



Oprogramowanie
Naukowo-Techniczne
sp. z o.o.

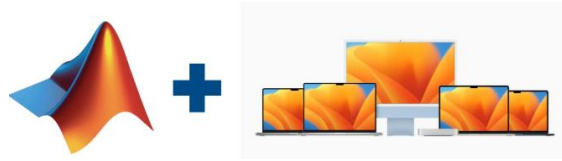
R2023b

What's new in MATLAB and Simulink 2023b
Paulina Kozakiewicz, Junior Application Engineer

Cracow, 18.10.2023 r.

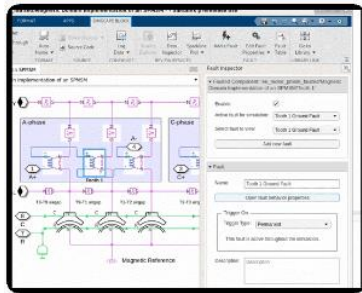
Agenda

1. Overview
2. New Products
3. Major Updates



MATLAB on Macs

R2023b



Simulink Fault Analyzer

NEW PRODUCT

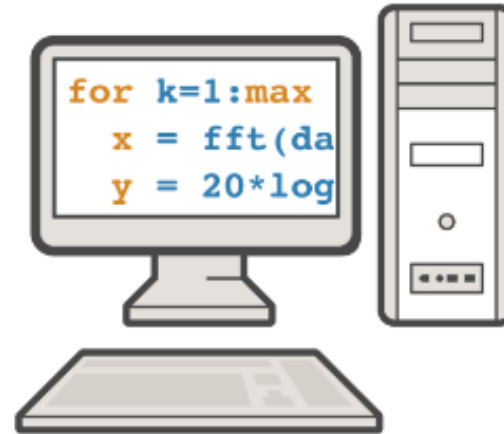


Polyspace Test

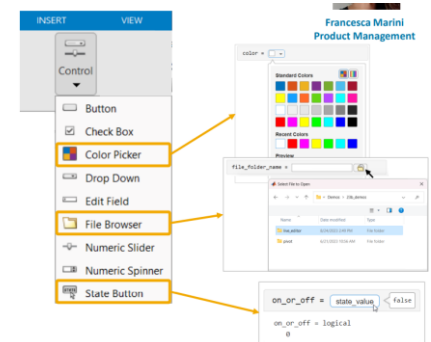
NEW PRODUCT



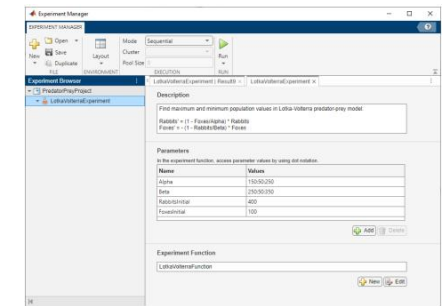
Simulink



MATLAB



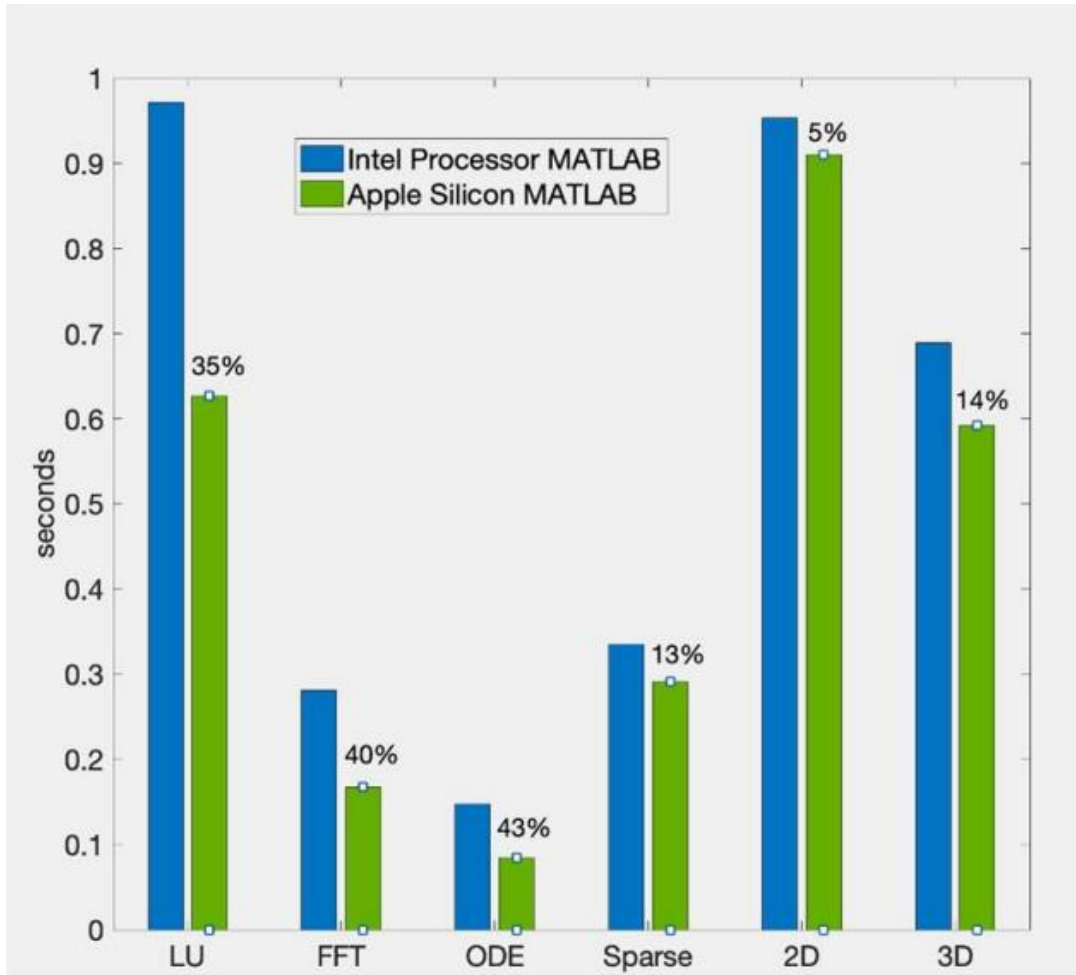
Live Editor



Experiment Manager App

MATLAB on Macs

MATLAB R2023b runs natively on M1 & M2 (Apple silicon) Macs
 (Earlier releases run in Intel emulation mode)



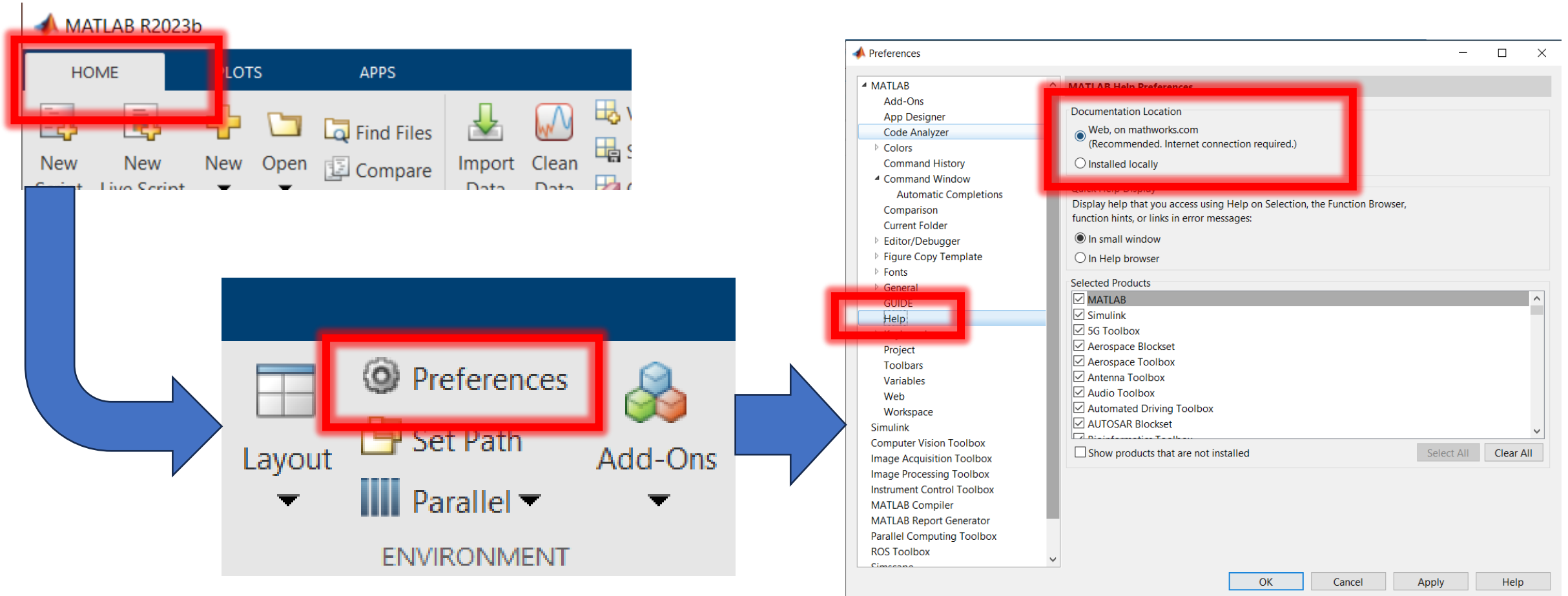
(time in seconds, lower is better)

Native Apple silicon MATLAB is *fast*

- Startup under 3 seconds
- Desktop is more responsive
- Battery life is improved

Access to documentation has changed

Doc isn't installed with MATLAB since **R2023a**



Ways to make sure your example and data files work

1. Get Example Using Help Center Web Page

Open in MATLAB Online

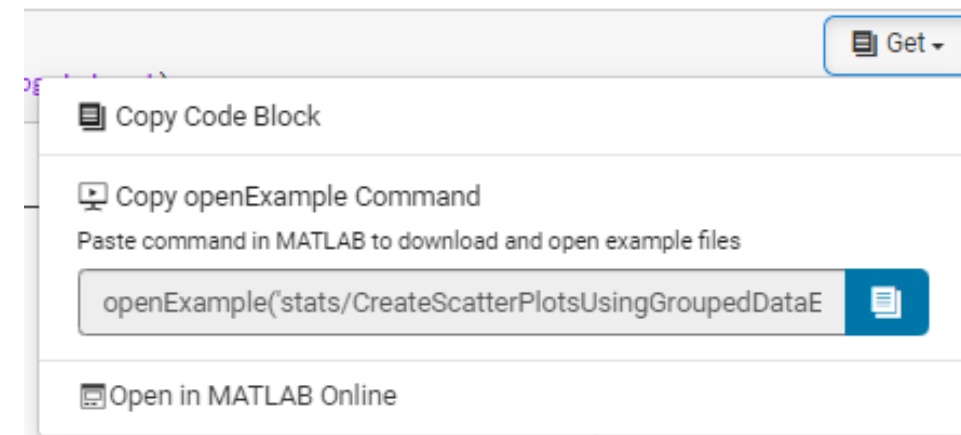
Copy Command

2. Get Example Files Using the MATLAB Help Browser

Open Live Script

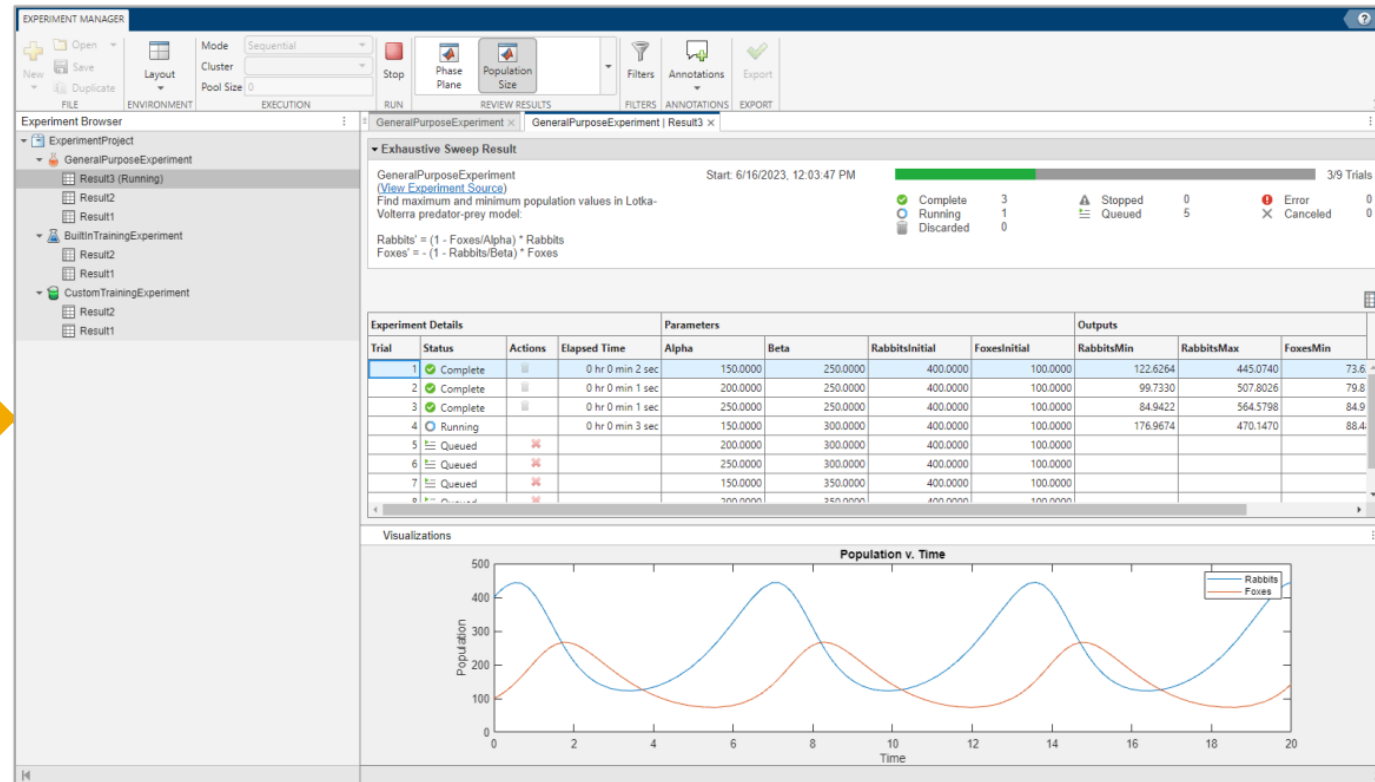
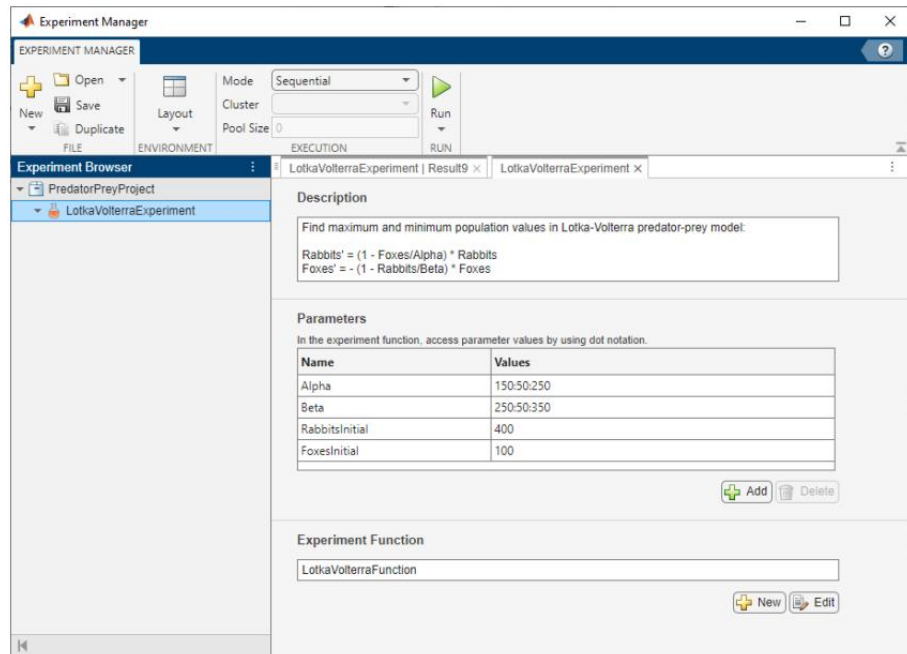
3. Get Example Files Using the Command line

```
openExample(examplename)
openExample(sfile)
openExample(__,Name,Value)
```



Experiment Manager App

Design experiments to run MATLAB code and use the app to visualize, filter and compare results

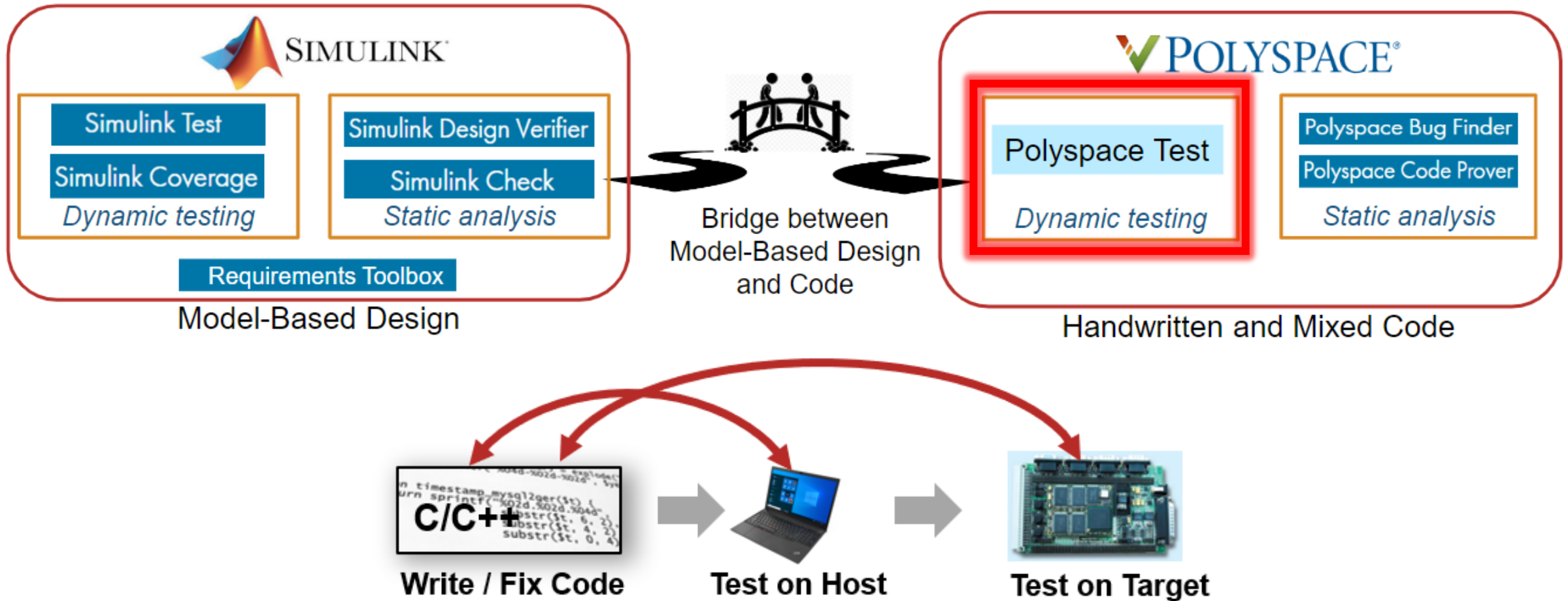


Run experiments sequentially or in parallel with Parallel Computing Toolbox

NEW PRODUCTS

Polyspace Test NEW PRODUCT

Develop, manage, and execute tests for C and C++ code in embedded systems



Polyspace Test **NEW PRODUCT**

Dynamic Test Capabilities with Polyspace Test

Key Capabilities

- **Unified platform for static analysis and dynamic test**
 - Includes ability to publish static + dynamic results to Polyspace Access
 - Single vendor solution
- **Integration with Polyspace Code Prover to automatically justify dead code**
 - Could help speedup review of coverage results produced by Polyspace Test
- **Automatic generation to speed up test creation**
 - Use of formal methods to generate test for achieving coverage goals
- **Check unit tests with Polyspace Bug Finder / Code Prover to confirm unit test is robust**
 - Confirm that there are no issues with the code under test when tested under specific test scenarios
- **Leverages the on-target testing capabilities from the MBD toolchain**
 - Author your target support once for generated and handwritten code



Qualification Kit

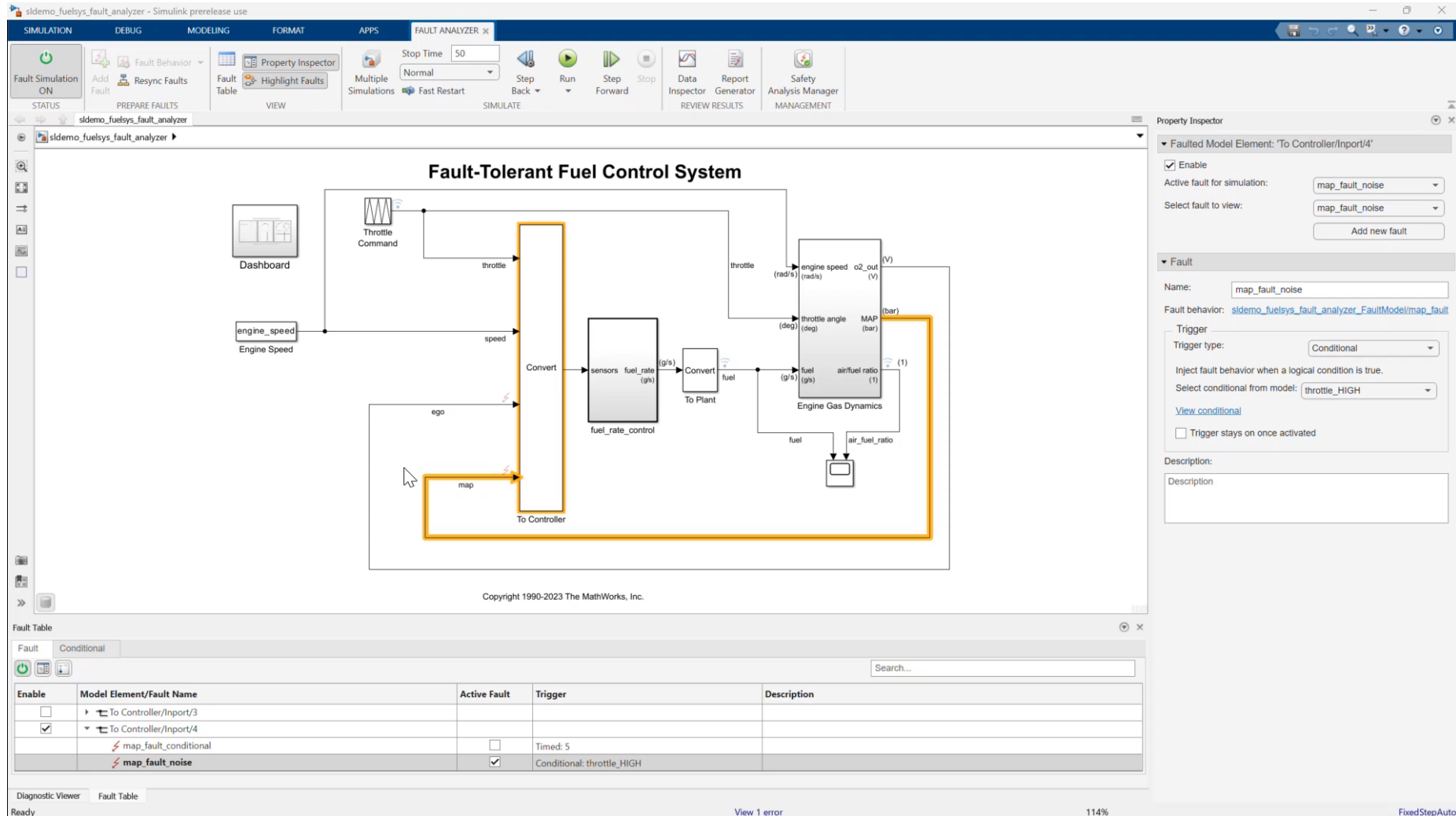
Qualify Polyspace Test according to DO-178C and DO-330 Standards



Simulink Fault Analyzer

NEW PRODUCT

Model faults and analyze effects



The screenshot displays the Simulink Fault Analyzer interface for a 'Fault-Tolerant Fuel Control System'. The main workspace shows a block diagram with components like 'Throttle Command', 'Convert', 'fuel_rate_control', 'To Plant', 'Engine Gas Dynamics', and 'To Controller'. A fault 'map_fault_noise' is highlighted in orange. The 'Property Inspector' on the right shows the fault configuration, including its name, behavior, and trigger conditions. The 'Fault Table' at the bottom lists the configured faults.

Enable	Model Element/Fault Name	Active Fault	Trigger	Description
<input type="checkbox"/>	To Controller/Inport/3			
<input checked="" type="checkbox"/>	To Controller/Inport/4	<input type="checkbox"/>	Timed: 5	
	map_fault_conditional	<input type="checkbox"/>		
	map_fault_noise	<input checked="" type="checkbox"/>	Conditional: throttle_HIGH	

Simulink Fault Analyzer

NEW PRODUCT

Multiple Simulations

The screenshot displays the Simulink Fault Analyzer interface. The main workspace shows a simulation model with blocks for 'Environment', 'Longitudinal Driver', and 'Controller'. A 'Fault Table' window is open at the bottom, listing several faults. The 'Property Inspector' window is also open, showing the configuration for the 'HighTemperatureFault'.

Enable	Model Element/Fault Name	Active Fault	Trigger	Description
<input type="checkbox"/>	Environment/Constant6/Outpo...			
<input checked="" type="checkbox"/>	HighTemperatureFault		Conditional: highS...	
<input checked="" type="checkbox"/>	LowTemperaturFault		Conditional: Samp...	
<input type="checkbox"/>	Environment/Constant7/Outpo...			
<input checked="" type="checkbox"/>	HighPressureFault		Always On	

Property Inspector - Fault Configuration:

- Enable:
- Select fault to view: HighTemperatureFault
- Name: HighTemperatureFault
- Fault behavior: EvReferenceApplication FaultModel/HighTemperatureFau
- Trigger type: Conditional
- Inject fault behavior when a logical condition is true.
- Select conditional from model: highSpeedCondition
- View conditional: [View conditional](#)
- Trigger stays on once activated:
- Description: Description

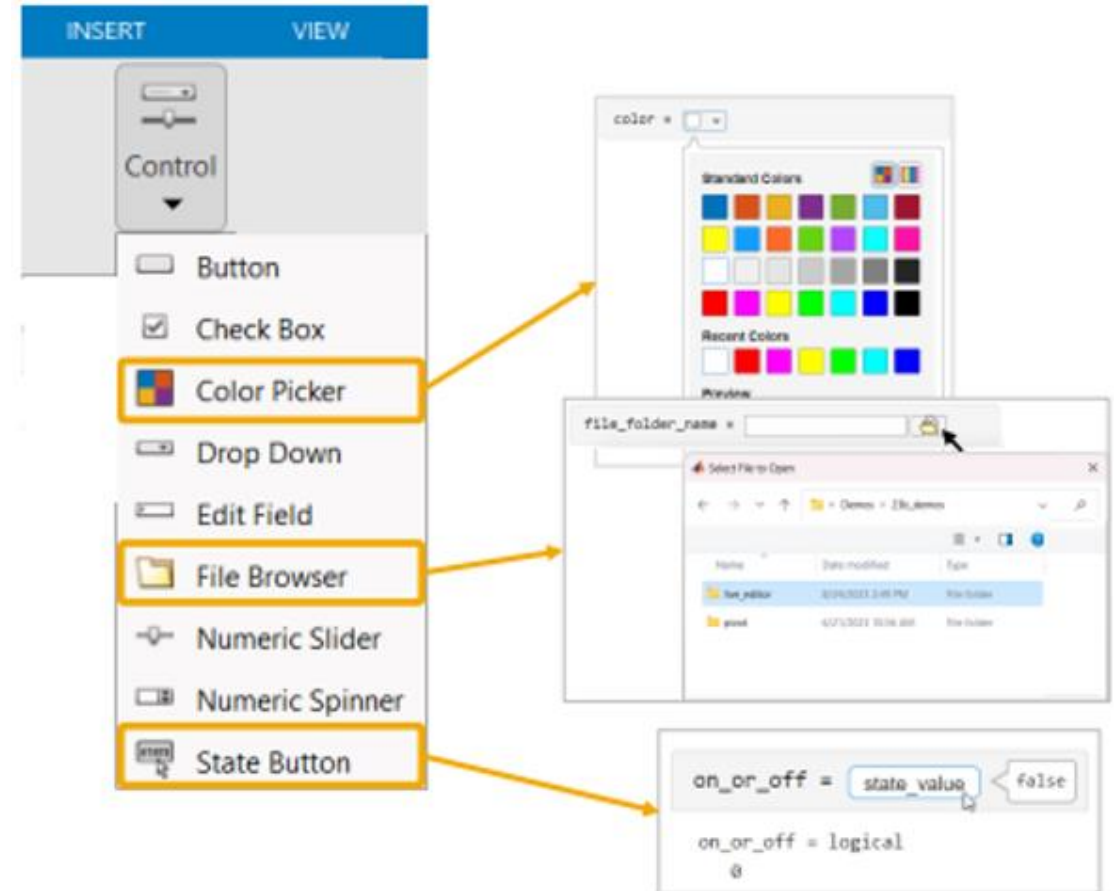
MAJOR UPDATES

MATLAB

Updates for interactive controls, *pivot* function, custom Live Editor tasks and a lot more

New Interactive Controls!

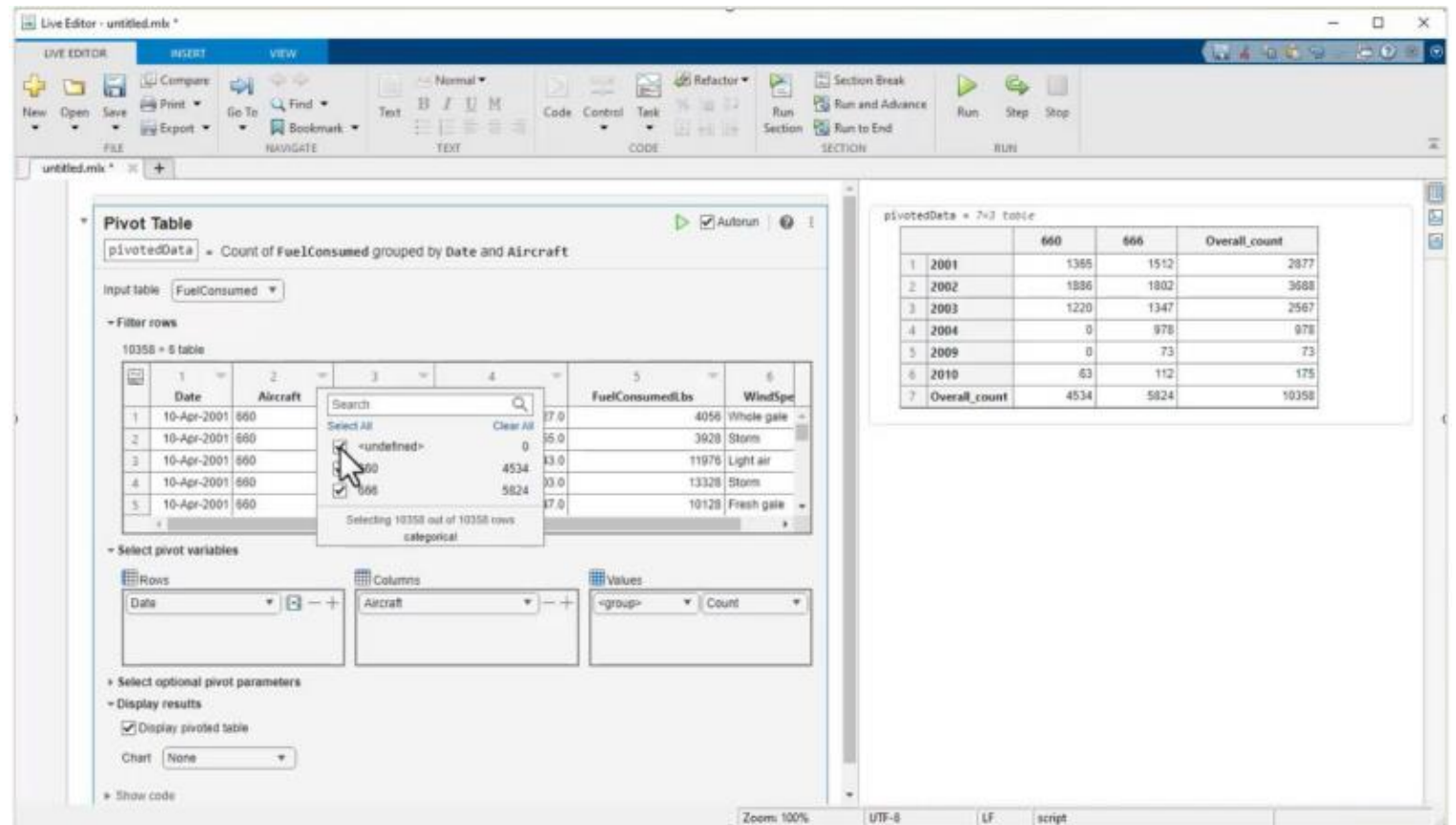
- **Color Picker** for interactively assigning a color value to a variable
- **File Browser** for interactively assigning a file or folder name to a variable
- **State Button** for interactively toggling a variable between two binary states (e.g., 0/1, True/False)



MATLAB

pivot function

- ***pivot*** function was released **R2023a**
- New live task makes the functionality interactive



The screenshot shows the MATLAB Live Editor interface with the Pivot Table tool. The tool is titled "Pivot Table" and is configured to pivot the "FuelConsumed" table. The pivot variables are "Date" (Rows) and "Aircraft" (Columns), with "Count" as the value. The resulting pivoted data table is displayed on the right.

Pivot Table Configuration:

- Input table: FuelConsumed
- Filter rows: 10358 x 6 table
- Rows: Date
- Columns: Aircraft
- Values: -group- Count
- Display results: Display pivoted table
- Chart: None

Pivoted Data Table:

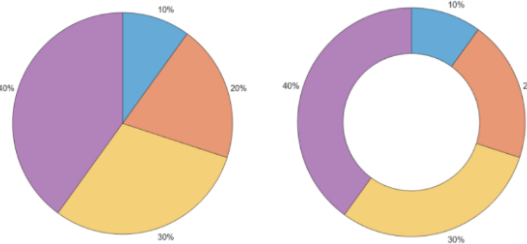
	660	666	Overall_count
1 2001	1365	1512	2877
2 2002	1896	1802	3698
3 2003	1220	1347	2567
4 2004	0	978	978
5 2009	0	73	73
6 2010	63	112	175
7 Overall_count	4534	5824	10358

MATLAB

Graphics: new charts, colormaps, conveniences; New graphics convenience functions to simplify your work

- *piechart* and *donutchart*

New!



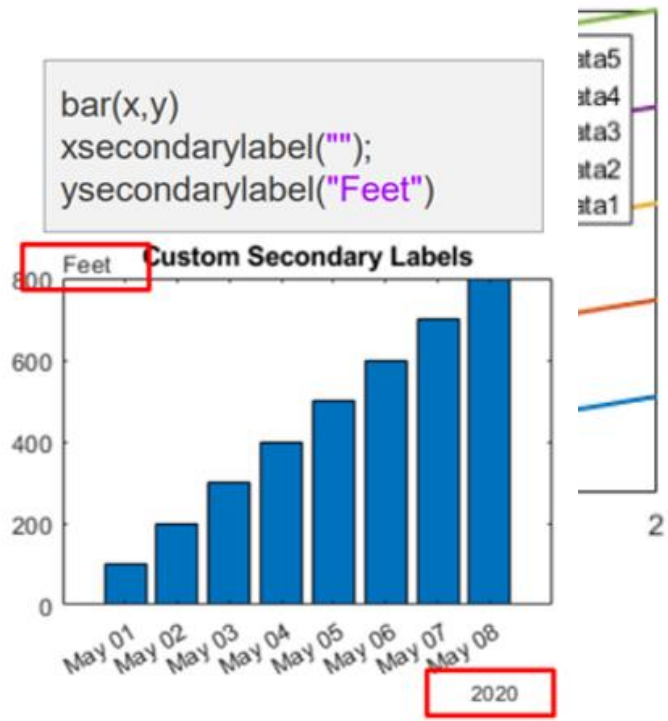
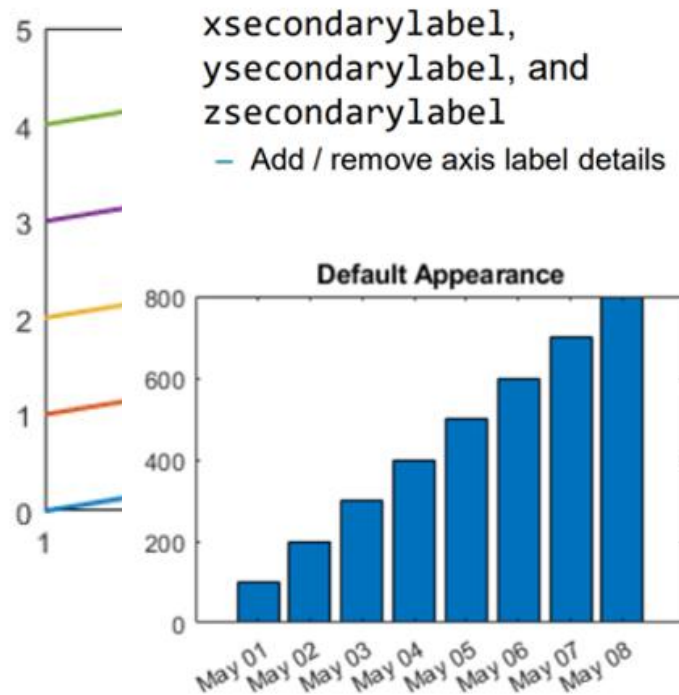
```

lgd = legend;
lgd.Direction = "reverse";
  
```

- Change order of legend

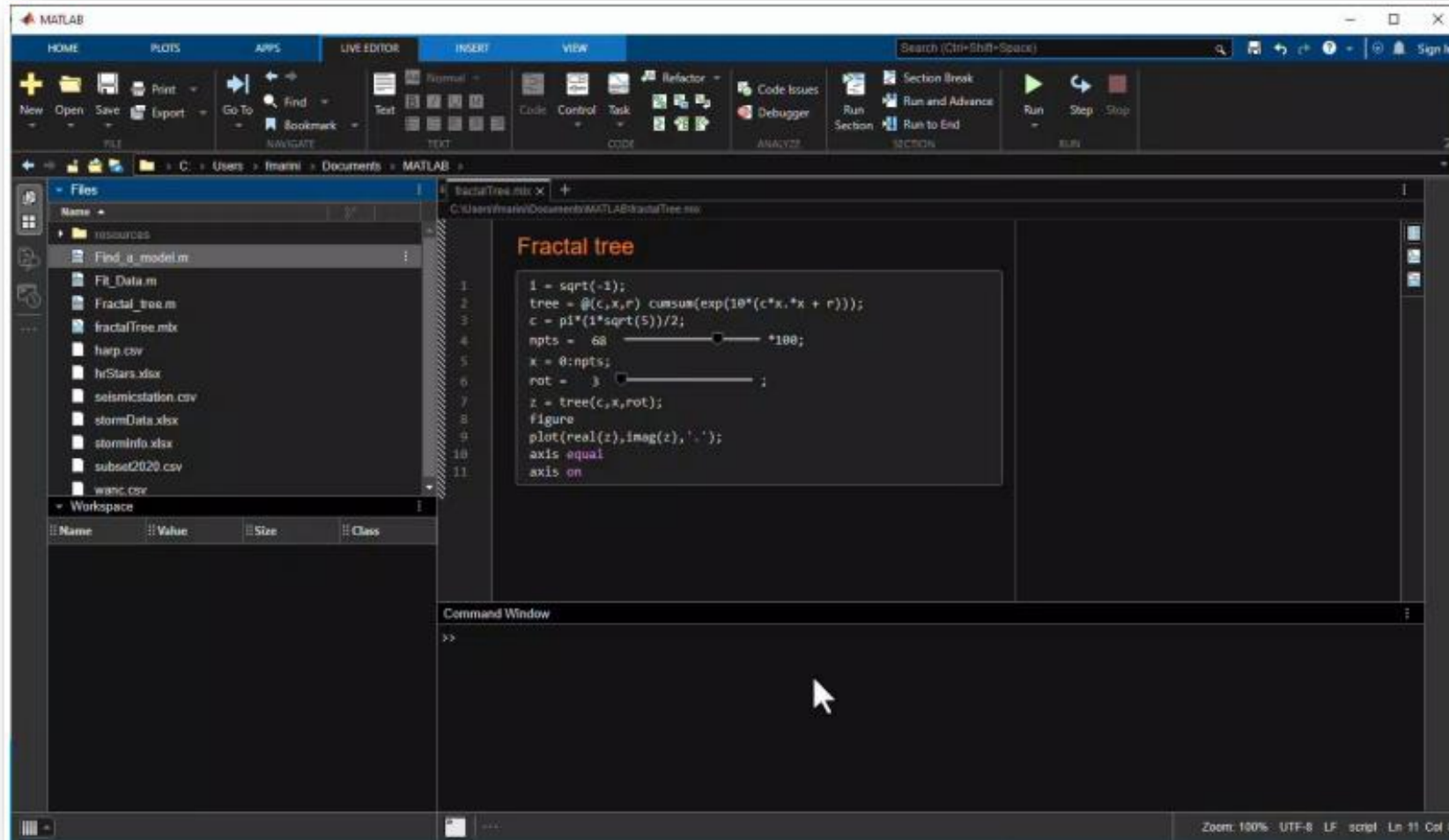
- Logarithmic scale

- Secondary label



MATLAB - New Desktop (Beta)

New Desktop is in MATLAB Online and Public Beta

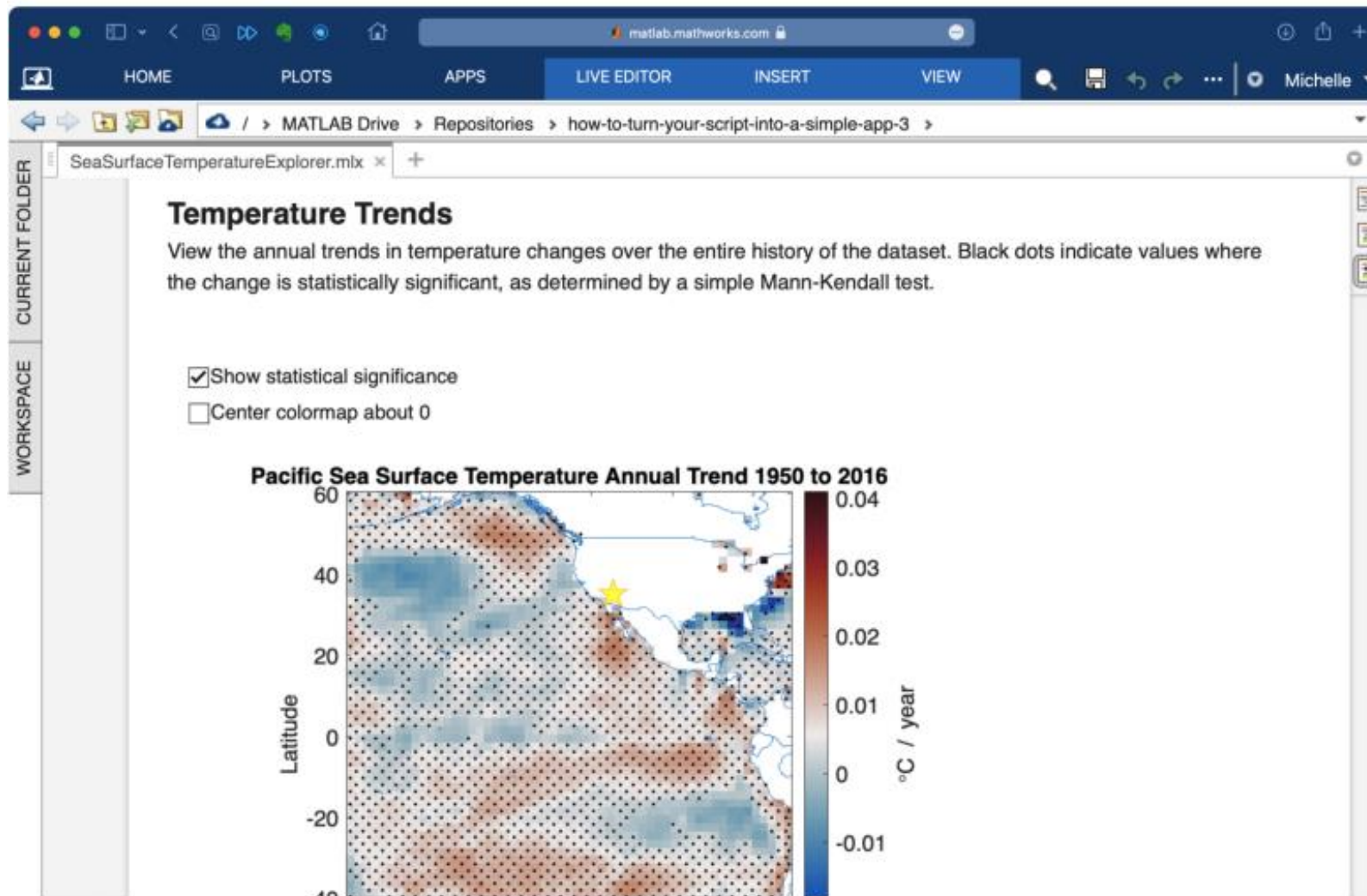


- Key features include:
- Dark theme
 - Layout enhancements
 - Figure toolstrip – etc.
 - Available to anyone
 - Found via:
 - Add-On Manager
 - File Exchange

[File Exchange - dark theme](#)

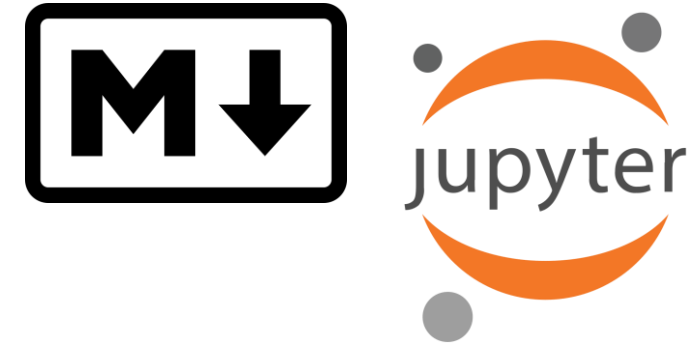
MATLAB

Activity on GitHub and FileExchange continues to grow



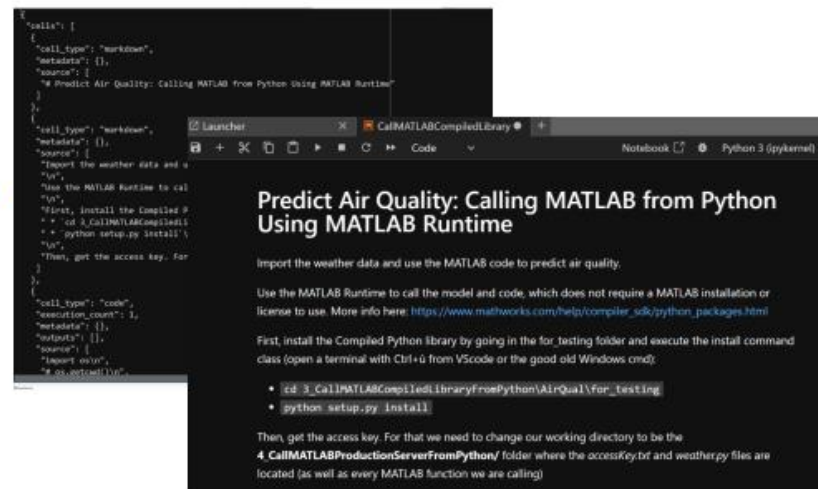
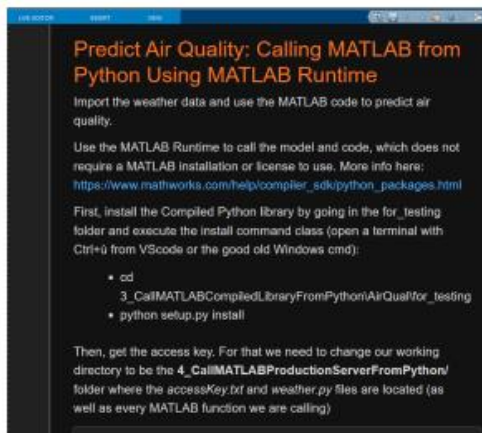
MATLAB

Continue to make it easier to configure, convert data, and use 3rd party environments



- Ability to run MATLAB directly in Jupyter
 - MATLAB kernel for Jupyter
 - Initially on Linux; Windows support recently added
- Convert live scripts to Jupyter notebooks

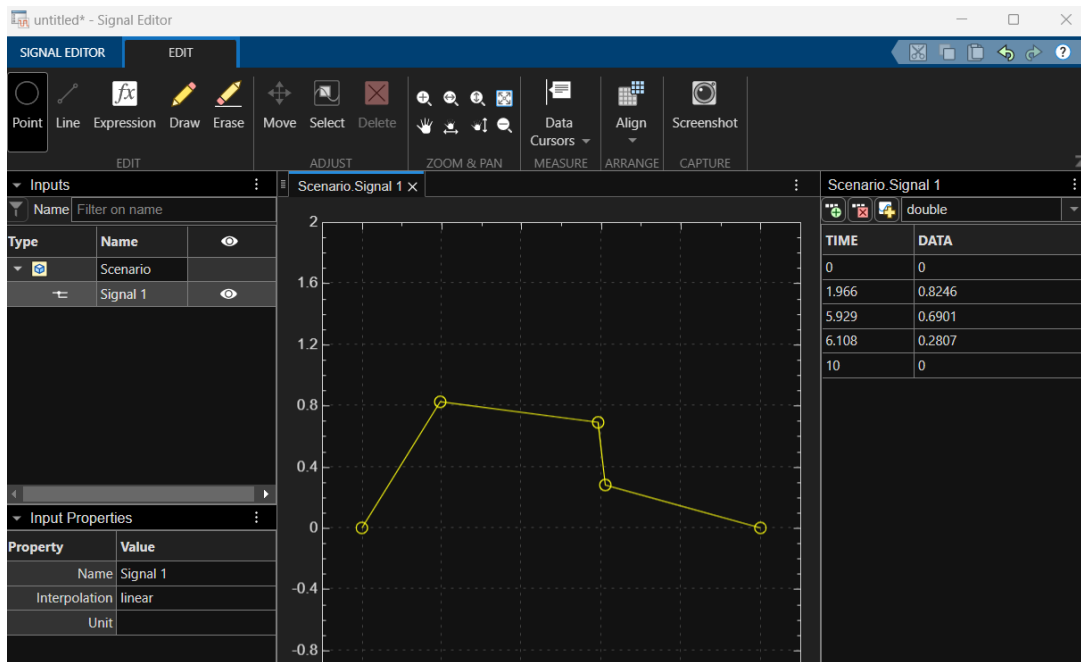
```
export("penny.mlx", "penny.md")
export("penny.mlx", "penny.ipynb")
```



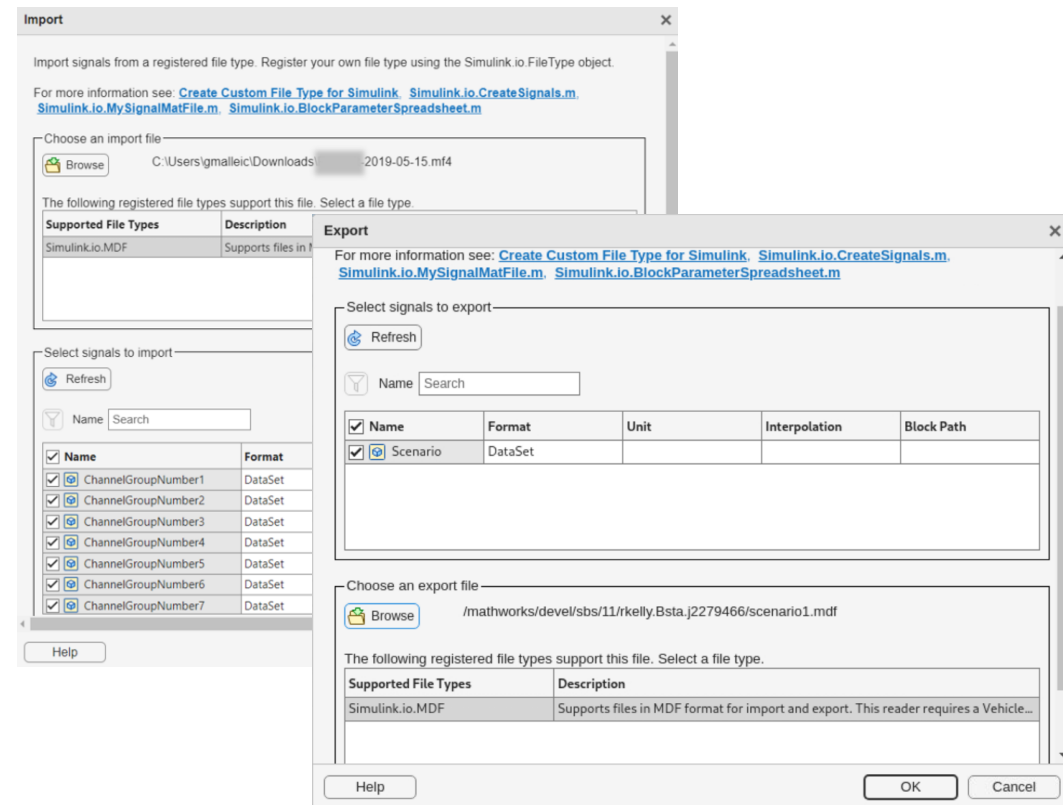
Simulink

Signal Editor: UI

- Cut, Copy, Paste signal data from Microsoft Excel to Signal Editor data table.
- Import MDF files in the Signal Editor or at the command line with Simulink.io.MDF



Signal Editor



Simulink.io.MDF class

Simulink

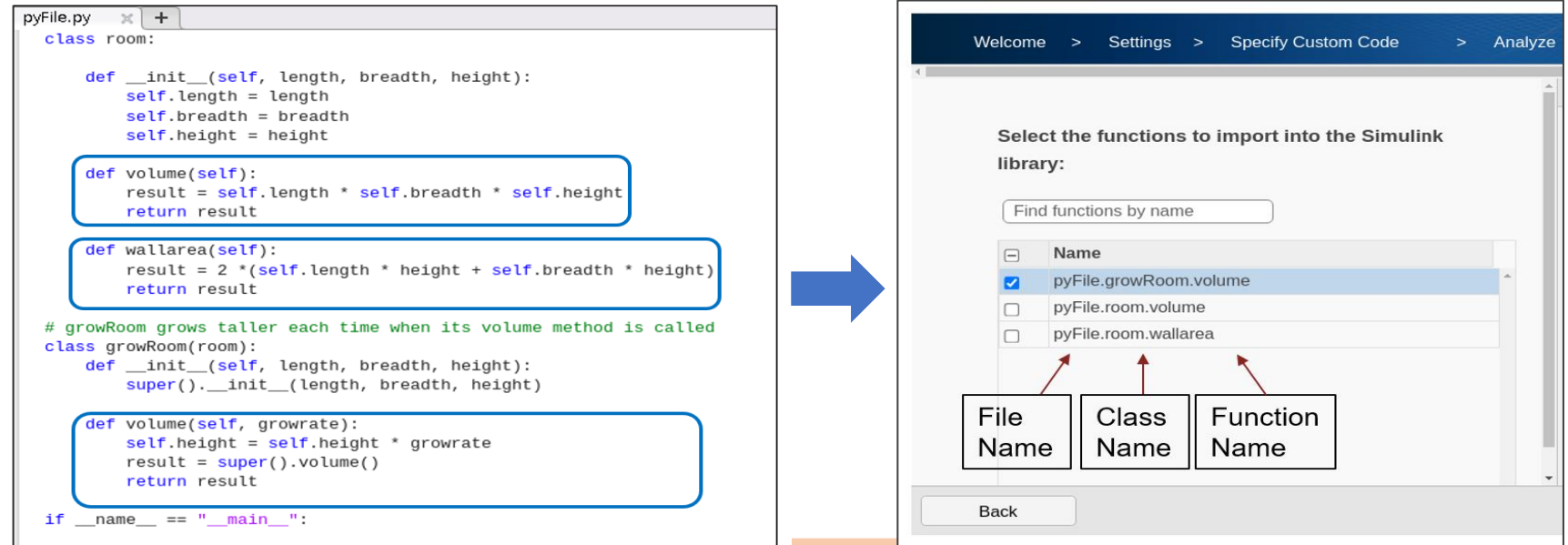
Interact with Git from MATLAB Command Line

Interact with Git source control programmatically

Source Control API	Git Command
gitclone	git clone
gitrepo	git repo
gitinit	git init
createBranch, deleteBranch, switchBranch	git branch, git branch -d, git switch
add, rm, commit	git add, git rm, git commit
log	git log
status	git status
fetch or pull	git fetch, git pull
merge	git merge
push	git push

Simulink

Integrate Python functions defined within Python classes for simulation using Python Importer wizard



The image shows a Python code editor on the left and the Simulink Python Importer wizard on the right. A blue arrow points from the code to the wizard.

Python Code (pyFile.py):

```

class room:
    def __init__(self, length, breadth, height):
        self.length = length
        self.breadth = breadth
        self.height = height

    def volume(self):
        result = self.length * self.breadth * self.height
        return result

    def wallarea(self):
        result = 2 * (self.length * height + self.breadth * height)
        return result

# growRoom grows taller each time when its volume method is called
class growRoom(room):
    def __init__(self, length, breadth, height):
        super().__init__(length, breadth, height)

    def volume(self, growrate):
        self.height = self.height * growrate
        result = super().volume()
        return result

if __name__ == "__main__":
  
```

Simulink Python Importer Wizard:

Welcome > Settings > Specify Custom Code > Analyze

Select the functions to import into the Simulink library:

Find functions by name

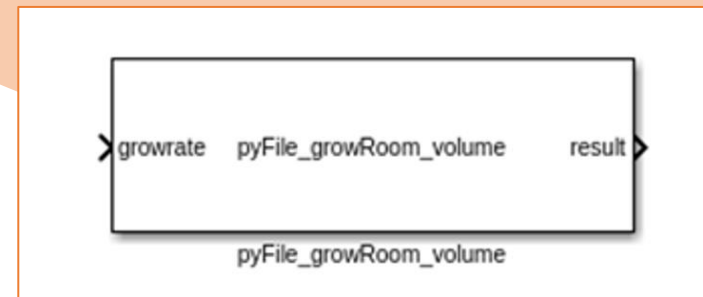
Name
<input checked="" type="checkbox"/> pyFile.growRoom.volume
<input type="checkbox"/> pyFile.room.volume
<input type="checkbox"/> pyFile.room.wallarea

File Name Class Name Function Name

Back

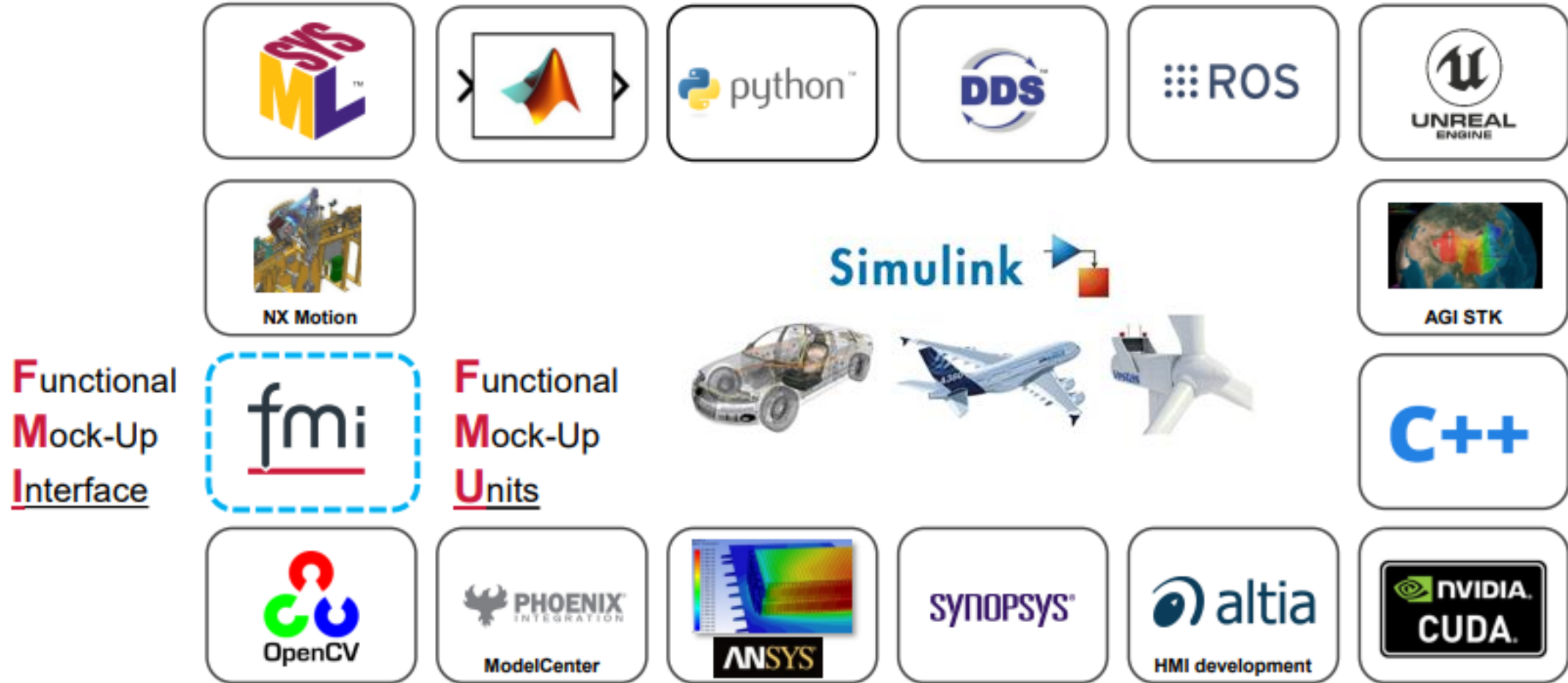
%Launch Python Importer from Command line:

```
obj = Simulink.PythonImporter();
obj.view();
```



Simulink

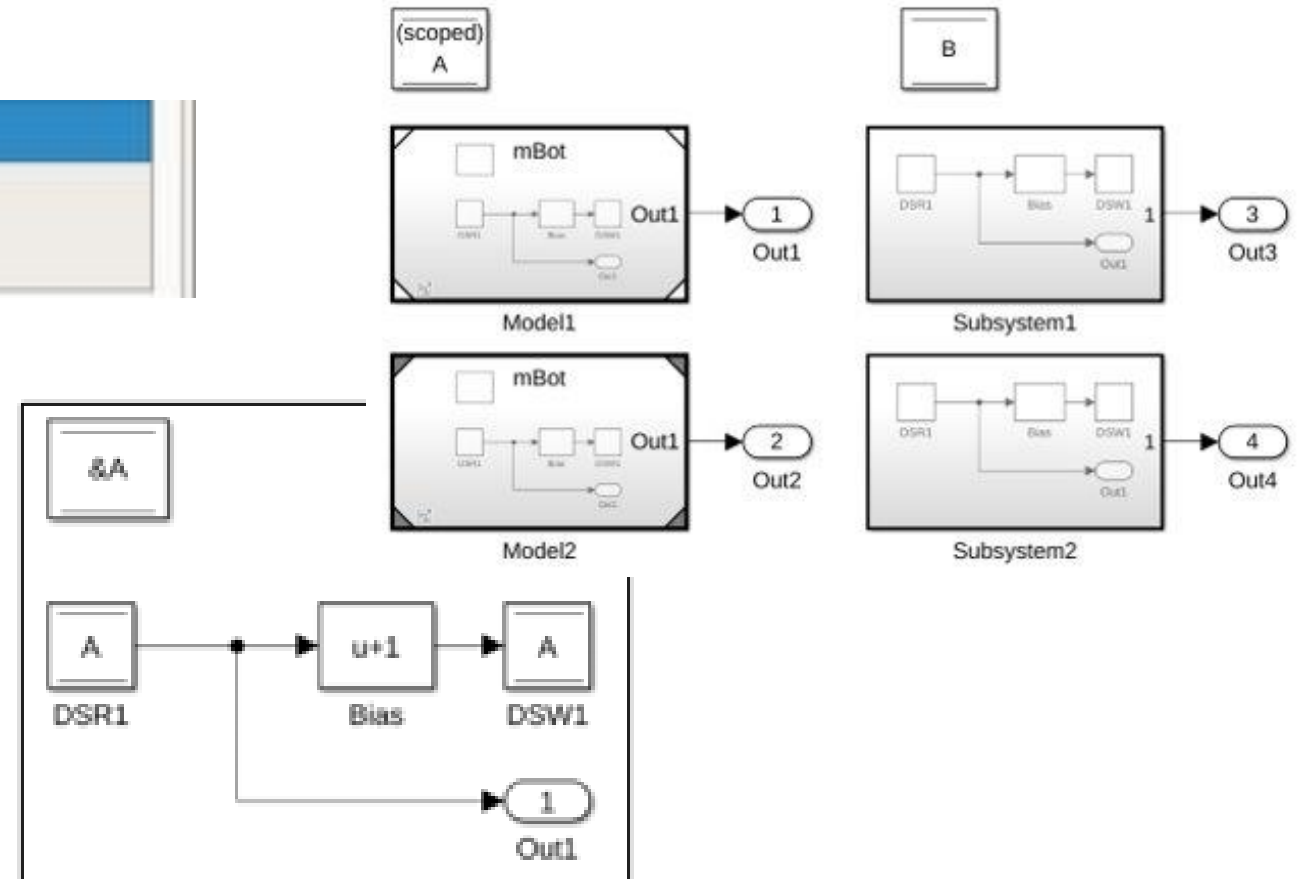
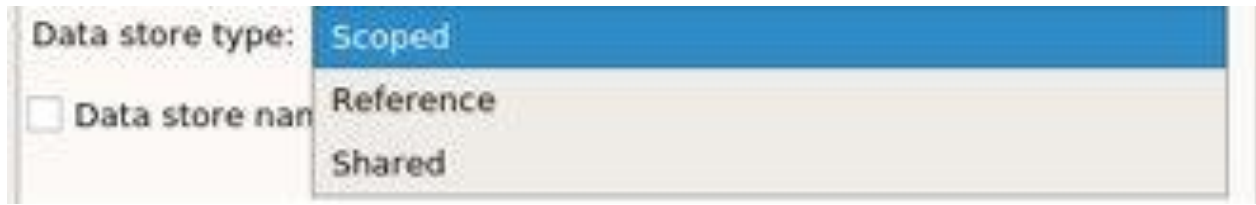
Simulink is a simulation integration platform



Simulink

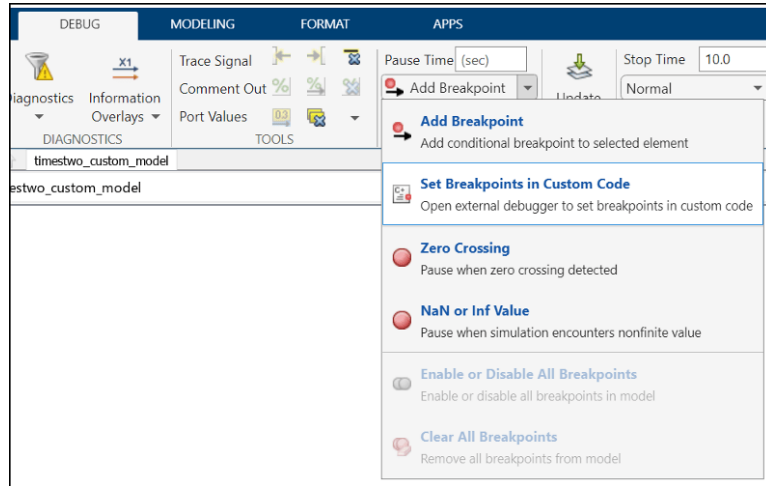
Scoped Data Store Memory

Use Scoped Data Store Memory blocks to extend the visibility of local Data Stores across child model boundaries

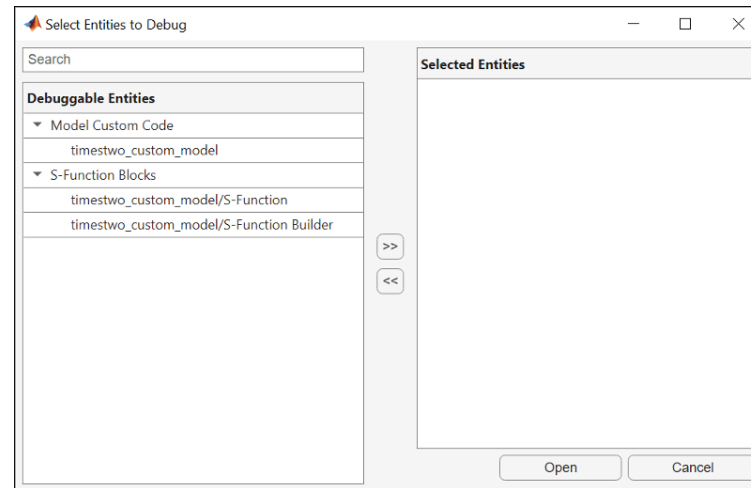


Simulink

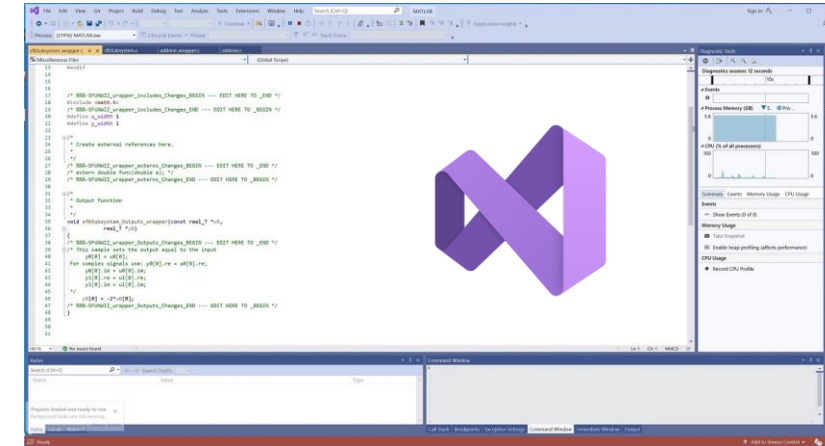
S-Function External Debugger



Open the dialog by clicking the button from toolstrip



Select blocks to debug



Set breakpoint in the external debugger and simulate the model

Note:

Linux – GDB with DDD

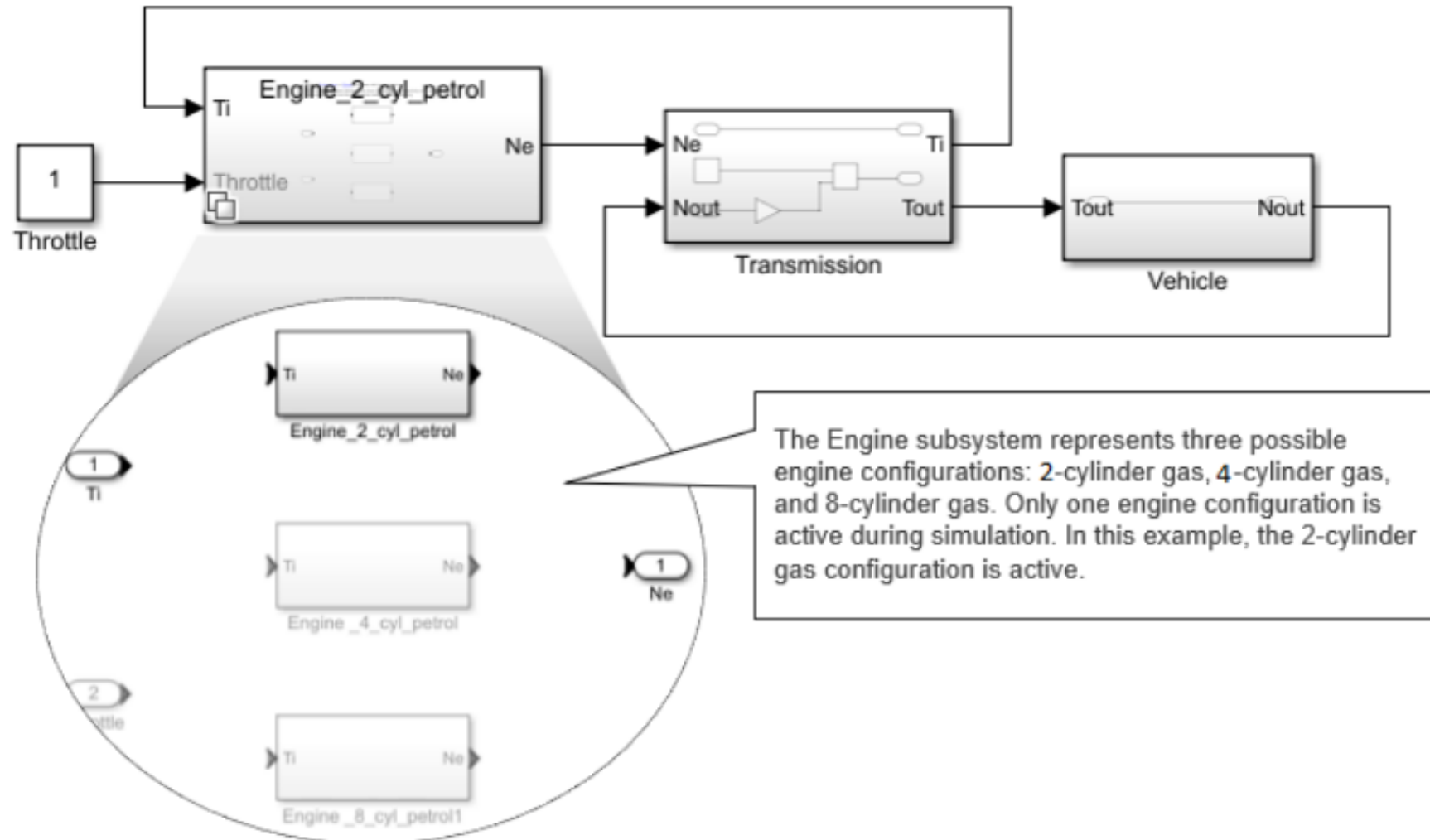
Mac – LDB from Terminal

Windows – Visual Studio, VS Code planned

Simulink

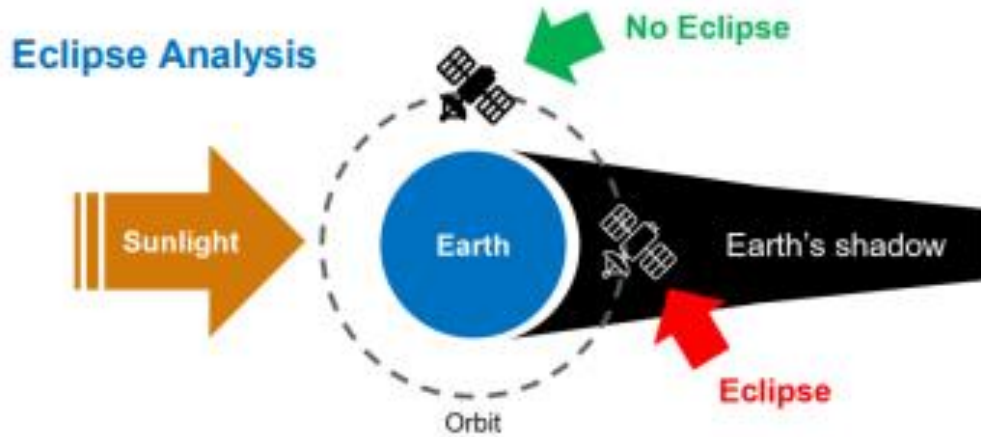
Variant Navigation

Multiple Implementations of a Single System



Aerospace Toolbox

Propagate and Visualize Orbits of Satellite Constellations;
 Perform Line-of-sight (LOS) and Eclipse Analyses



Calculate percentage of time when solar occultation occurs

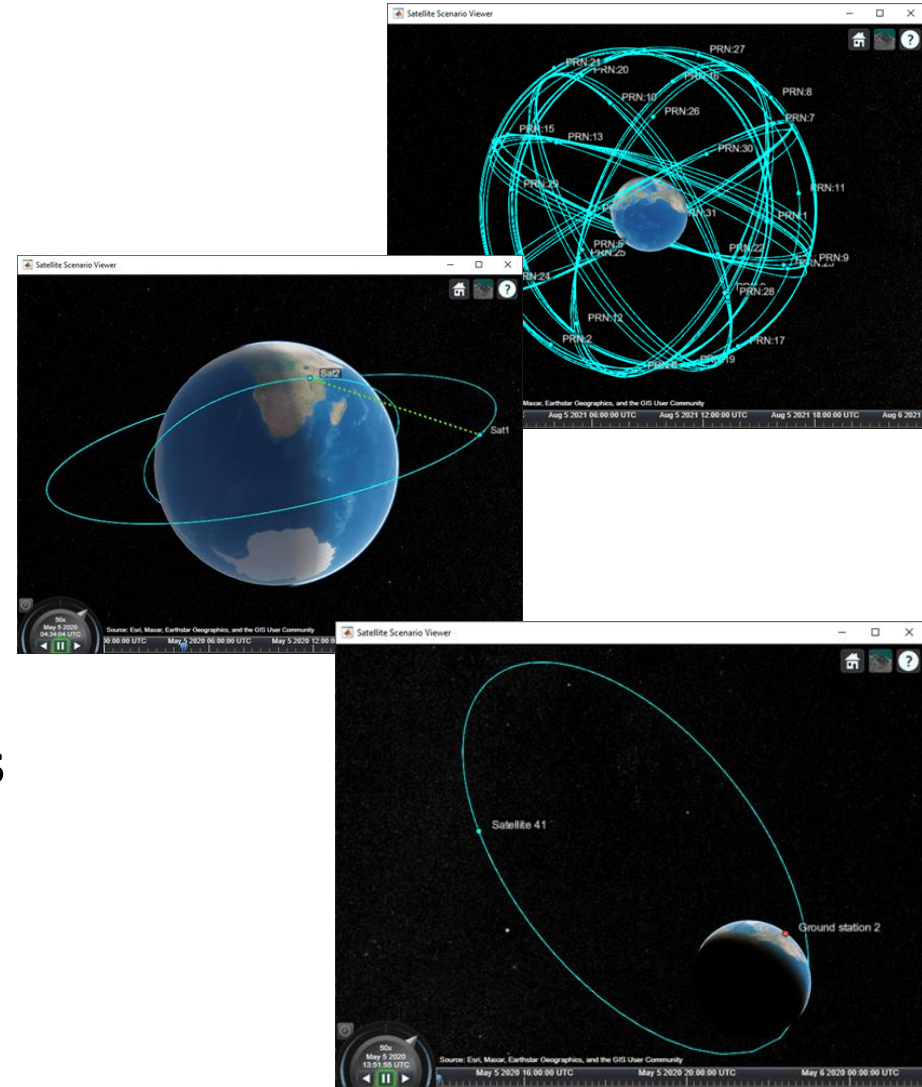
```
eclPercent = eclipsePercentage(eclipse_objects)
```

Calculate intervals of solar occultation

```
interval_table = eclipseIntervals(eclipse_objects)
```

Calculate status of solar occultation

```
status = eclipseStatus(eclipse_objects)
```



[Aerospace Toolbox Release Notes](#)

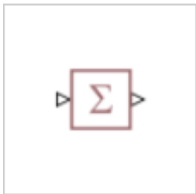
Simscape

New Blocks - physical signal



PS Bias

Add constant bias to input physical signal



PS Concatenate

Concatenate two vector or matrix physical signals



PS Sum of Elements

Calculate sum of elements of vector or matrix input physical signal

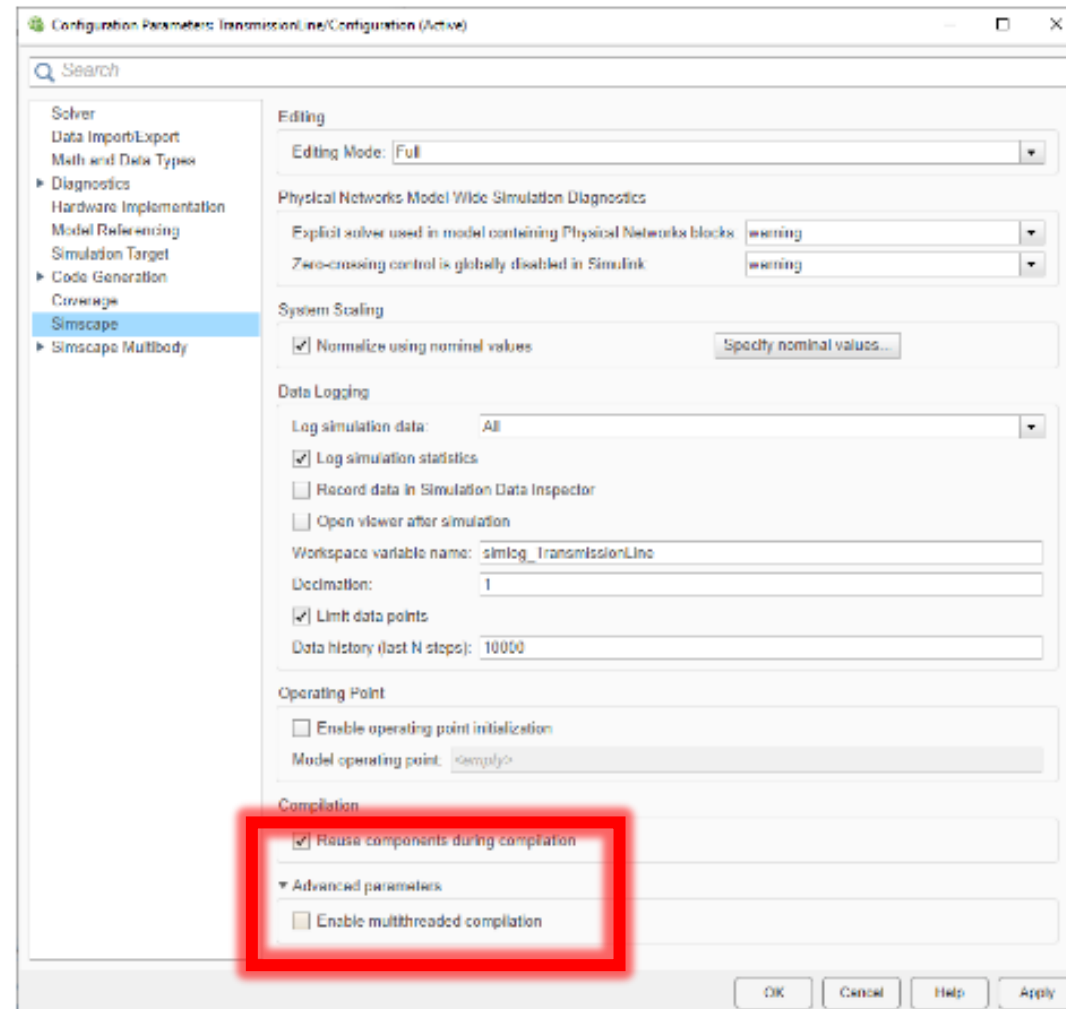


PS Selector

Select elements from vector or matrix input physical signal

Simscape

Reduce compilation



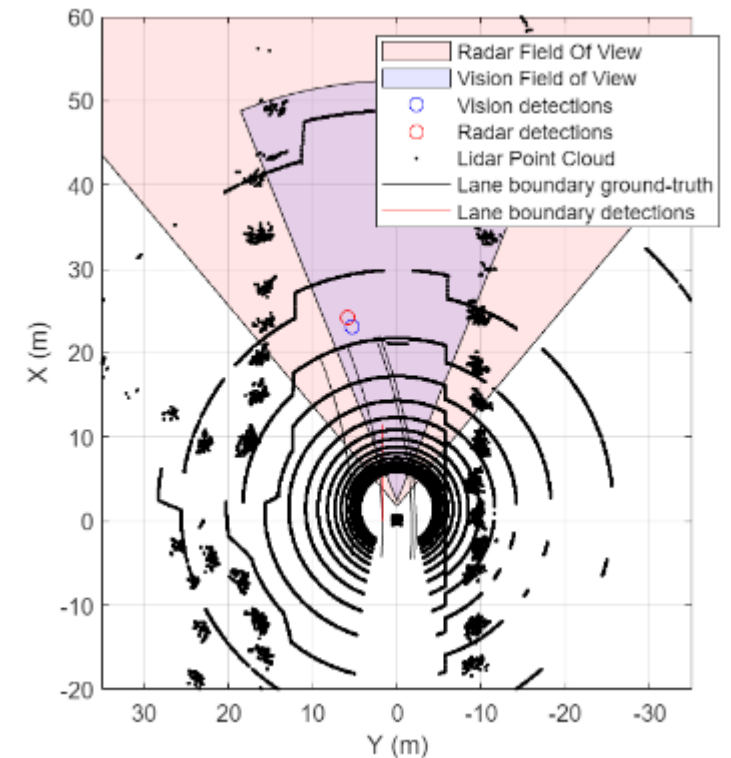
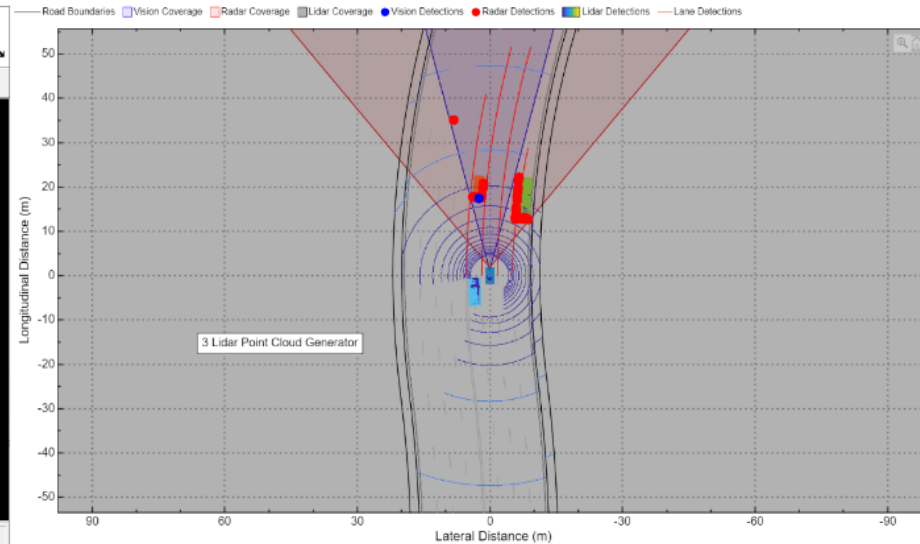
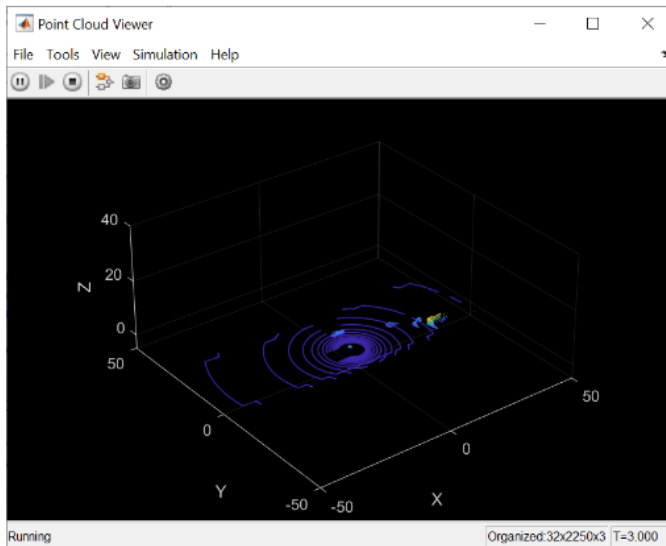
```
set_param(bdroot, 'SimscapeCompileComponentReuse', 'on')
```

[Simscape Release Notes](#)

Automated Driving Toolbox

Sensor Simulation for RoadRunner Scenario Enhancements

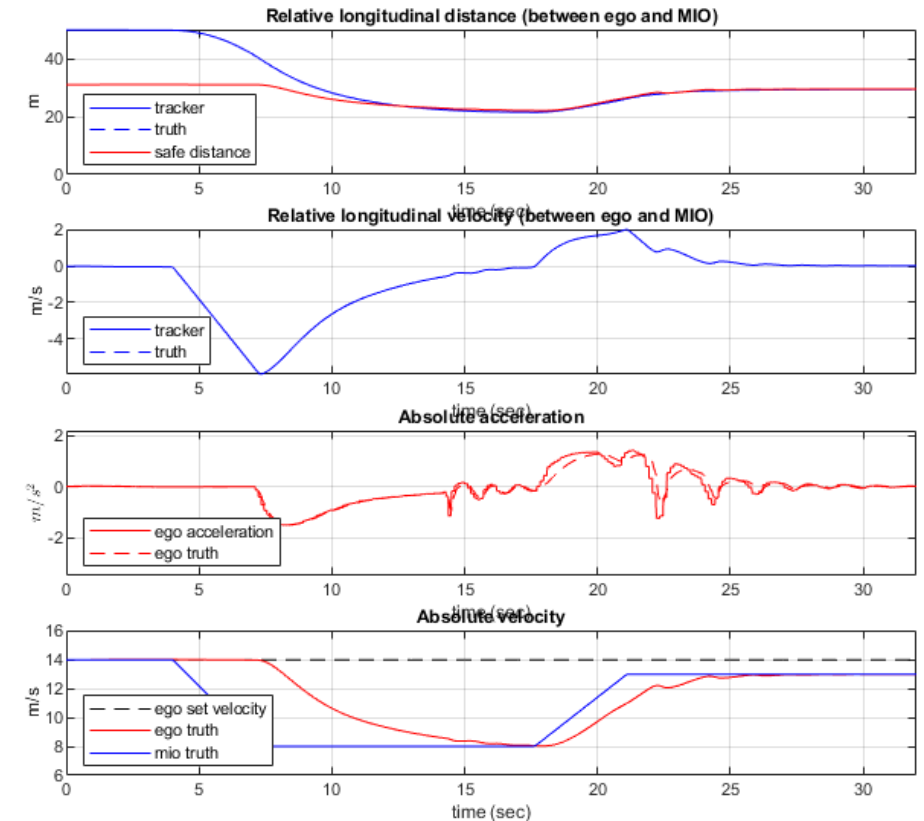
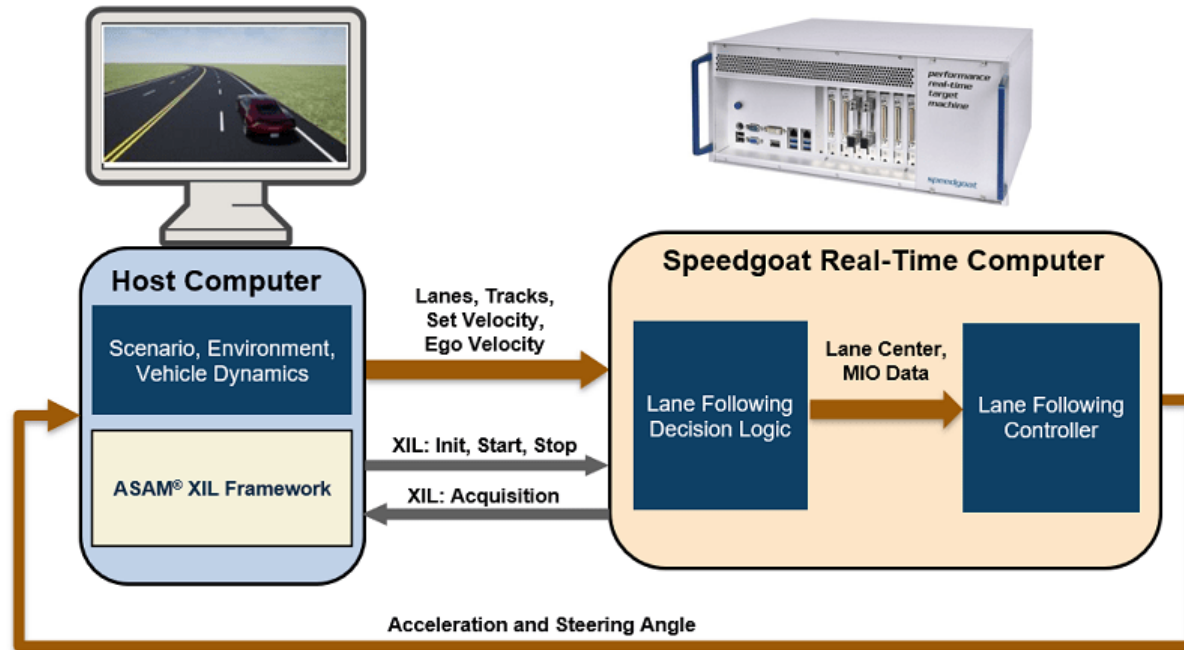
Simulate **RoadRunner Scenario** with **lidar sensor model** defined in Simulink, and use Bird's-Eye Scope to visualize sensor measurements from RoadRunner Scenario.



[Automated Driving Toolbox Release Notes](#)

Automated Driving Toolbox

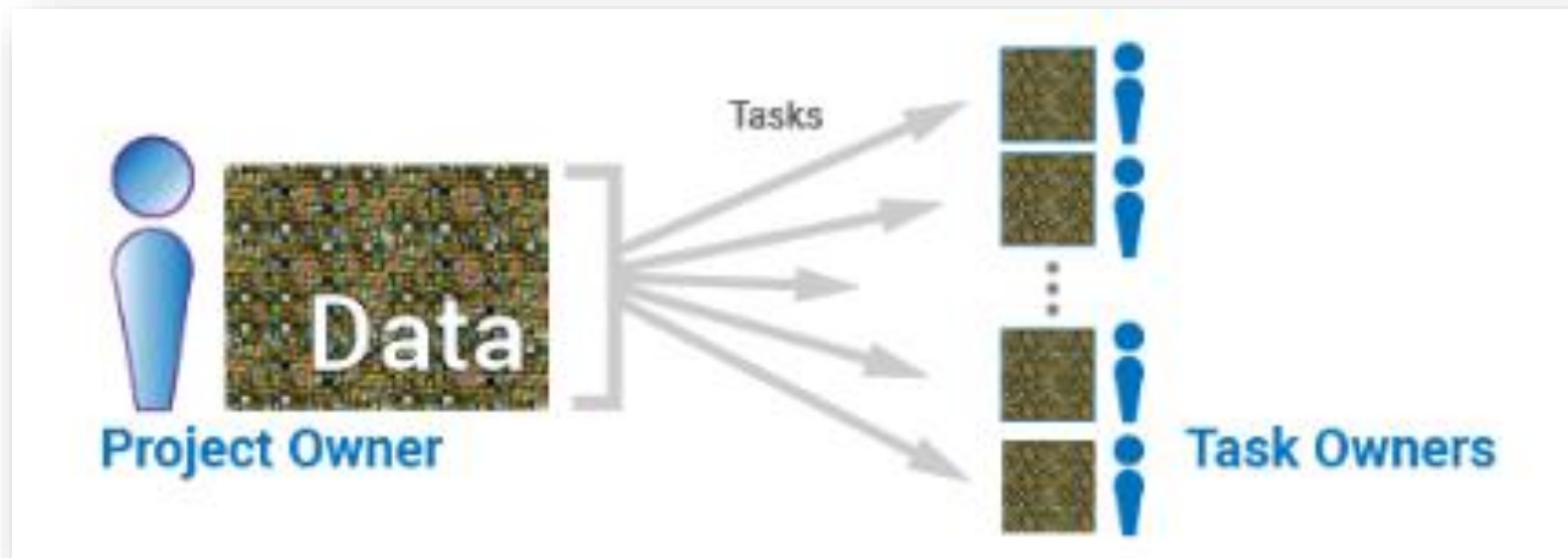
Automate Real-Time Testing of Highway Lane Following Controller Using ASAM XIL



[Automate Real-Time Testing](#)

Computer Vision Toolbox

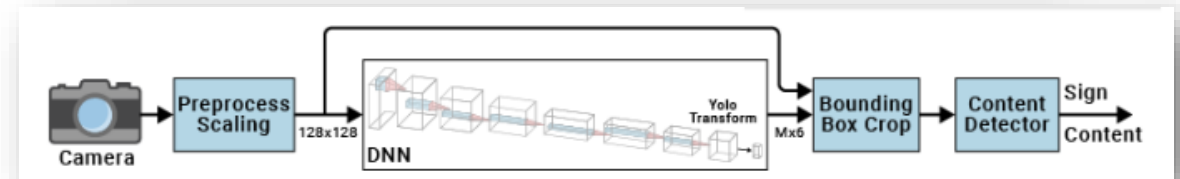
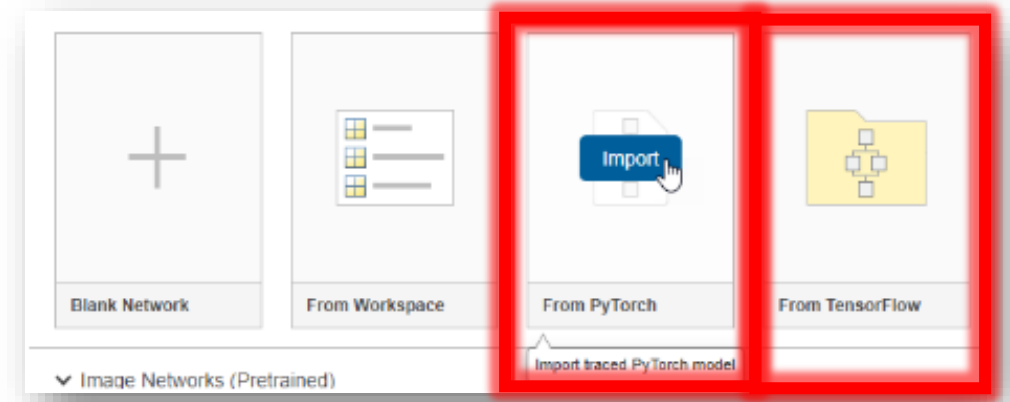
Detect objects in image using YOLOX deep learning network; Collaborative team labeling



```
detector = yoloxObjectDetector
```

Deep Learning Toolbox

PyTorch[®] ; TensorFlow[™]; New examples



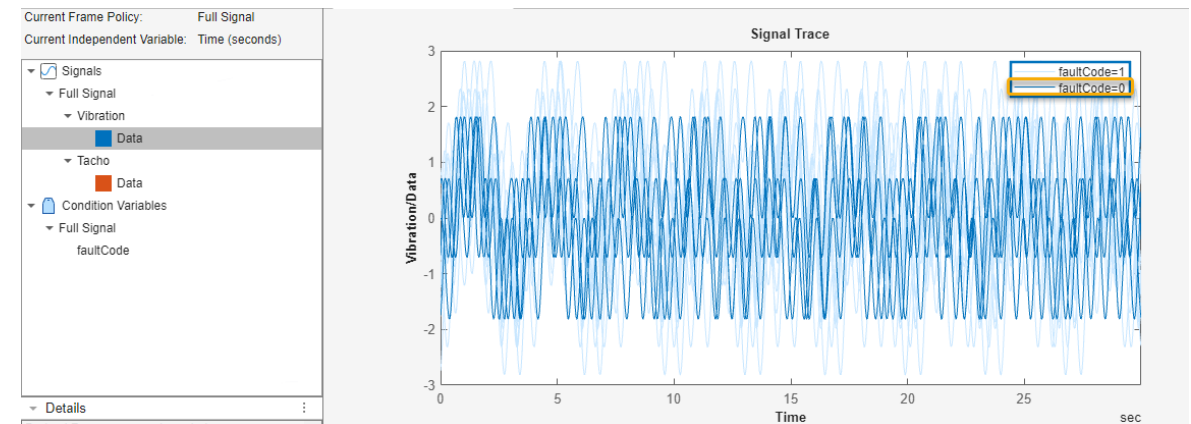
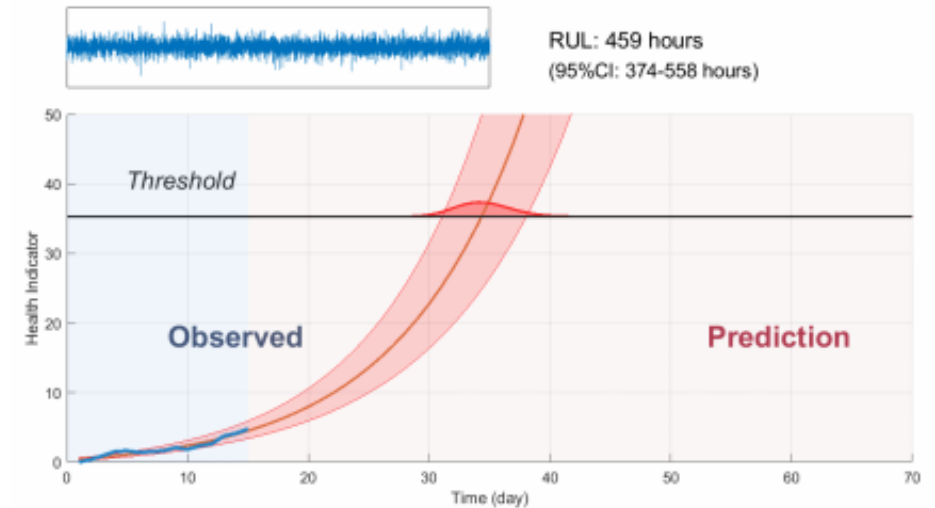
- Import models from PyTorch[®] and TensorFlow[™]
- [Verify airborne deep learning system](#) (example)
- New and updated examples and topics
 ([Learning Toolbox — Examples](#))

- [Generate Images Using Diffusion](#)
- [Import PyTorch[®] Model Using Deep Network Designer](#)
- [Train Neural Network with Tabular Data](#)
- [Deep Learning Data Formats](#)
- [Train Smaller Neural Network Using Knowledge Distillation](#)
- [Create Bidirectional LSTM \(BiLSTM\) Function](#)
- [Train Neural ODE Network](#)
- [Compress Network for Estimating Battery State of Charge](#)
- [Convert Convolutional Network to Spiking Neural Network](#)
- [Export Network to FMU](#)
- [Create Custom Deep Learning Training Plot](#)
- [Custom Stopping Criteria for Deep Learning Training](#)

Predictive Maintenance Toolbox

Design and test condition monitoring and predictive maintenance algorithms

- Import time-domain data
- Identify harmonics and sidebands automatically
- Perform batch processing for data processing and feature extraction
- Delete features
- Distinguish signal conditions using legend box



Simulink Desktop Real-Time

Execute Simulink Models in Real-Time on Linux Computers

- **Closed-loop control of hardware devices** from your desktop computer running **Linux**



Release Notes

The screenshot shows the MathWorks Help Center interface. At the top, there is a navigation bar with the MathWorks logo and links for Products, Solutions, Academia, Support, Community, and Events. A search bar is located on the right side of the top bar. Below the top bar, the 'Help Center' title is displayed on the left, and a search bar with the text 'Search Help Center' is on the right. The main content area is divided into several sections. On the left, there is a 'CONTENTS' sidebar with a 'Category' list including MATLAB, Simulink, 5G Toolbox, Aerospace Blockset, Aerospace Toolbox, and Antenna Toolbox. The main content area has a top navigation bar with links for Documentation, Examples, Functions, Blocks, Apps, Videos, and Answers. Below this, the 'Release Notes' section is highlighted with a blue box. It contains the text 'Release Notes' in orange, 'Explore Release Notes' in bold, and a 'Select a Product' dropdown menu. To the right of the 'Release Notes' section, there is a 'Resources' section with links for 'Installation and Licensing Changes', 'System Requirements', 'Bug Reports', and 'Bug Fixes'. The version 'R2023b' is displayed in the top right corner of the main content area.

[Release Notes](#)





Oprogramowanie
Naukowo-Techniczne
sp. z o.o.

www.ont.com.pl



matlab.pl



oprogramowanie-
naukowo-techniczne



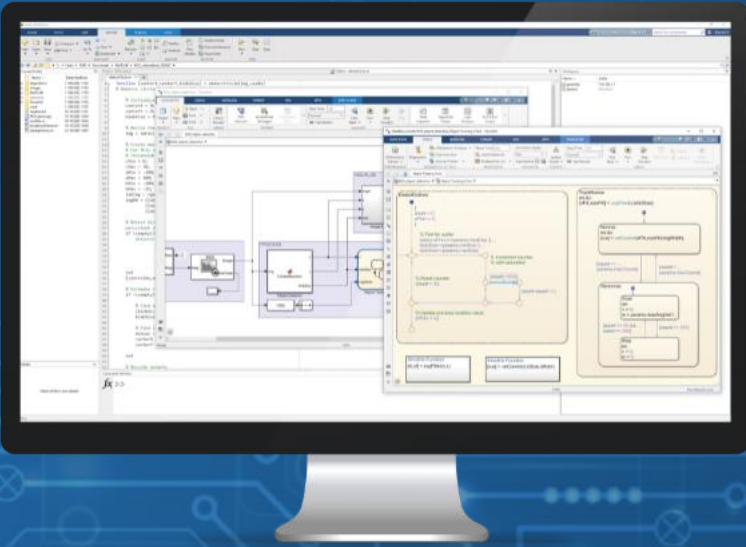
ONT MATLAB



Paulina Kozakiewicz

Junior Application Engineer, ONT

paulina.kozakiewicz@ont.com.pl



APPLICATIONS

- ▶ Robotics and Automation
- ▶ Computational Finance
- ▶ Autonomous Vehicles
- ▶ Electronics
- ▶ Artificial Intelligence
- ▶ Biomedical Engineering
- ▶ Systems Engineering and certification
- ▶ Power Electronics and Systems
- ▶ Communications and Radar Systems

Let's stay in touch

Oprogramowanie Naukowo-Techniczne sp. z o.o.
MATLAB and Simulink authorised reseller for Poland
ul. Pod Fortem 19, 31-302 Kraków, Poland | www.ont.com.pl