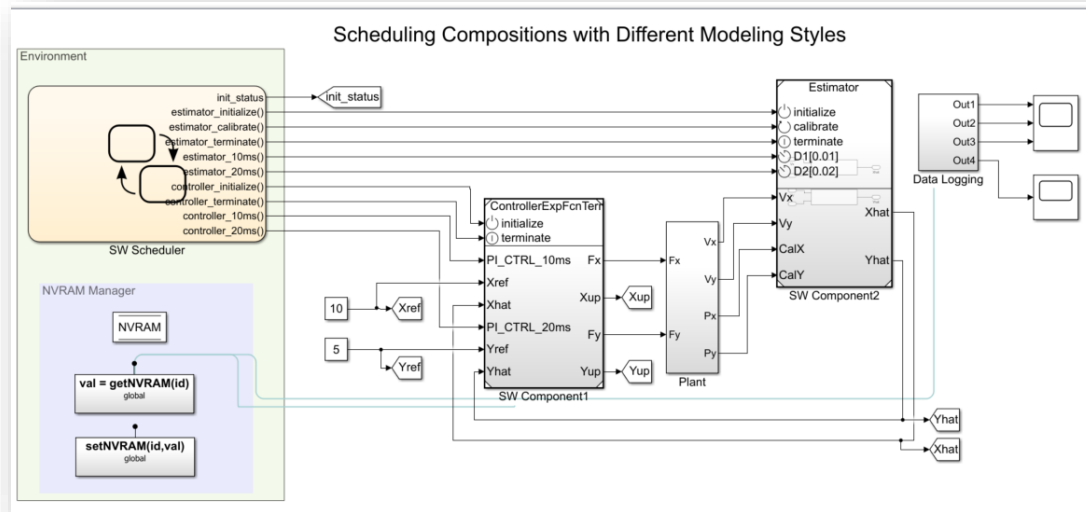
# Prepare Your Model for Code Generation

Prepare model components  
for code generation



# Prepare Your Model for Code Generation

Prepare model components for code generation



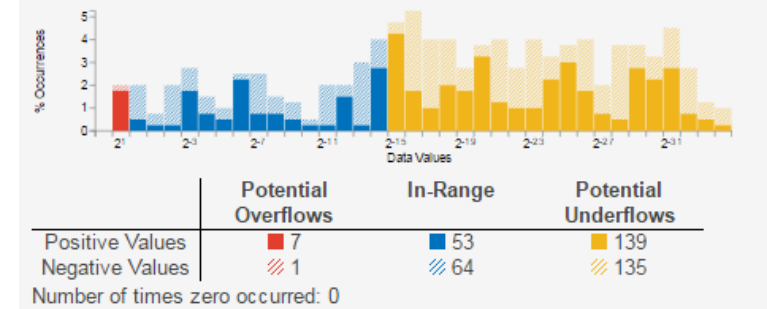
Prepare model data for code generation

The screenshot shows the "FIXED-POINT TOOL" interface. Key features are highlighted with blue boxes and a green arrow:

- Simulation Ranges:** A dropdown menu for selecting simulation ranges.
- Propose Data Types:** A button to automatically propose data types for the model.
- Compare Results:** A button to compare simulation results.

The interface also shows a "MODEL HIERARCHY" tree on the left and a "Results" table on the right.

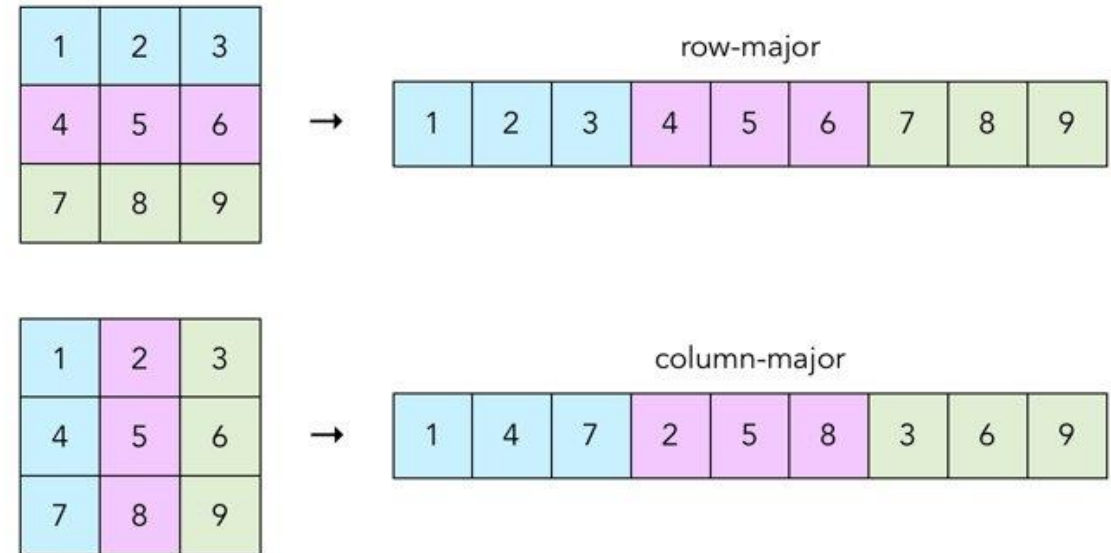
Visualization of Simulation Data



# Generate Code from Simulink Models

## Access and define all the information in your model related to code generation

- View and define implementation data in one place
- View implementation details without model details
- Improve code performance and ease integration with other C code



## Row-major memory layout option

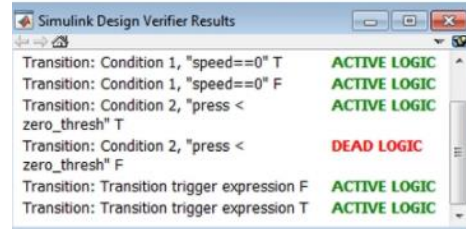
# Verification and Validation

## Products for the entire workflow

### Simulink Requirements R2017b

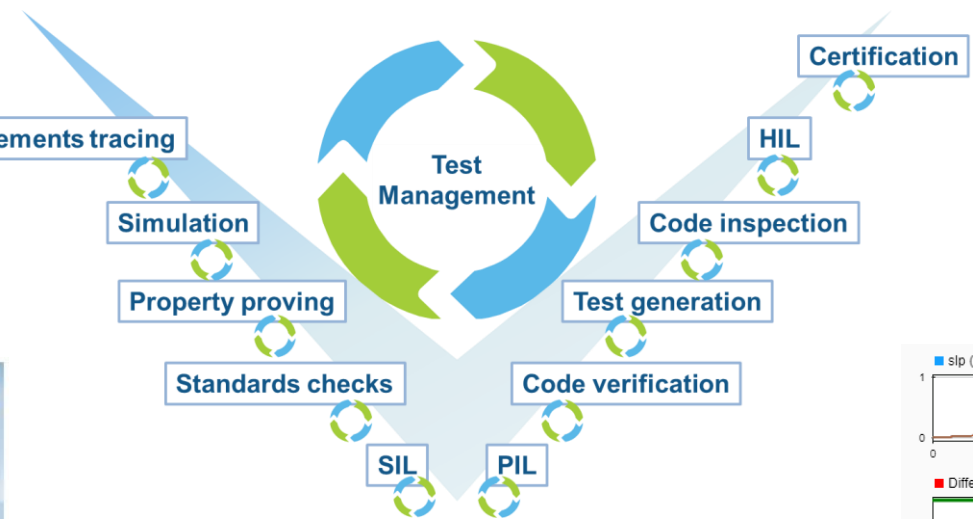


### Simulink Design Verifier

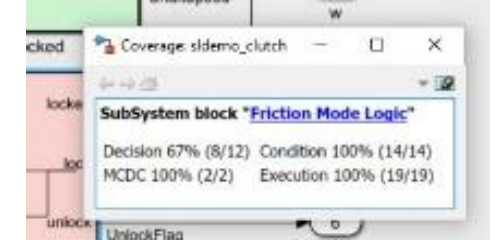


### Simulink Check R2017b

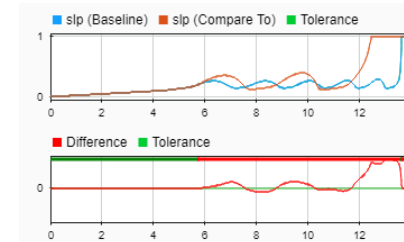
- Modeling Standards for Secure Coding (CERT C, CWE, ISO/IEC TS 17961)
  - Check configuration parameters for secure coding standards
  - Check for blocks not recommended for C/C++ production code deployment
  - Check for blocks not recommended for secure coding standards
  - Check usage of Assignment blocks
  - Check for switch case expressions without a default case
  - Check for bitwise operations on signed integers
  - Check for equality and inequality operations on floating-point values
  - Check integer word lengths
  - Detect Dead Logic



### Simulink Coverage R2017b



### Simulink Test



### Polyspace

```

29  ----- INTEGER DIVIS
30  +-----+
31  int_intdiv(int p)
32  {
33      int i;
34      int j = 1;
35
36      i = 1024; (j - p);
37      return i;
    
```

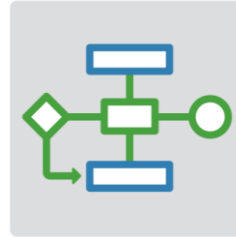
Probable cause for 'Integer division by zero':  
intdiv(1);  
operator / on type int 32  
left: 1024  
right: 0  
result: [-1024 .. 1024]



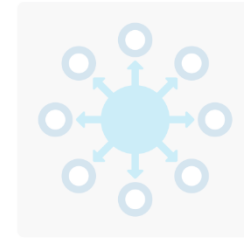
## Platform Productivity



## Workflow Depth



## Application Breadth

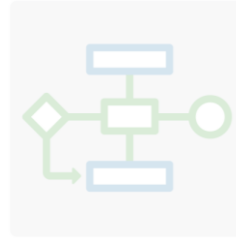


- **Deployment of MATLAB Algorithms and Applications**
- **Code Generation from Simulink Models**
- **Verification and Validation**

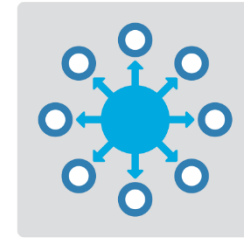
## Platform Productivity



## Workflow Depth

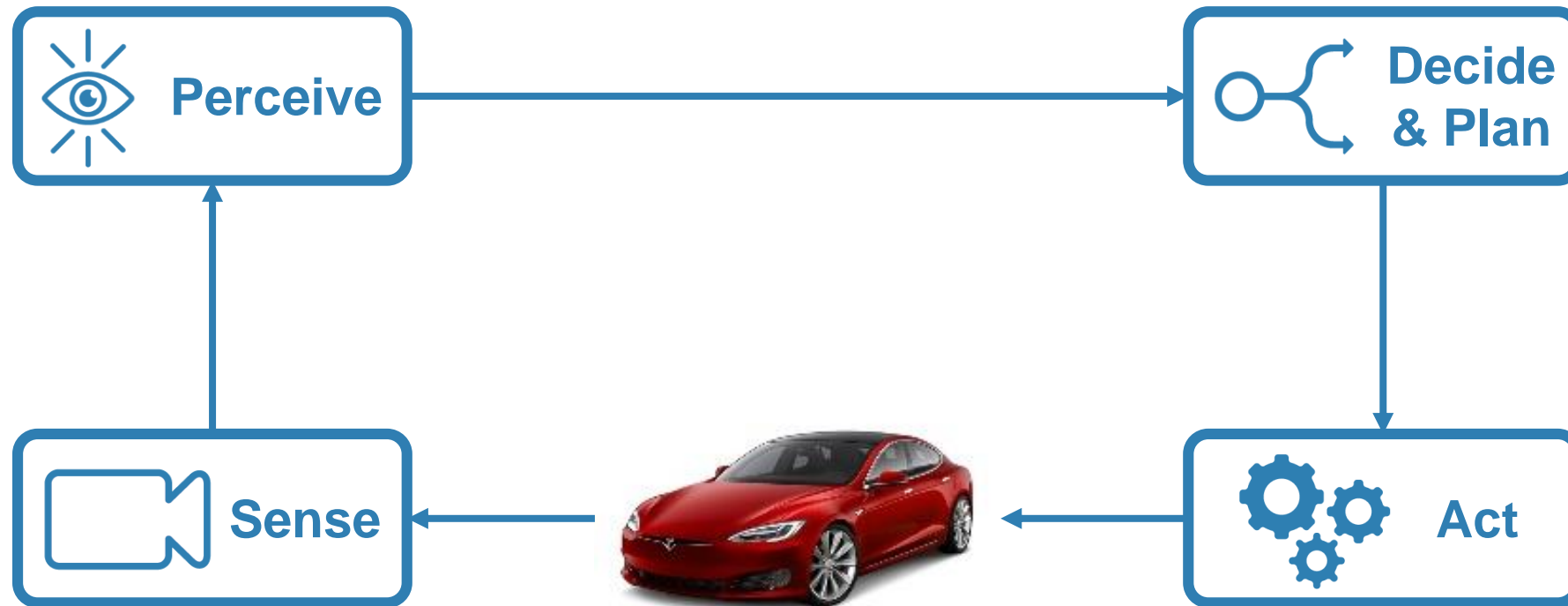


## Application Breadth



- **Autonomous Systems**
- **Wireless Communications**
- **Artificial Intelligence (AI)**

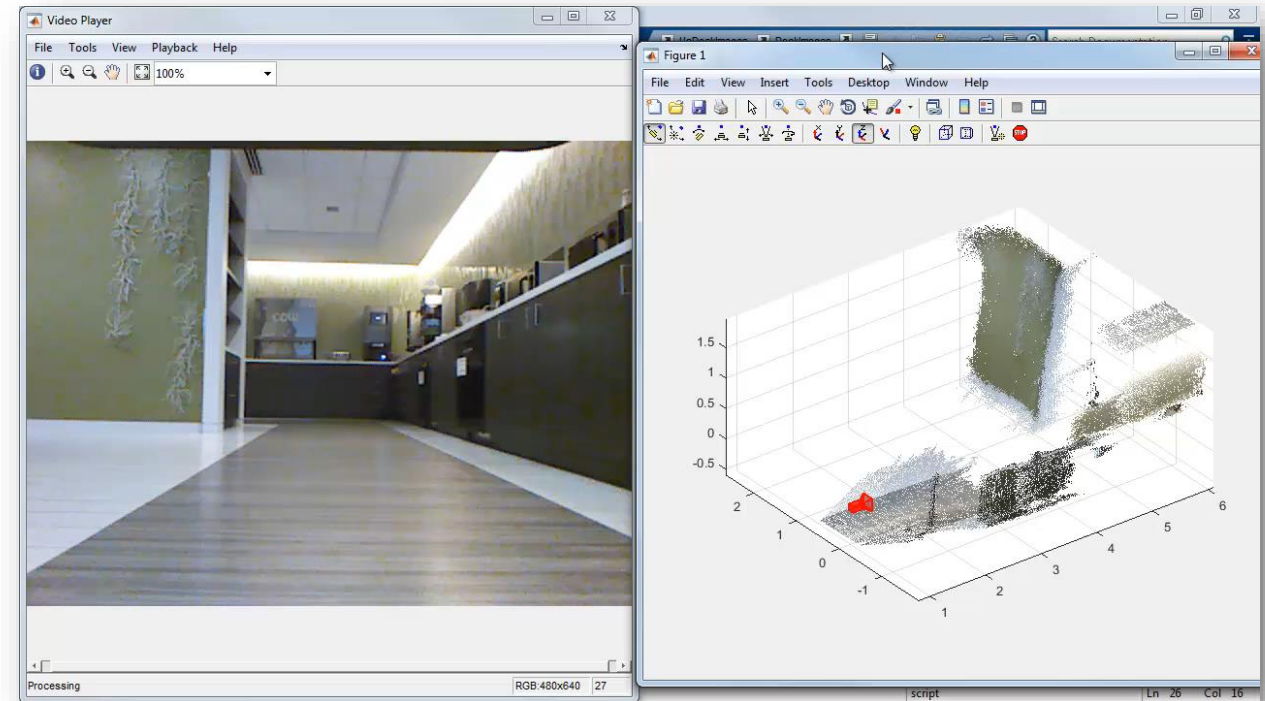
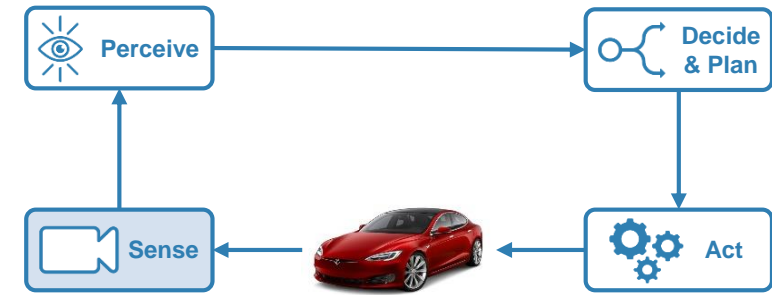
# Designing Autonomous Systems



# Designing Autonomous Systems

## Mapping of environments using sensor data

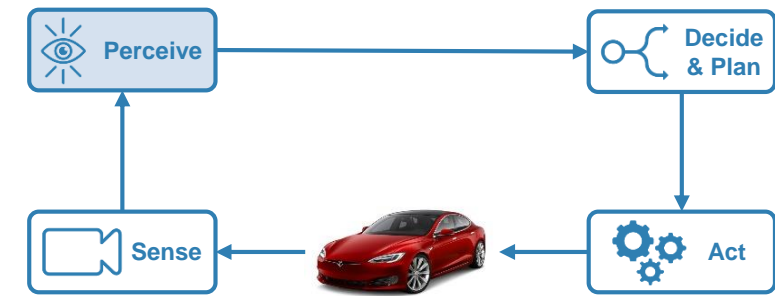
- Segment and register lidar point clouds
- Lidar-Based SLAM: Localize robots and build map environments using lidar sensors



# Designing Autonomous Systems

## Understanding the environment using computer vision and deep learning techniques

- Object detection and tracking
- Semantic segmentation using deep learning

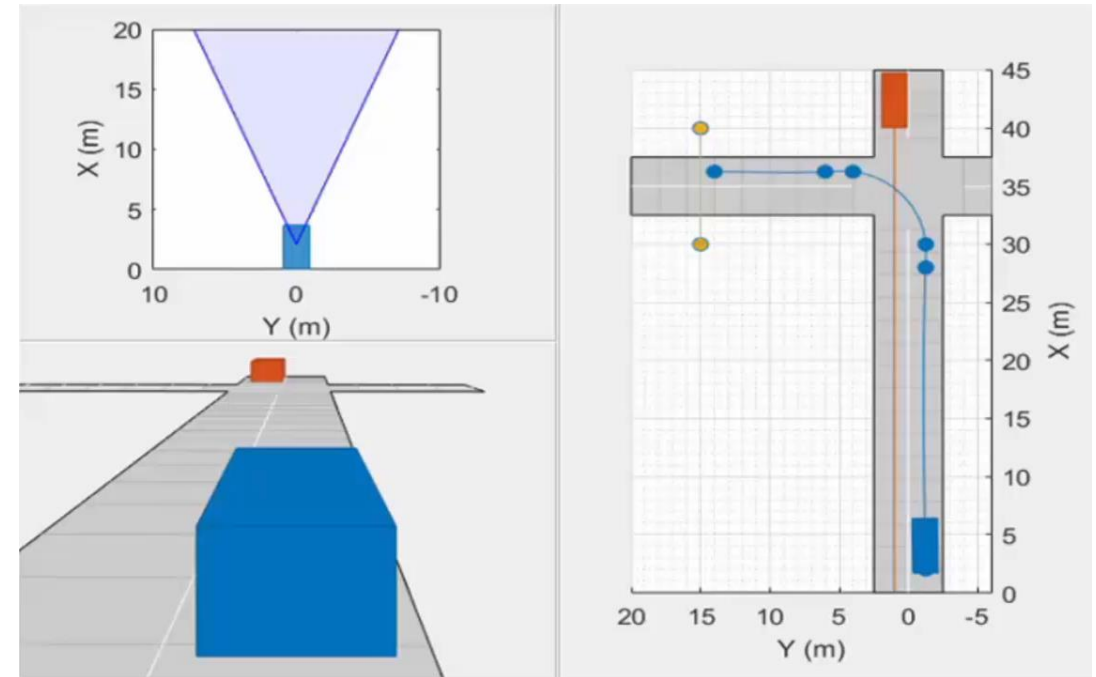
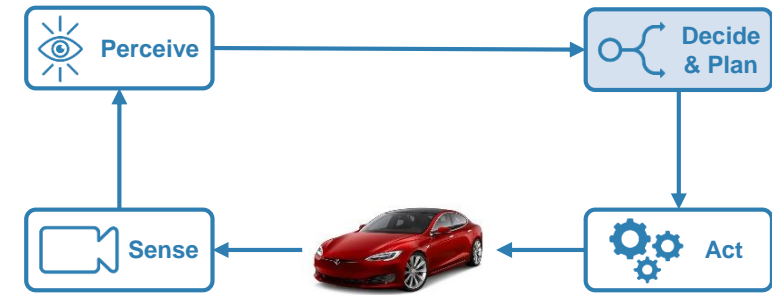


CamVid Database: Brostow, Gabriel J., Julien Fauqueur, and Roberto Cipolla. "Semantic object classes in video: A high-definition ground truth database." *Pattern Recognition Letters* Vol 30, Issue 2, 2009, pp 88-97.

# Designing Autonomous Systems

Design synthetic driving scenarios to test controllers and sensor fusion algorithms

- Interactively design synthetic driving scenarios composed of roads and actors (*vehicles, pedestrians, etc.*)
- Generate visual and radar detections of actors

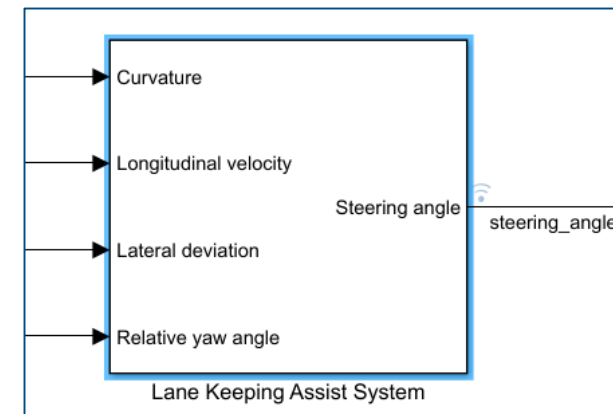
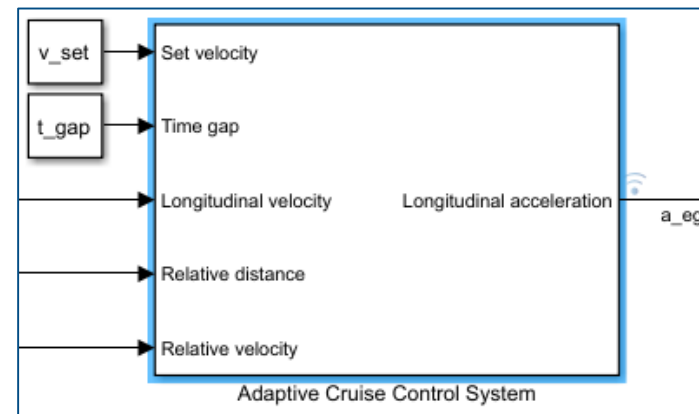
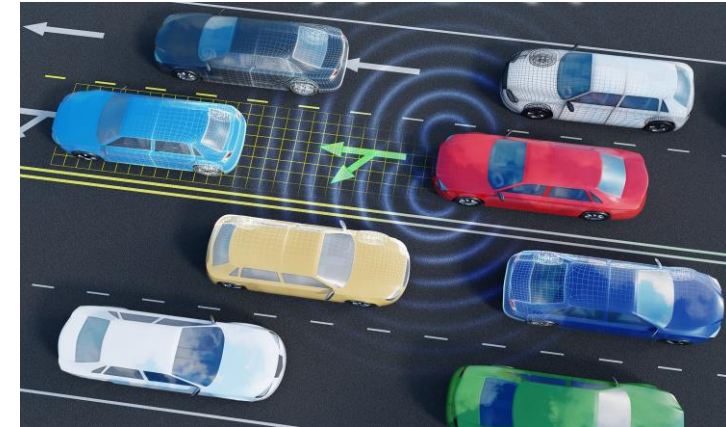
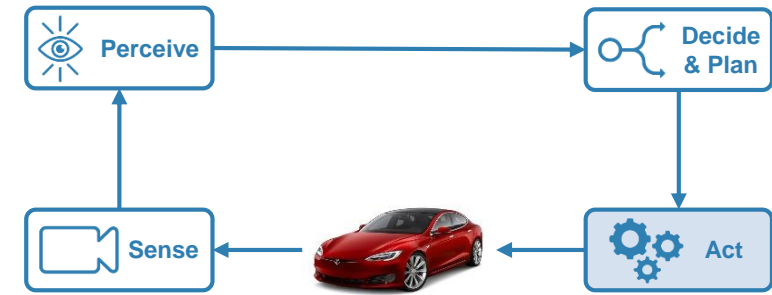


Driving Scenario Designer App

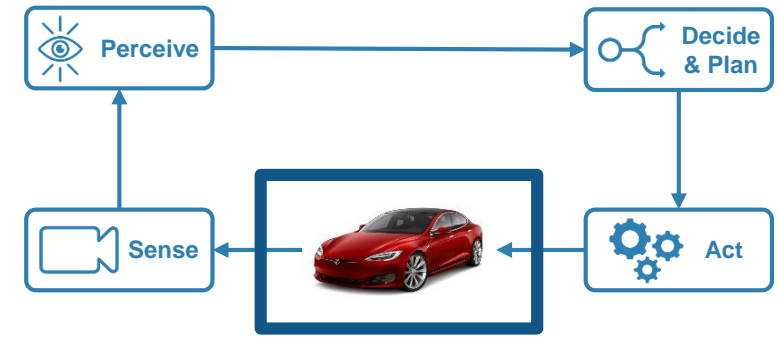
# Designing Autonomous Systems

## Model predictive control for adaptive cruise control and lane-keeping algorithms

- Use prebuilt blocks instead of starting from scratch
- Simplified application-specific interfaces for configuring model predictive controllers
- Flexibility to customize for your application



# Full Vehicle Simulation



Ride & handling



Chassis controls



Automated Driving



# Design with the Latest Wireless Standards



**Lte**™  
Advanced  
Pro



**5G**™



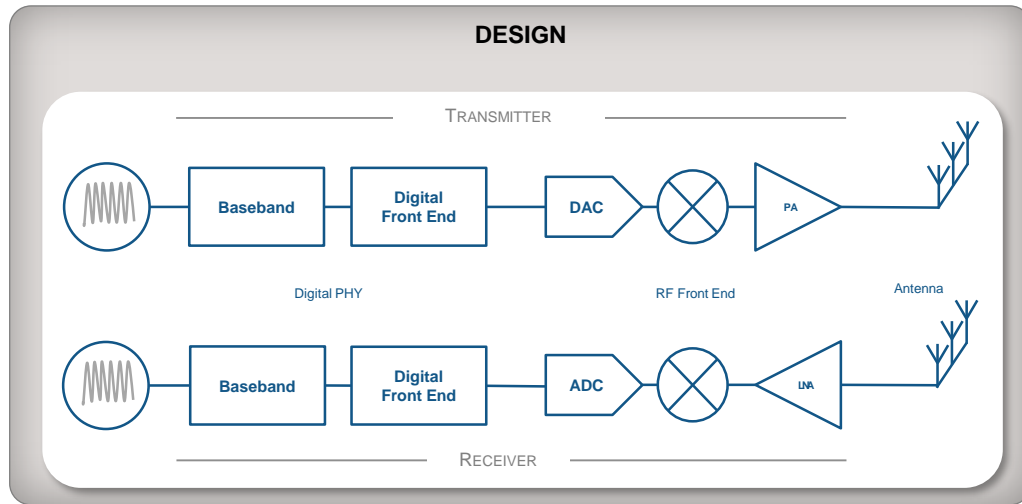
**WiFi**™  
802.11ax



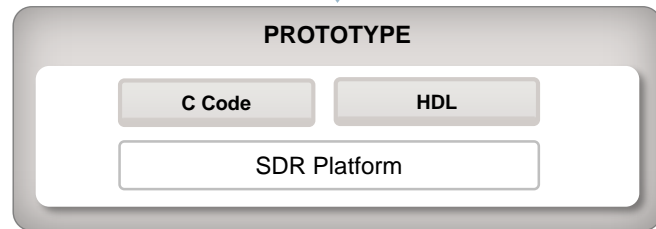
**ZigBee**®

**NB-IoT**

# Model-Based Design for Wireless Communications



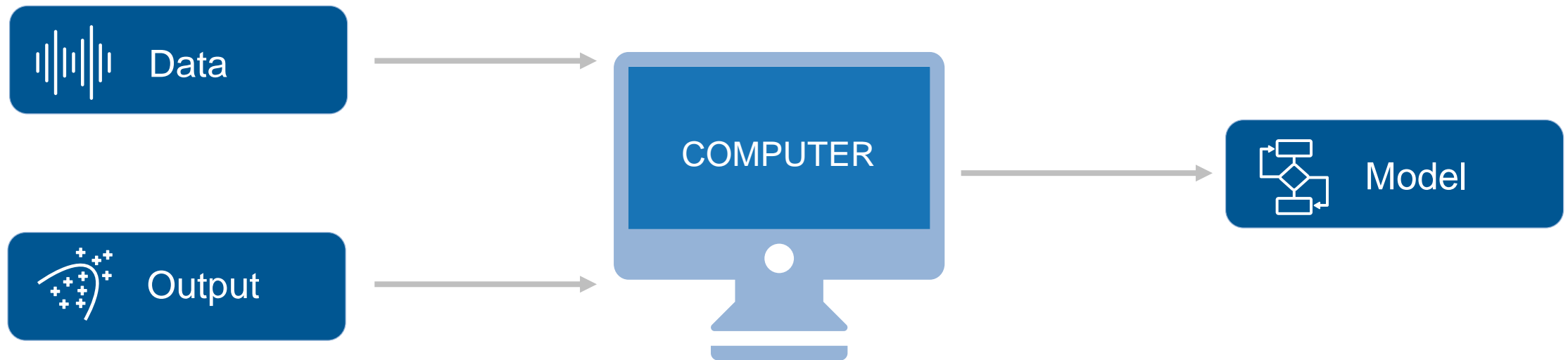
- Algorithm Design and Verification
- RF, Digital and Antenna Co-Design
- System Verification and Testing
- Rapid Prototyping and Production



**Code Generation and Verification**

Fixed-Point Designer  
 HDL Coder  
 HDL Verifier  
 LTE HDL Toolbox **R2017b**  
 Embedded Coder

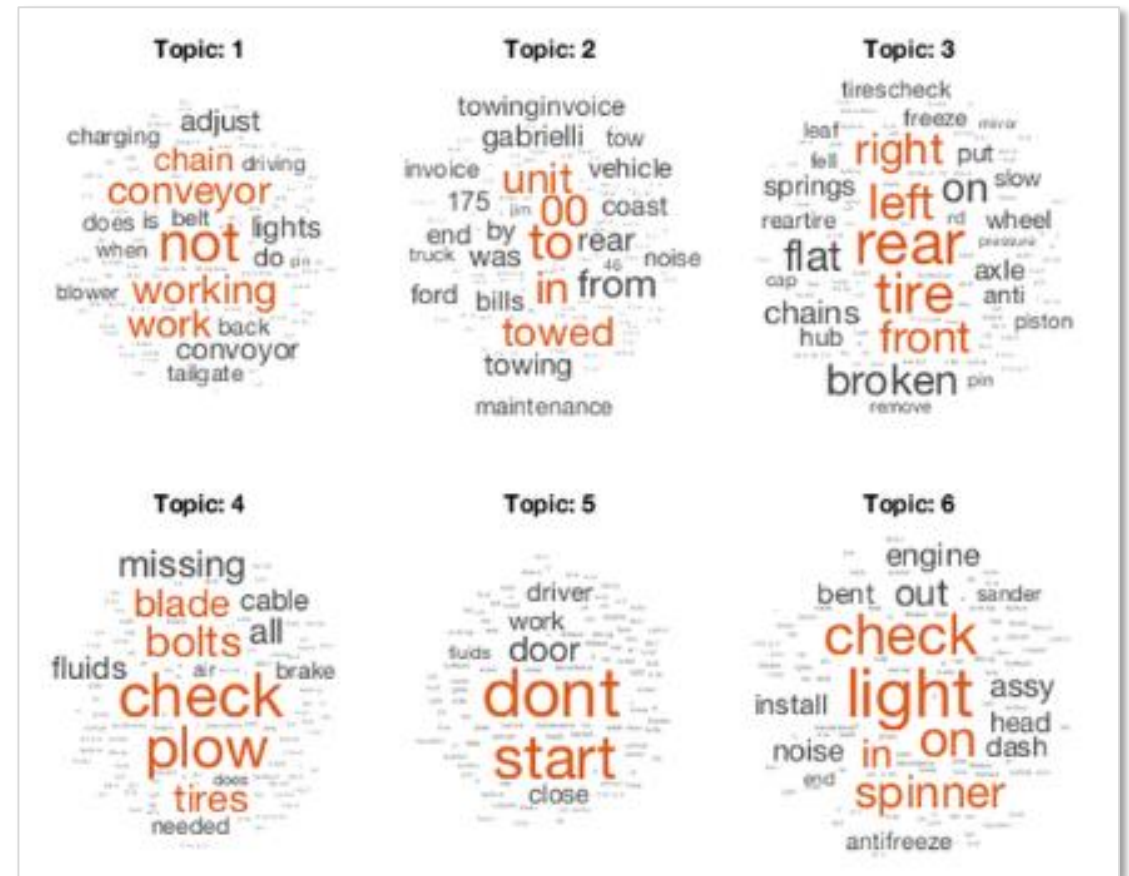
# Artificial Intelligence



# Text Analytics

## Work with text from equipment logs and operator reports

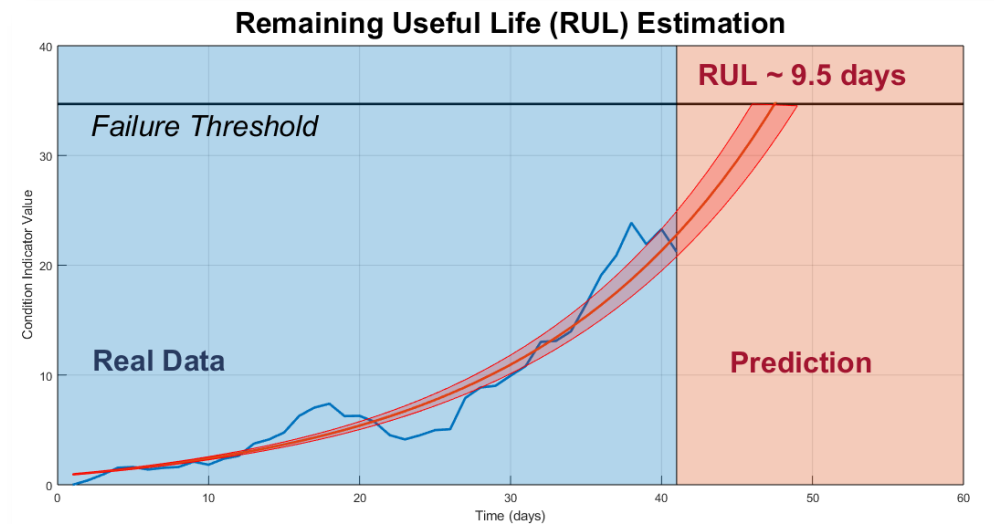
- **Preprocess** raw text data by extracting, filtering, and splitting
- **Visualize** text using word clouds and text scatter plots
- **Develop** predictive models using built-in machine learning algorithms (LDA, LSA, word2vec)



# Predictive Maintenance

## Design and test condition monitoring and predictive maintenance algorithms

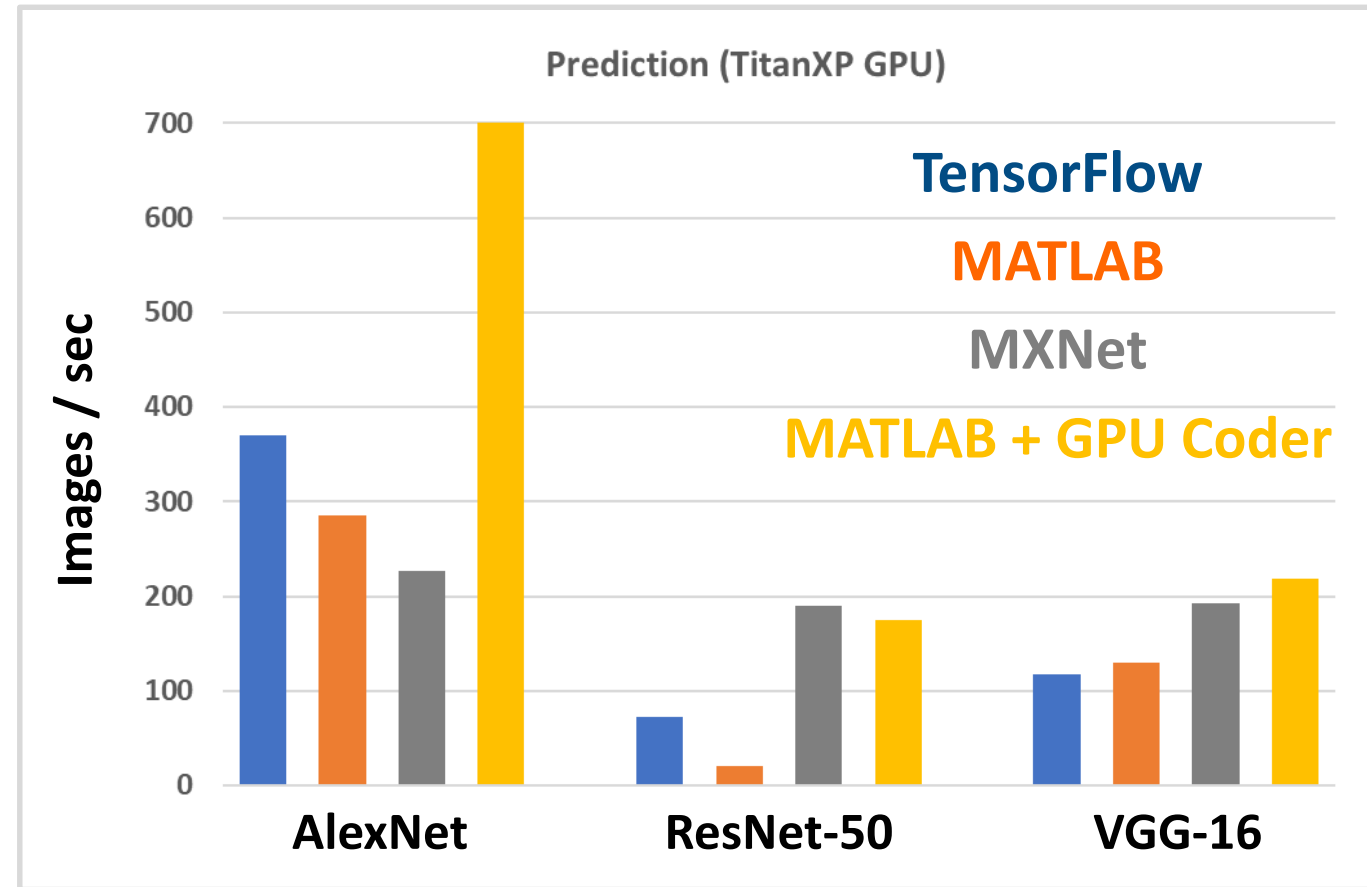
- Import sensor data from local files and cloud storage (*Amazon S3, Windows Azure Blob Storage, and Hadoop HDFS*)
- Use simulated failure data from Simulink models
- Estimate remaining useful life (RUL)
- Get started with examples (*motors, gearboxes, batteries, and other machines*)



# Deep Learning

## Design, build, and visualize convolutional neural networks

- Access the latest models
- Import pretrained models and use transfer learning
- Automate ground-truth labeling using apps
- Design and build your own models
- Use NVIDIA GPUs to train your models
- Automatically generate high-performance CUDA code for embedded deployment



FREE

# Learn to Use MATLAB for Deep Learning in 2 Hours

Launch Deep Learning Onramp

The screenshot shows the MATLAB Deep Learning Onramp interface. The top navigation bar includes "My Courses", "Deep Learning Onramp" (51% complete), and a user profile for "Chal Chitale". The main content area is titled "2.2 Making Predictions: (1/2) Make a prediction".

**Task 2** instructions:

Info: You can use the `classify` function to make a prediction on an image.

```
pred = classify(net,img);
```

Use the `classify` function with the pretrained AlexNet network to predict the subject of the image stored in the variable `img1`. Store the network's prediction in a variable called `pred1`.

You may want to leave off the semicolon to see the result.

Buttons: Submit, Hint, See Solution, Next task

Test Suite: Correct! ✓ Is pred1 created correctly? Show test suite details

**Task 1** instructions:

Load pretrained network

Task 1: Use the `alexnet` function to load a pretrained network.

```
deepnet = alexnet;
```

**Import, view, and classify an image**

Import and display the image in `file01.jpg`.

```
img1 = imread('file01.jpg');
imshow(img1)
```

Task 2: Classify the image in the variable `img1`.

```
pred1 = classify(deepnet,img1)
```

**Classify further images**

Task 3: Classify the images in `file02.jpg` and `file03.jpg`.

```
img2 = imread('file02.jpg');
```

The interface also shows a workspace window with a preview of a beach image and the output: `pred1 = categorical seashore`.

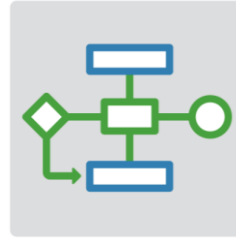
# What's New in MATLAB and Simulink?

## Platform Productivity



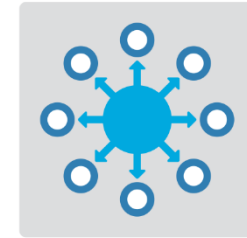
- Design Creation
- Analysis
- Simulation, Scaling
- Collaboration

## Workflow Depth



- Deployment
- Code Generation
- Verification and Validation

## Application Breadth



- Autonomous Systems
- Wireless Communications
- Artificial Intelligence (AI)



# Upgrade your MATLAB Code and Simulink Models

Web Browser - (3 Errors) Code Compatibility Report

(3 Errors) Code Compatibility Report

Code Compatibility Report [Top](#) [3 Errors](#) [1 Warning](#) [304 Checks](#) [2 Files](#)

Analysis Date: 05-Sep-2017 14:32:08  
MATLAB Version: R2017b

**Incompatibility and Syntax Errors**

Row	Filename	Line	Description
1	classifyBloodPressure.m	18	TREEFIT has been removed
2	classifyBloodPressure.m	21	TREEDISP has been removed. Use TREEVIEW methods instead.
3	classifyBloodPressure.m	24	TREEVAL has been removed. Use TREEPREDICT methods instead.

**Warnings and Other Recommendations**

Row	Filename	Line	Description
1	classifyBloodPressure.m	Z	RAND or RANDN with the 'state' argument is no longer recommended. Use RANDI or RANDJ instead.

Upgrade Advisor - sf\_climate\_control

File Edit Run Settings Help

Find:

**Upgrade Project Report**

100% Passed

	Models	Libraries	MATLAB Code
Passed	7	1	8
Need attention	-	-	-

Show:

Filename	Check Name	Result
AnalogControl.mdl	Check model settings for migration to simplified initialization mode	Passed
analyzeModelFiles.m	Check that the model is saved in SLX format	Passed with fixes
billOfMaterials.m	Check usage of function-call connections	Need attention
checkCodeProblems.m	Check and set embedded target model to use ert.tlc system target file	Passed
DigitalControl.slx	Check and update masked blocks in library to use promoted parameters	Passed
f14_airframe.slx	Check and update mask image display commands with unnecessary imread() function calls	Passed
f14_airframe_test.m	Check and update mask to affirm icon drawing commands dependency on mask workspace	Passed
find_top_models.m	Check and update model to use toolchain approach to build generated code	Passed
LinearActuator.slx		
NonLinearActuator.mdl		
rebuild_s_functions.m		
runUnitTest.m		
slproject_f14.slx		
upgrade_project.m		
vertical_channel.slx		
wind_gust_lib.slx		

Checks run on 02/01/2018 10:44

**Identify Variant Model blocks and convert those to Vari**

Analysis

Upgrade Variant Model blocks to Variant Subsystems contain offers enhanced capabilities while maintaining equivalent fun variant models will be removed in a future release.

Result: ✔ Passed

Identify Variant Model blocks at model level.

**Passed**

No Variant Model blocks found.

MATLAB EXPO 2018

41

MATLAB EXPO 2018

