

# MATLAB EXPO

2024年5月30日 | 東京

## MATLAB and Simulink 最新情報

*MathWorks Japan*

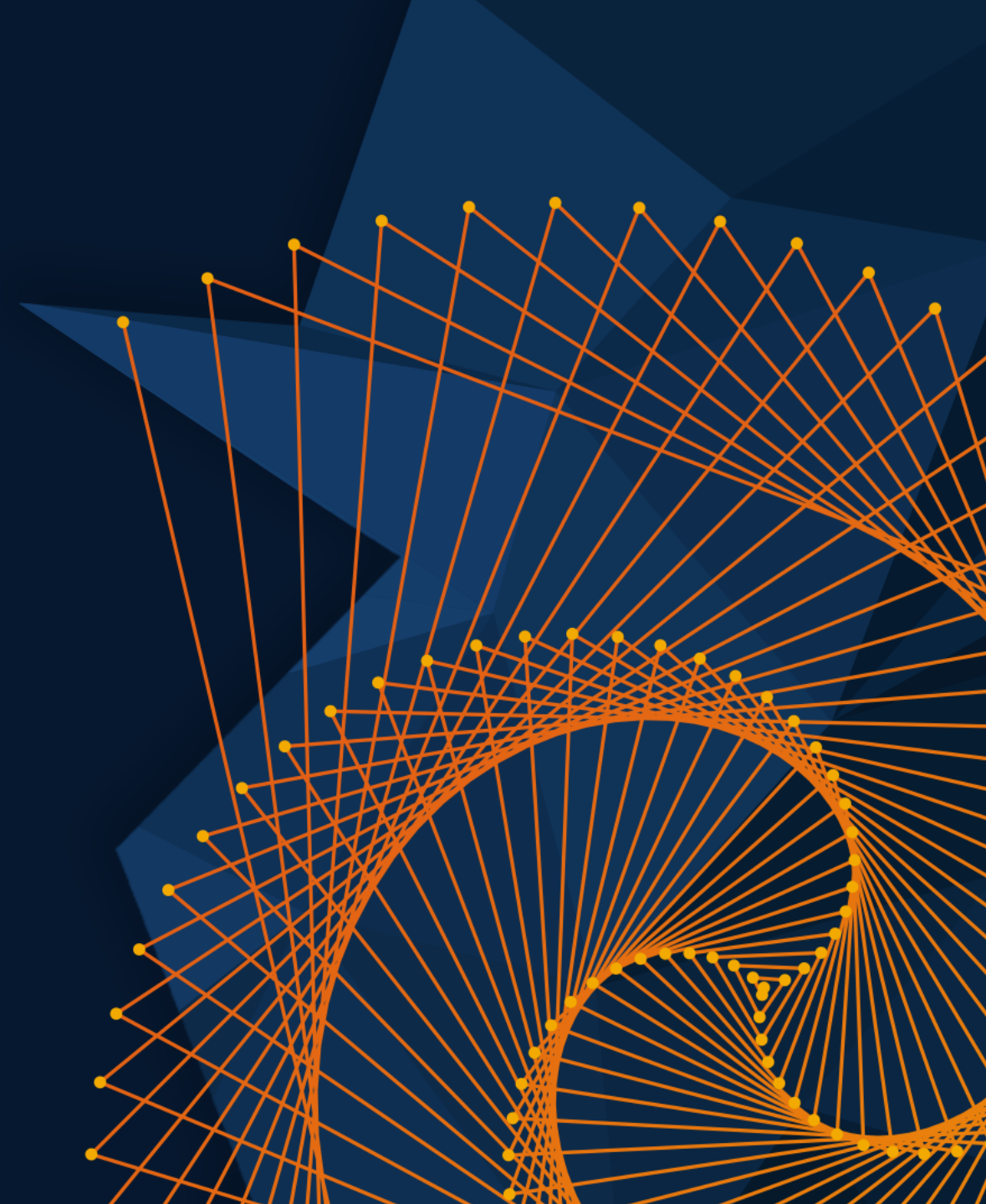


上野 敬志

*MathWorks Japan*



岩本 光平



3,744



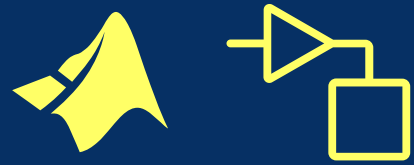
MATLAB®  
& SIMULINK®



**Integrations**



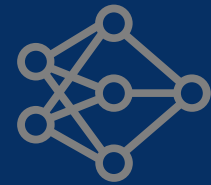
**AI**



MATLAB®  
& SIMULINK®



Integrations



AI



Ease of Use



Performance



Verification

# Local Functions



```
x = 1:10;  
n = length(x);  
avg = mymean(x,n);
```

```
function a = mymean(v,n)  
% MYMEAN Local function  
  
    a = sum(v)/n;  
end
```

# Local Functions: Define functions anywhere in scripts



```
x = 1:10;  
n = length(x);
```

```
function a = mymean(v,n)  
% MYMEAN Local function
```

```
    a = sum(v)/n;  
end
```

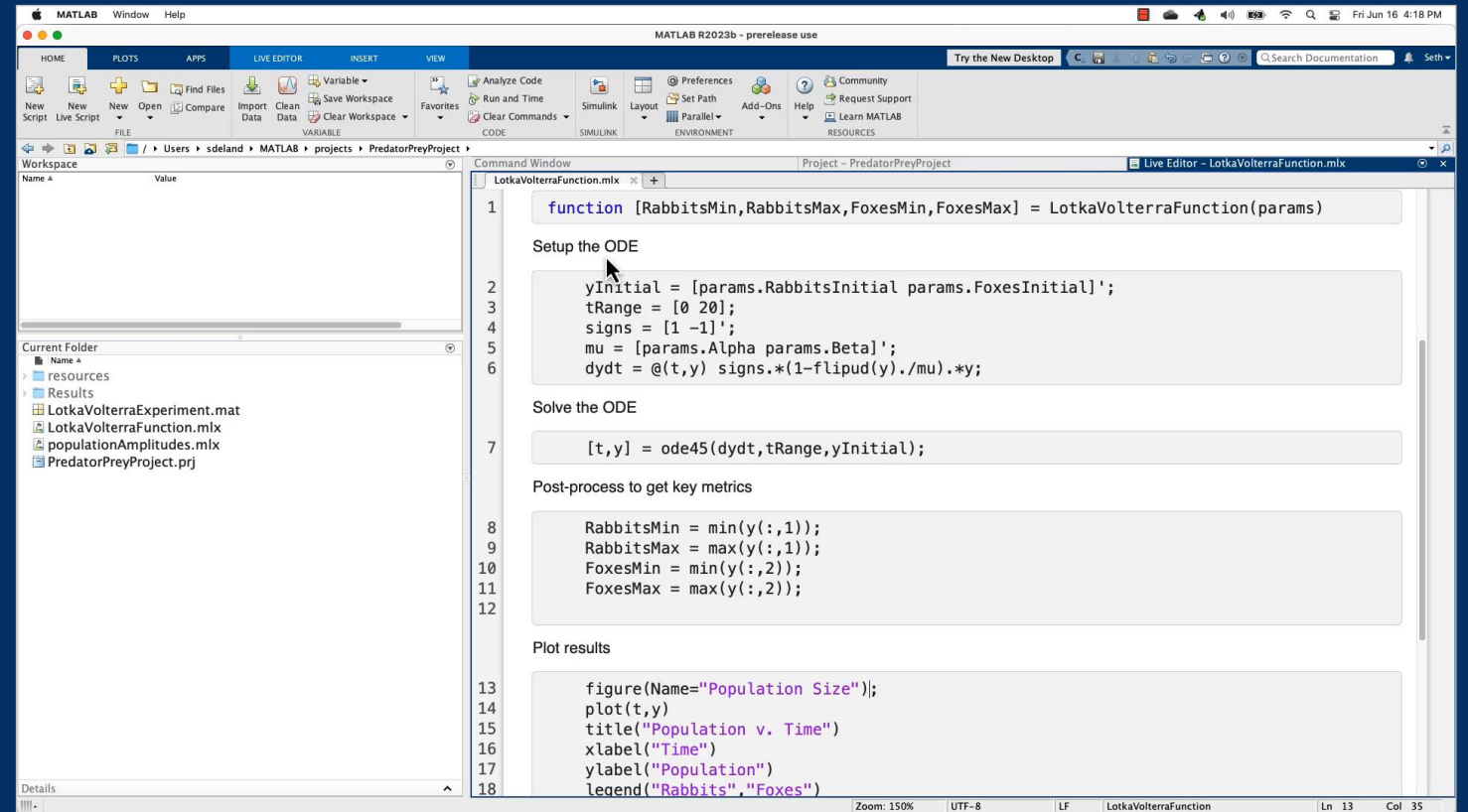
```
avg = mymean(x,n);
```

# Solve problems with little to no coding – *using apps*



Design *experiments* to run MATLAB code

- Visualize, filter, and compare results



**Experiment Manager App**

# Solve problems with little to no coding – *using apps and tasks*

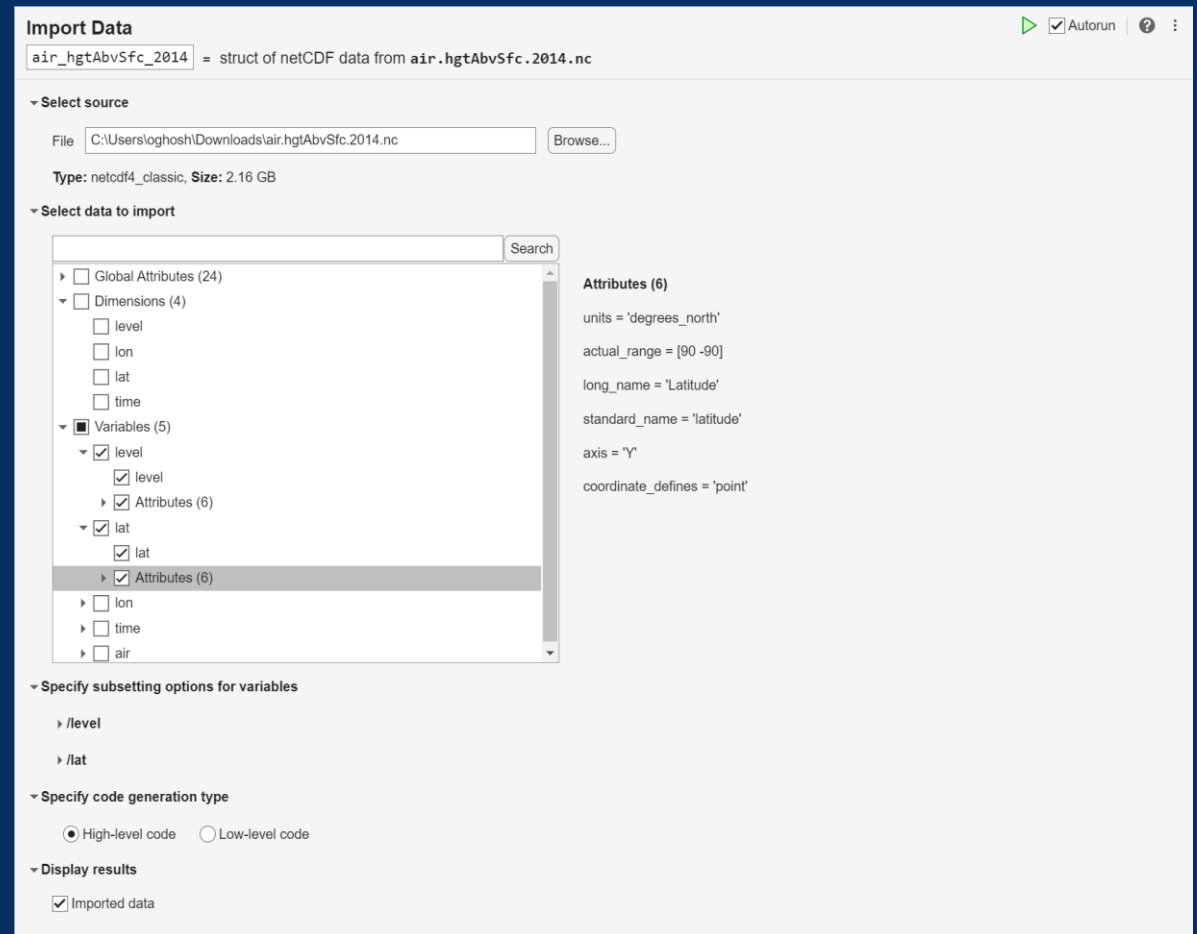


Design experiments to run MATLAB code

- Visualize, filter, and compare results

Interactively visualize and preview hierarchical data formats

- NetCDF
- HDF5



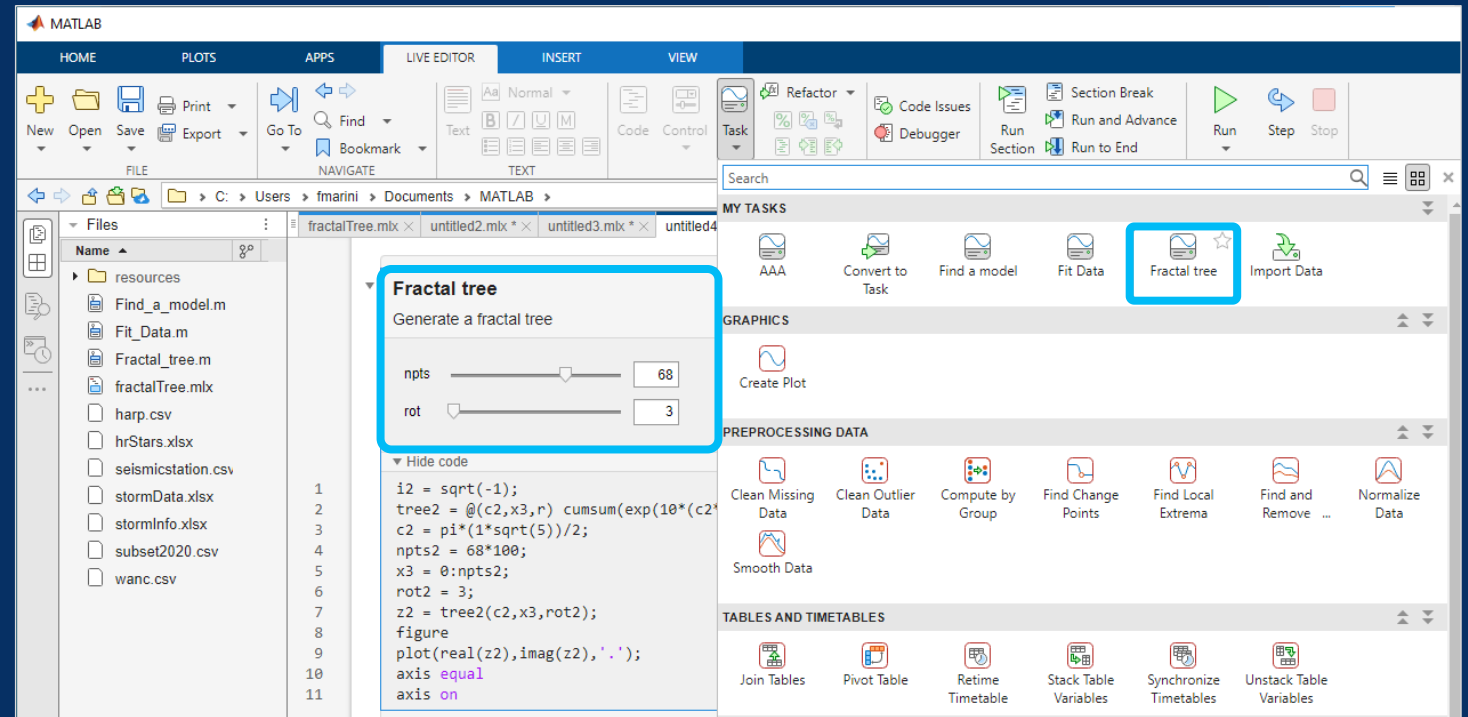
Import Data Live Editor Task



# Easier to create custom Live Editor tasks



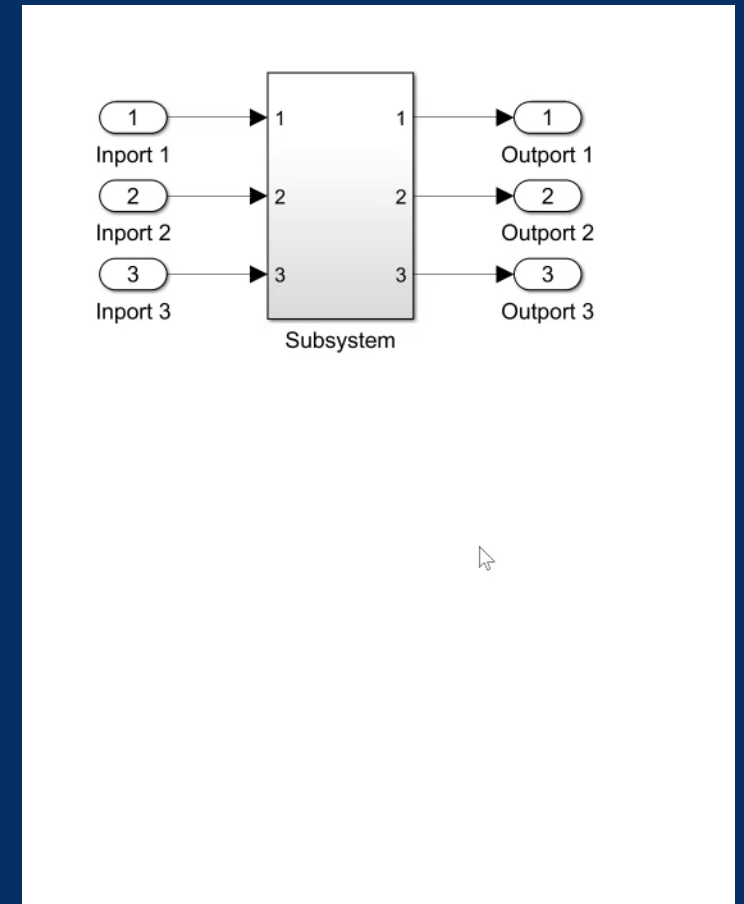
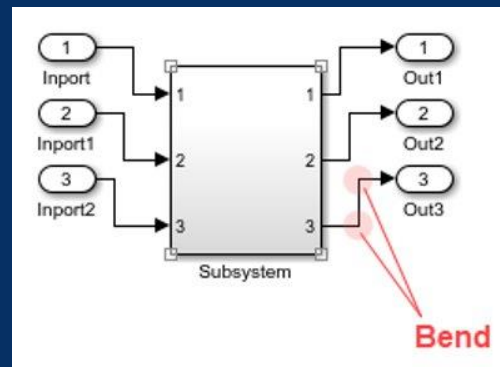
Convert a selection (*code and interactive controls*) into your own Live Editor task



# Edit models at the speed of thought



Automatically  
**preserve signal line  
shape** when moving  
or resizing blocks



# Interactively create, edit, and visualize signal data



Copy, cut, or paste  
**Excel data** to or from  
Signal Editor

The screenshot shows the MATLAB Signal Editor interface with a plot of a signal and a table of data. The plot shows a signal with 10 data points connected by a blue line. The table on the right lists the time and data values for each point.

TIME	DATA
1	0.473813245
2	-0.867088305
3	-4.168979031
4	-7.775208909
5	-9.749319903
6	-9.655416552
7	-8.958926245
8	-9.679292042
9	-12.53330603
10	-16.276094

The Excel spreadsheet on the left shows the same data in a table with columns for Time, Signal1, and Signal2. The formula bar shows the formula for Signal2:  $=\text{SIN}(\text{A2}) * 2.34567 - (1.5 * \text{A2})$ .

# Create interactive simulation controls using dashboards



Dashboard blocks



Group and promote dashboard blocks to a **panel**

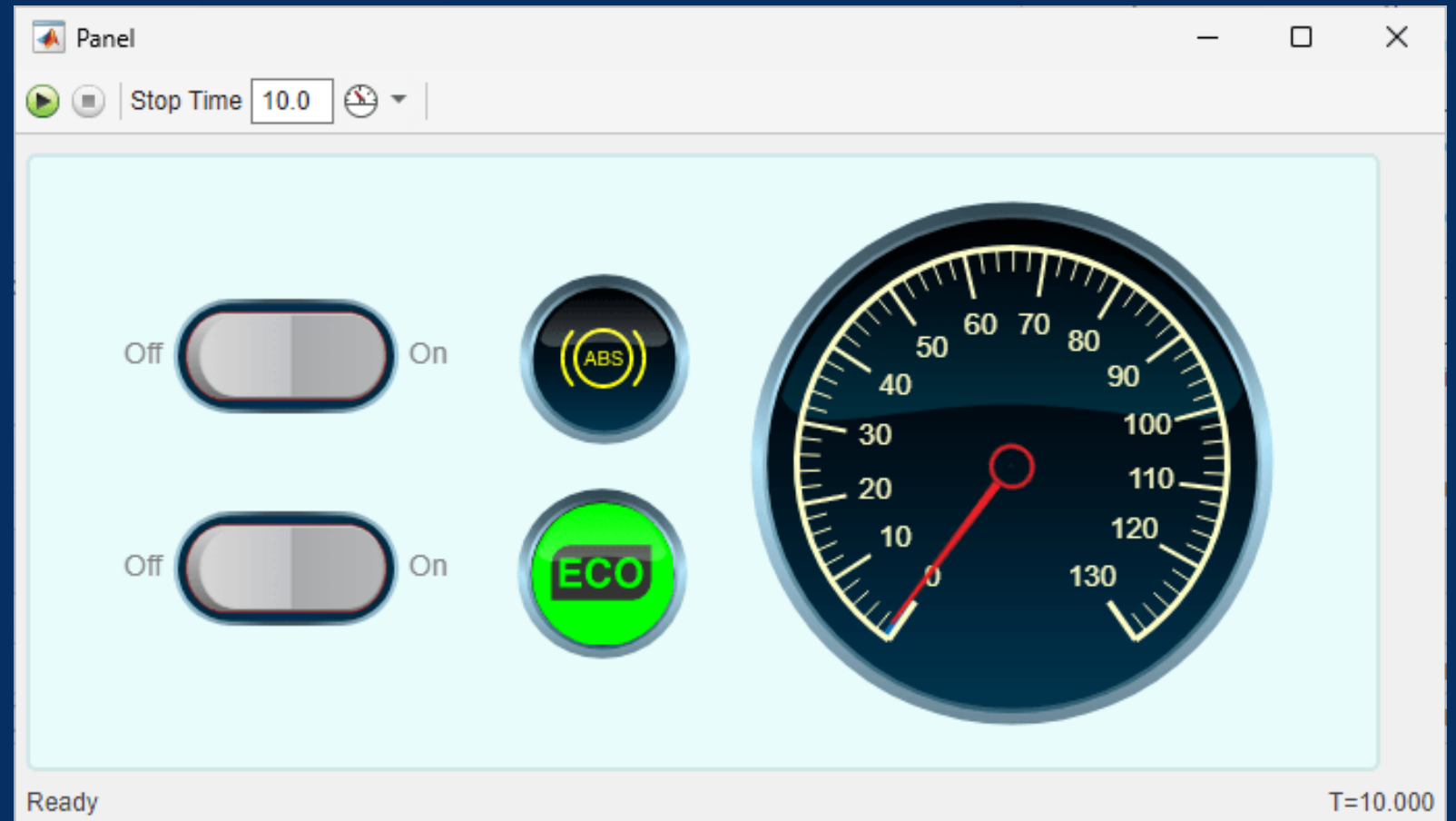


Dashboard panel

# Deploy interactive simulation controls as apps



Convert **dashboard panels** into apps

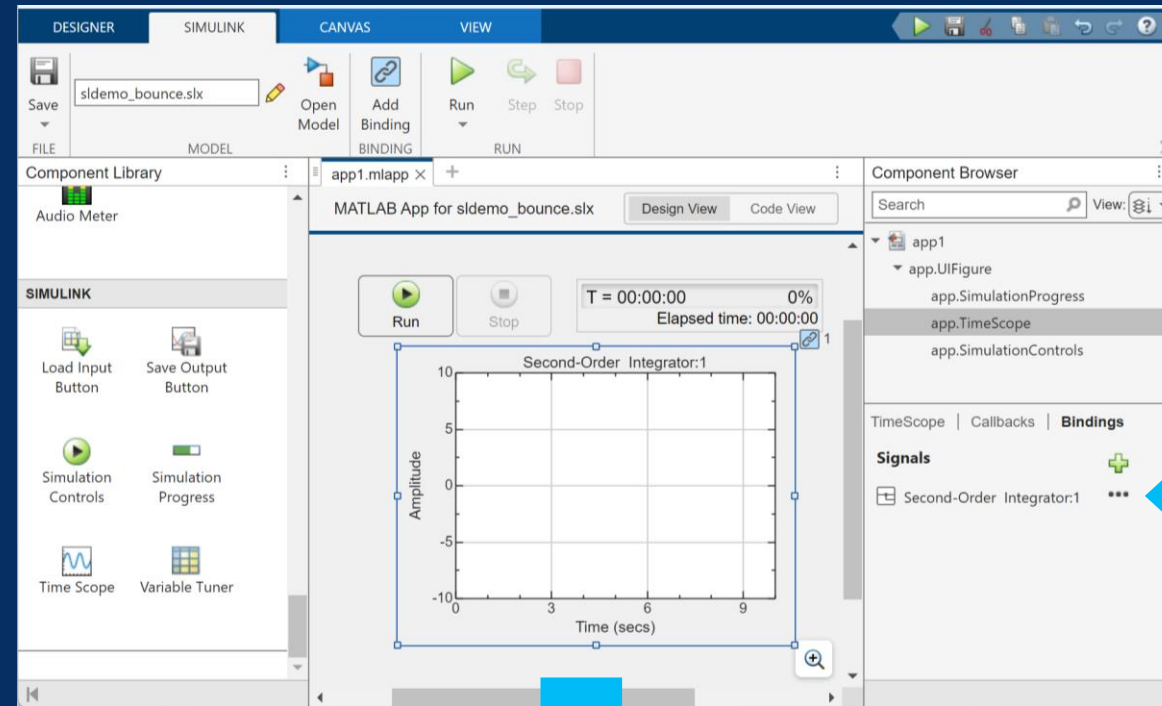


# Design simulation apps without writing code



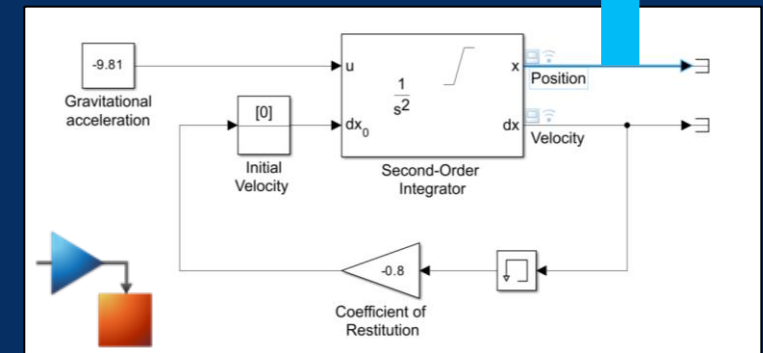
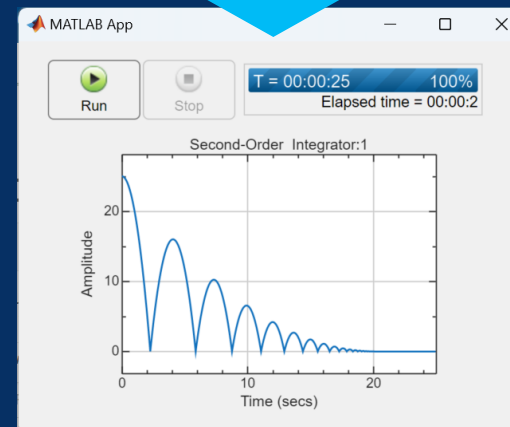
App **integration** with Simulink models, including signal **binding**

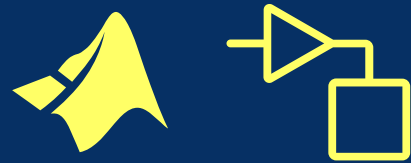
Out-of-the-box, **Simulink-specific** graphical components



**Signal binding with model**

**Simulation app**

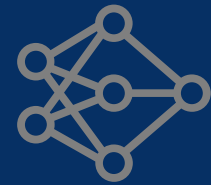




MATLAB®  
& SIMULINK®



Integrations



AI



Ease of Use



Performance



Verification

# MATLAB on Macs



MATLAB and Simulink  
run natively on Apple silicon

Better performance and  
improved battery life on  
MacBooks





# Targeted Performance Improvements each Release



Release notes document the top examples

Each release note includes:

- Example code
- Measured performance improvement
- Hardware used

**Performance**

▼ Language and Programming: Improved performance for reading and writing reading and writing class property values

```
classdef StorageClass
    properties
        data
        average
        ...
        ...
        ...
        ...
        ...
        ...
    end
end
```

This code is about 17.5x faster than in the previous release.

```
s = StorageClass(1:1e6);
timeit(@()s.movingAverage)
```

The approximate execution times are:

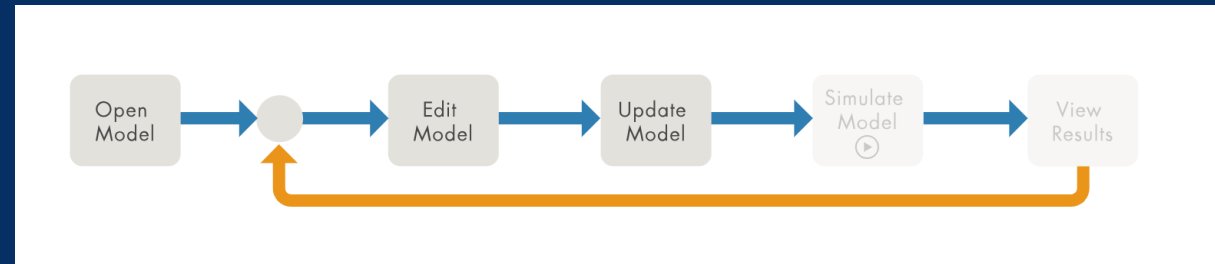
R2023a: 0.497 s

R2023b: 0.0284 s

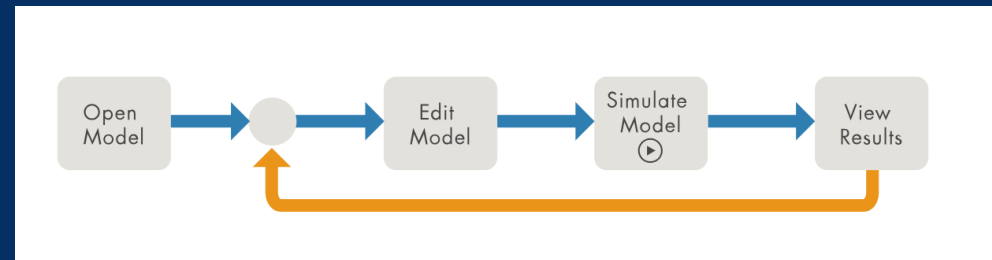
The code was timed on a Windows® 10, Intel® Xeon® CPU E5-1650 v3 @ 3.50 GHz test system.

# Improve simulation performance based on your **workflow**

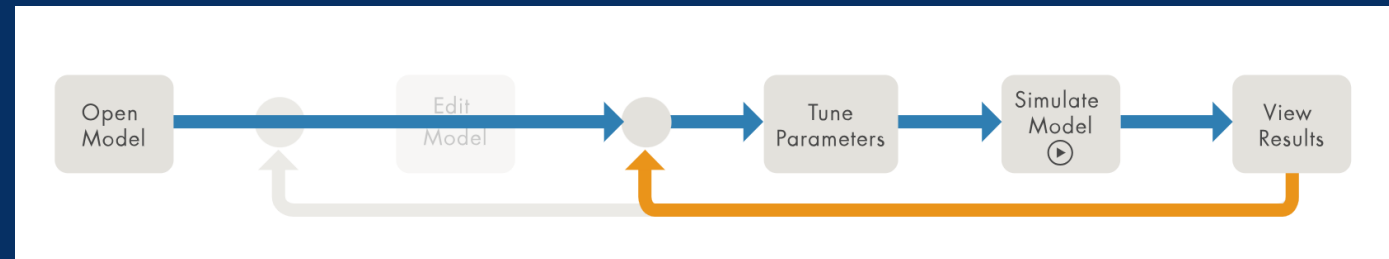
Edit-Update-Repeat



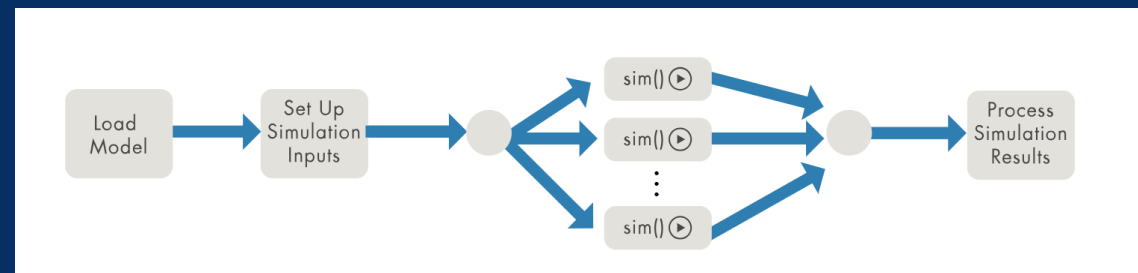
Edit-Sim-Repeat



Tune-Sim-Repeat




Multiple-Sim



# Apply **workflow-specific techniques** to improve performance

Techniques	Workflow			
	Edit-Update-Repeat	Edit-Sim-Repeat	Tune-Sim-Repeat	Multiple-Sim

 **MathWorks**<sup>®</sup> [Products](#) [Solutions](#) [Academia](#) [Support](#) [Community](#)

## Technical Articles

[About MathWorks](#) ▾ | [Careers](#) | [Social Mission](#) ▾ | [Newsroom](#) | [Customer Stories](#) | [Contact Us](#)

# Improving Simulation Performance in Simulink

By Weiwu Li, Reid Spence, and Guy Rouleau, MathWorks



# Apply **workflow-specific techniques** to improve performance

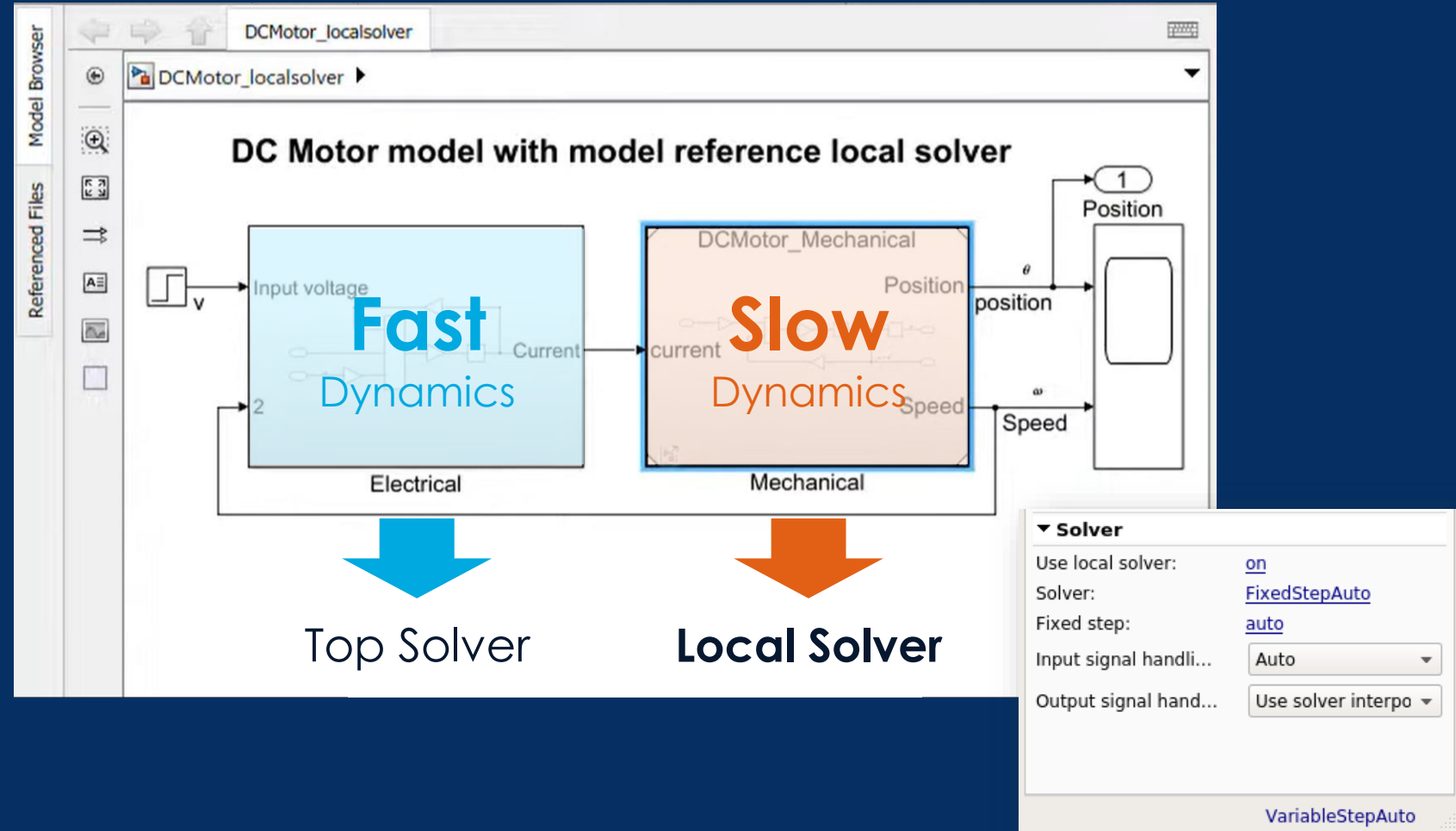


Techniques	Workflow			
	Edit-Update-Repeat	Edit-Sim-Repeat	Tune-Sim-Repeat	Multiple-Sim
Simulation Mode		X	X	X
Fast Restart			X	X
Simulation Cache	X	X	X	X
Model Reference - Parallel Build	X	X		
Model Reference - Incremental Loading & Rebuilding	X	X		
Simulink Profiler	X	X	X	X
Solver Profiler		X	X	X
Modify Your Models		X	X	X
Parallel Simulation				X

# Speed up simulations using the **local solver**



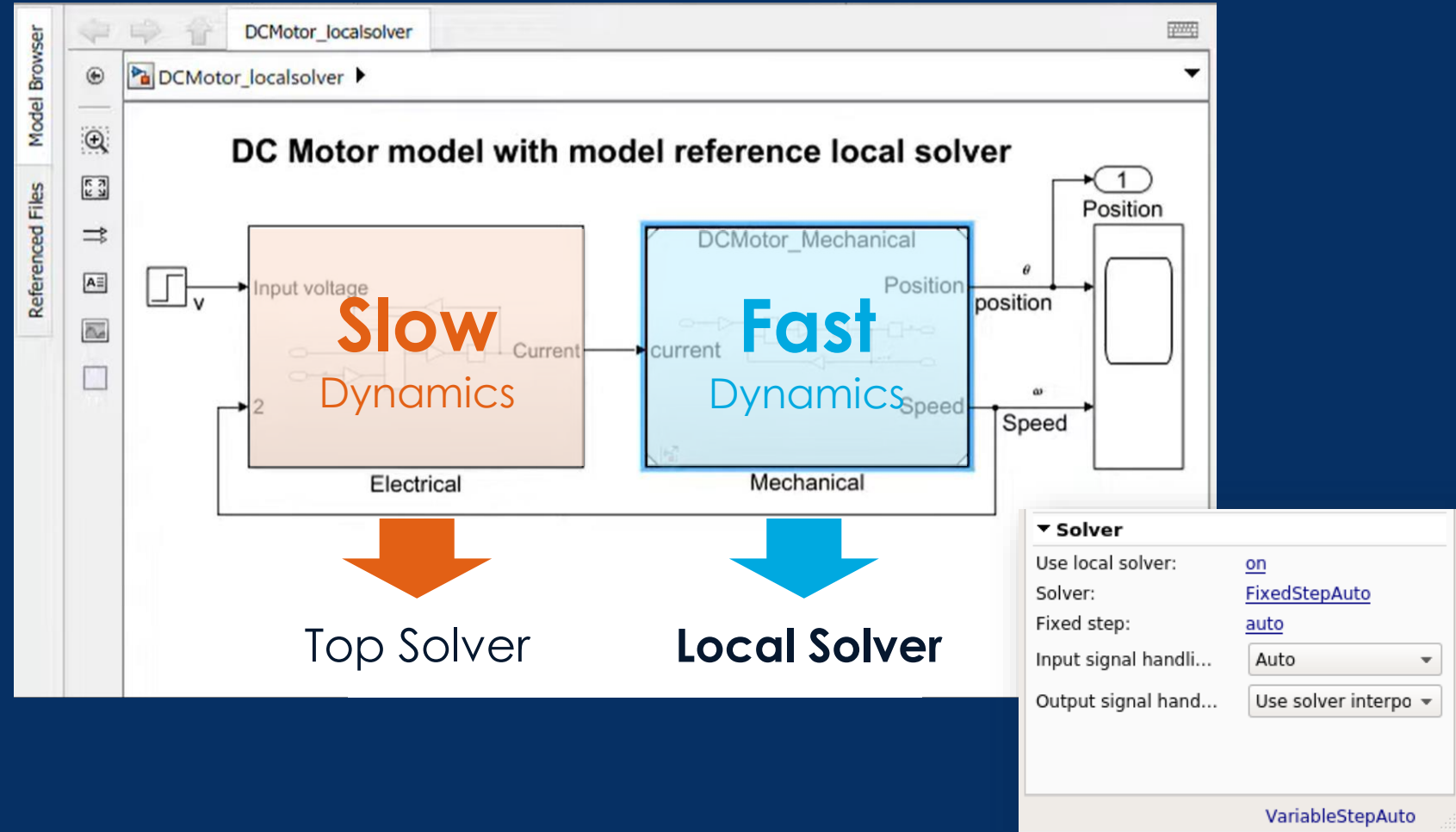
Decouple systems of different dynamics with the **local solver**

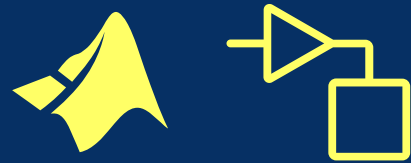


# Speed up simulations using the **local solver**



Support **faster local dynamics**

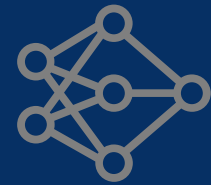




MATLAB®  
& SIMULINK®



Integrations



AI



Ease of Use



Performance



Verification

**New products**

**Major updates**



### Design

Simulink Design Verifier

Simulink Check

HDL Verifier

Simulink Fault Analyzer

### Test

Simulink Coverage

Simulink Test

MATLAB Test

Polyspace Test

### Code

Polyspace Bug Finder

Polyspace Code Prover

Polyspace Access

### Certify

DO Qualification Kit

IEC Certification Kit

Simulink Code Inspector

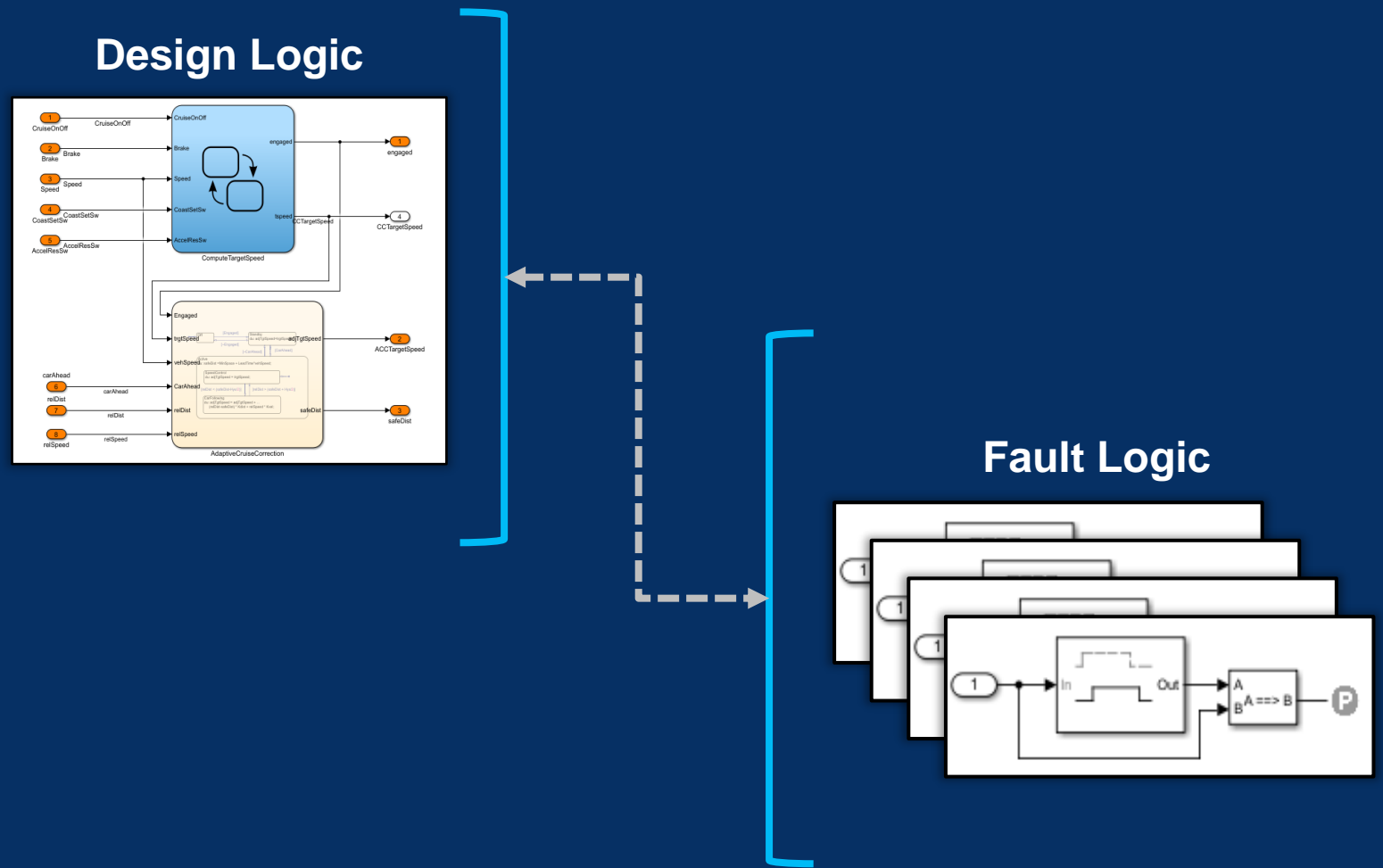
Requirements Toolbox



# Model faults and analyze effects with **Simulink Fault Analyzer**



Model faults **without** modifying the design



# Model faults and analyze effects with **Simulink Fault Analyzer**



Model faults without  
modifying the design

Manage faults across  
**multiple domains**

Enable	Model Element/Fault Name	Active Fault	Trigger
<input checked="" type="checkbox"/>	Environment/Constant6/Outport/1		
	HighTemperatureFault	<input type="checkbox"/>	Conditional: highSpeedCondition
	LowTemperaturFault	<input checked="" type="checkbox"/>	Conditional: SampleConditional
<input checked="" type="checkbox"/>	Environment/Constant7/Outport/1		
	HighPressureFault	<input checked="" type="checkbox"/>	Timed: 20
	LowPressureFault	<input type="checkbox"/>	Always On
<input checked="" type="checkbox"/>	Environment/Constant2/Outport/1		
	Grade_fault	<input checked="" type="checkbox"/>	Always On
	Grade_fault_1	<input type="checkbox"/>	Always On
<input checked="" type="checkbox"/>	Environment/Constant3/Outport/1		
	wind_x_fault	<input checked="" type="checkbox"/>	Always On
<input checked="" type="checkbox"/>	Passenger Car/Electric Plant/Simscape/Inductor1/Inductor		
	Inductor1_fault	<input checked="" type="checkbox"/>	Behavioral

# Model faults and analyze effects with **Simulink Fault Analyzer**



Model faults without  
modifying the design

Manage faults across  
multiple domains

Simulate, explore and  
**analyze fault effects**

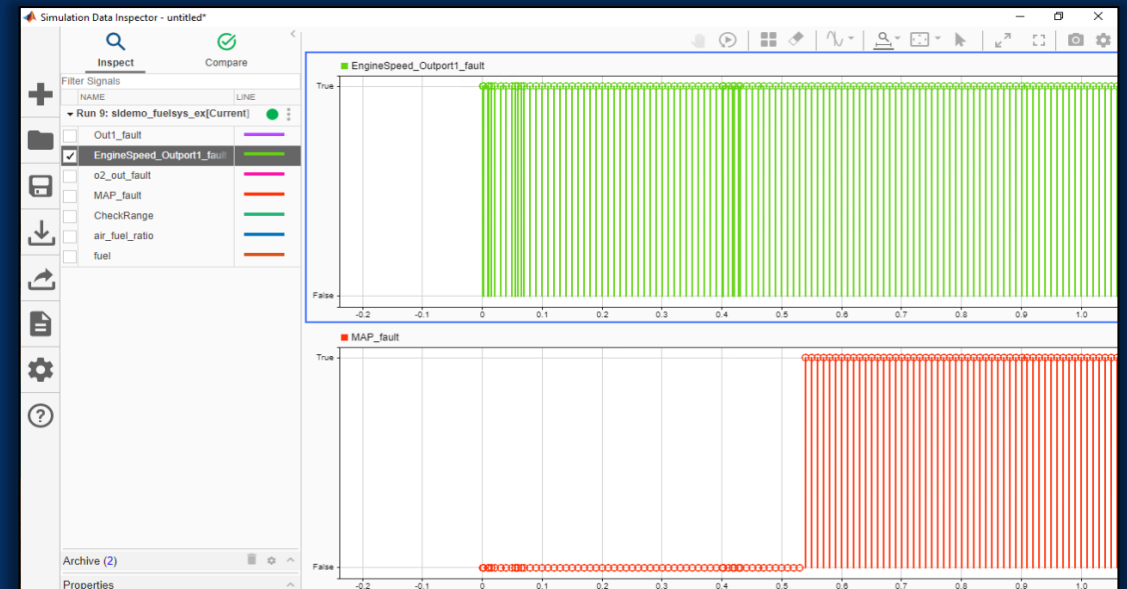
Details: Design Study

Specification Run Options

Root Parameter Set

Fault Set\_1

<input checked="" type="checkbox"/>	Fault	Component
<input checked="" type="checkbox"/>	HighTemperatureFault	EvReferenceApplic...
<input checked="" type="checkbox"/>	HighPressureFault	EvReferenceApplic...
<input checked="" type="checkbox"/>	LowTemperaturFault	EvReferenceApplic...
<input checked="" type="checkbox"/>	LowPressureFault	EvReferenceApplic...
<input checked="" type="checkbox"/>	Grade_fault	EvReferenceApplic...
<input checked="" type="checkbox"/>	Grade_fault_1	EvReferenceApplic...
<input checked="" type="checkbox"/>	wind_x_fault	EvReferenceApplic...



# Model faults and analyze effects with **Simulink Fault Analyzer**



Model faults without modifying the design

Manage faults across multiple domains

Simulate, explore and analyze fault effects

Perform **systematic safety analysis**

The screenshot shows the Safety Analysis Manager interface with a table of failure modes. The table has three columns: Failure Mode, Failure Effect, and Detection Method. The first three rows are visible, each with a link icon in the first column.

	Failure Mode	Failure Effect	Detection Method
1	Single Fault: O2 Fault	Engine Operation Interrupted	O2 Fault Detection
2	Single fault: MAP Fault	Engine Operation Interrupted	Pressure Fault Detection
3	Multiple Faults: O2 Fault and MAP Fault	Engine Inoperative	Multiple Faults Detected

The Properties panel on the right shows the 'Description' and 'Links' sections. The 'Links' section shows a link to 'ego\_fault'.



New products



Major updates



### Design

Simulink Design Verifier

Simulink Check

HDL Verifier

Simulink Fault Analyzer

### Test

Simulink Coverage

Simulink Test

MATLAB Test

Polyspace Test

### Code

Polyspace Bug Finder

Polyspace Code Prover

Polyspace Access

### Certify

DO Qualification Kit

IEC Certification Kit

Simulink Code Inspector

Requirements Toolbox

# Test Product Family



## R2015a

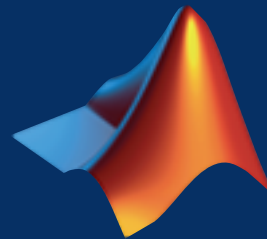
Simulink Test



Model Centric

## R2023a

MATLAB Test



MATLAB Code Centric

## R2023b

Polyspace Test



C/C++ Code Centric

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



Static Analysis

Polyspace Bug Finder

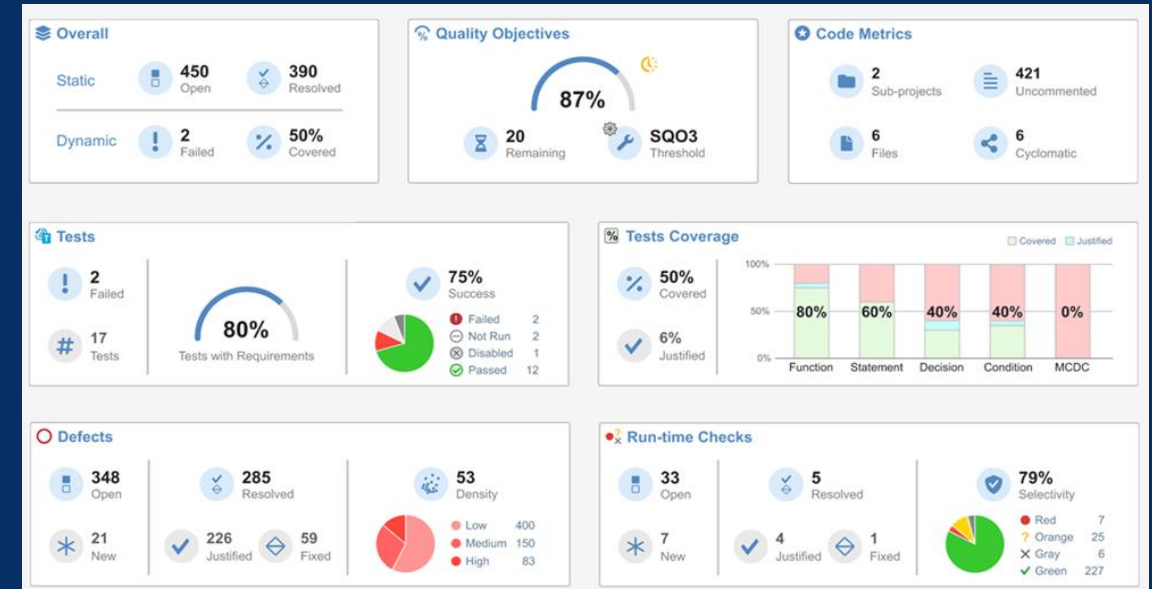


Polyspace Code Prover



Dynamic Testing

Polyspace Test



Centrally manage and combine static analysis with dynamic testing

ISO 26262 DO-178 IEC 61508 IEC 62304



MATLAB®  
& SIMULINK®



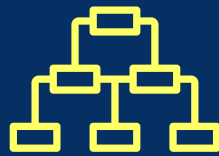
**Integrations**



AI



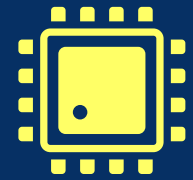
**Languages**



**Simulation**



**Visualization**



**Hardware**



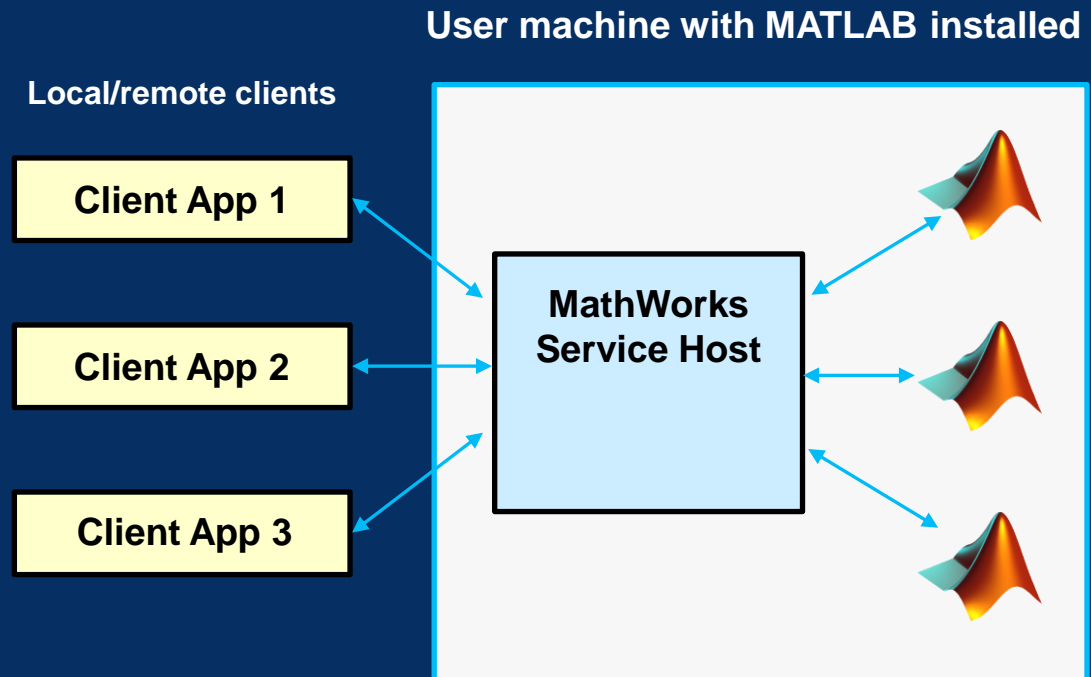


# Call MATLAB from any local or remote client program using **REST**

Write client programs to call MATLAB using the **MATLAB Function Service**

The service uses HTTPS protocol

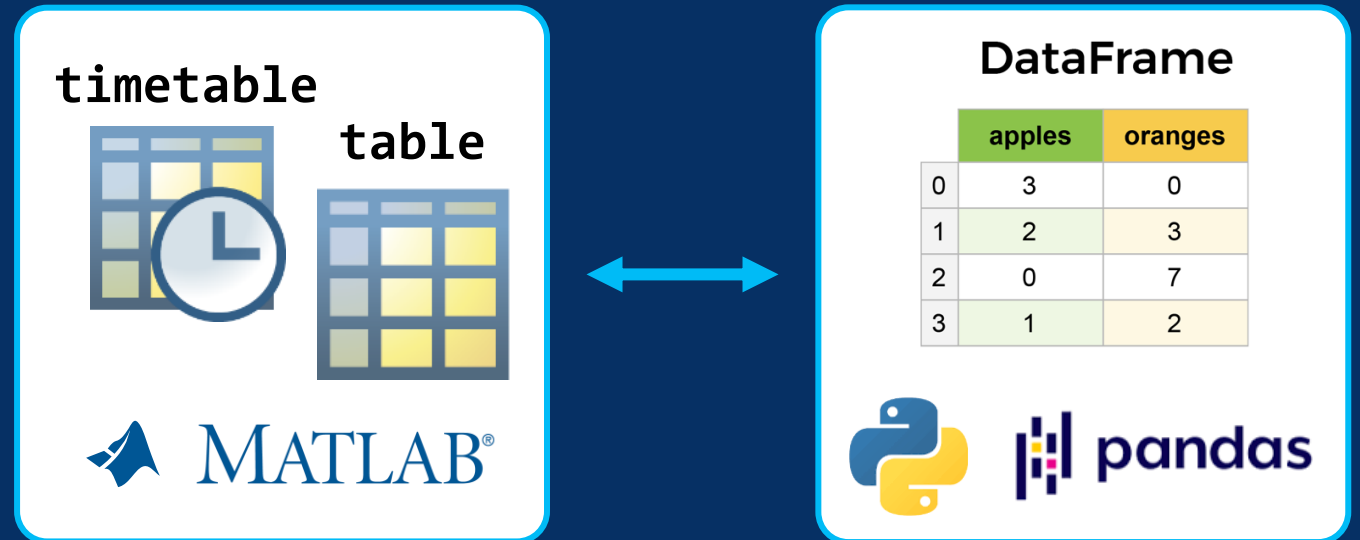
## REST Function Service





# Use MATLAB with Python

Automatically convert between  
MATLAB `table` or `timetable`  
and **Python Pandas DataFrame**





# Use MATLAB with Python

Automatically convert between MATLAB `table` or `timetable` and Python Pandas DataFrame

Interactively run Python code with **Run Python Code Live Editor** task

```
a = 3;
b = 4;
```

**Run Python Code**  Autorun ? ⋮

= Run Python code using MATLAB variables a, b

▼ Select input type

Code  File

▼ Enter Python code

```
import math
h = math.sqrt(a**2 + b**2)
```

▼ Output options

Specify outputs  Return all  Return selected variables

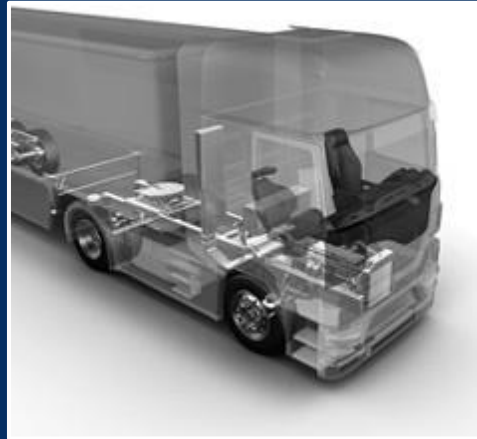
Display outputs

▶ Show code



# Simulink is a **Simulation Integration Platform**

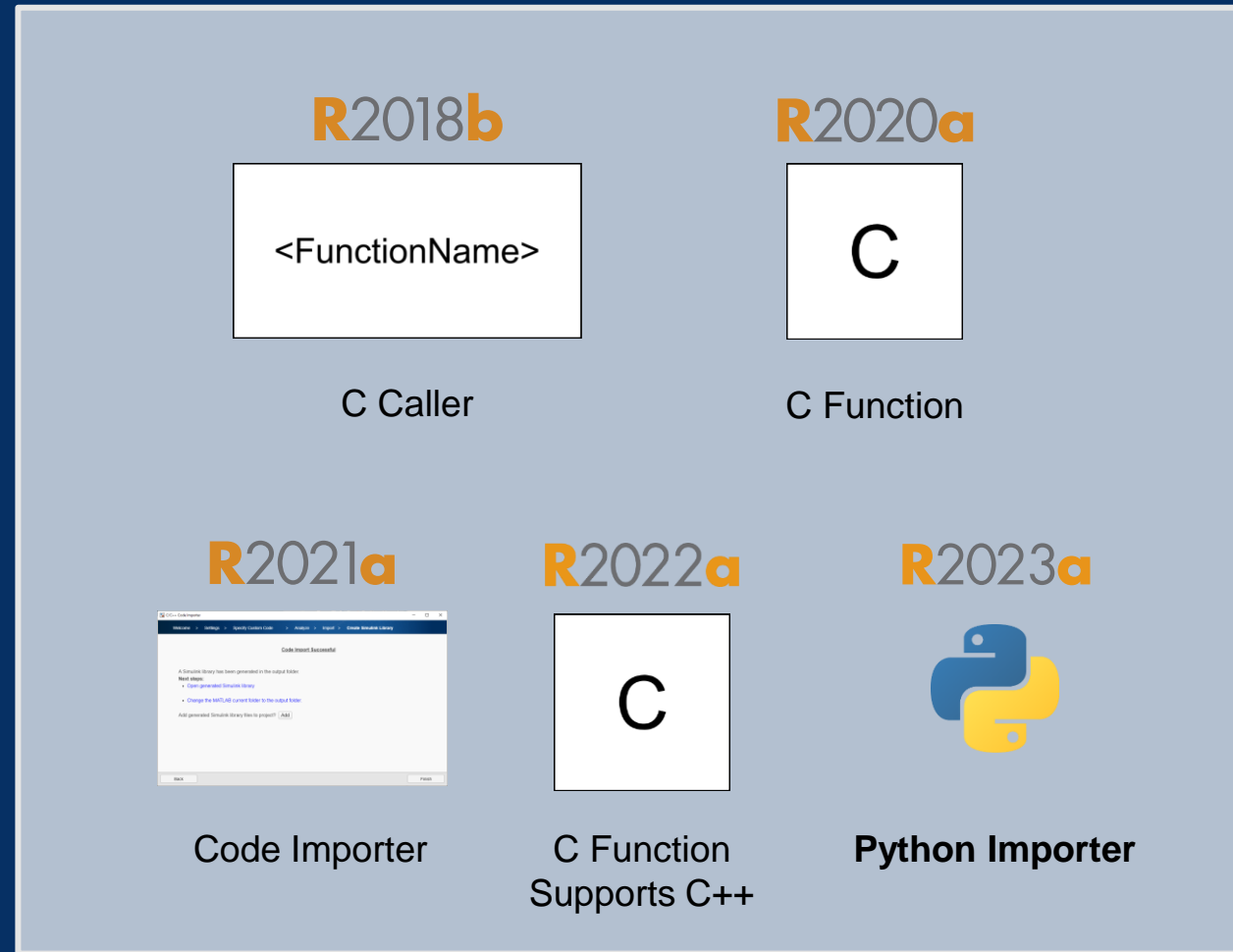
**Ecosystem and interoperability** with 100+ third-party languages and tools





# Import **custom code** into Simulink

Build custom code components using C, C++, and **Python**





# Import Python functions within classes

Python importer supports Python functions specified within Python classes

```
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
```

Python class definition



# Import Python functions within classes

Wizard UI provides  
step-by-step import  
guidance

Python Importer

Welcome > Settings > Specify Custom Code > Analyze > **Import** > Create Simulink Library

**Edit the block port specifications for the previously selected functions.**

Search

Name	Scope	Label	Type	Complexity	Size
▼ fx pyFile.room.room					
self	Parameter	self	ClassObject	real	[1 1]
length	Parameter	length	double	real	[1 1]
breadth	Parameter	breadth	double	real	[1 1]
height	Parameter	height	double	real	[1 1]
▼ fx pyFile.room.volume					
result	Output	result	double	real	[1 1]
self	Parameter	self	ClassObject	real	[1 1]
▼ fx pyFile.room.wallarea					
result	Output	result	double	real	[1 1]
self	Parameter	self	ClassObject	real	[1 1]

What to consider

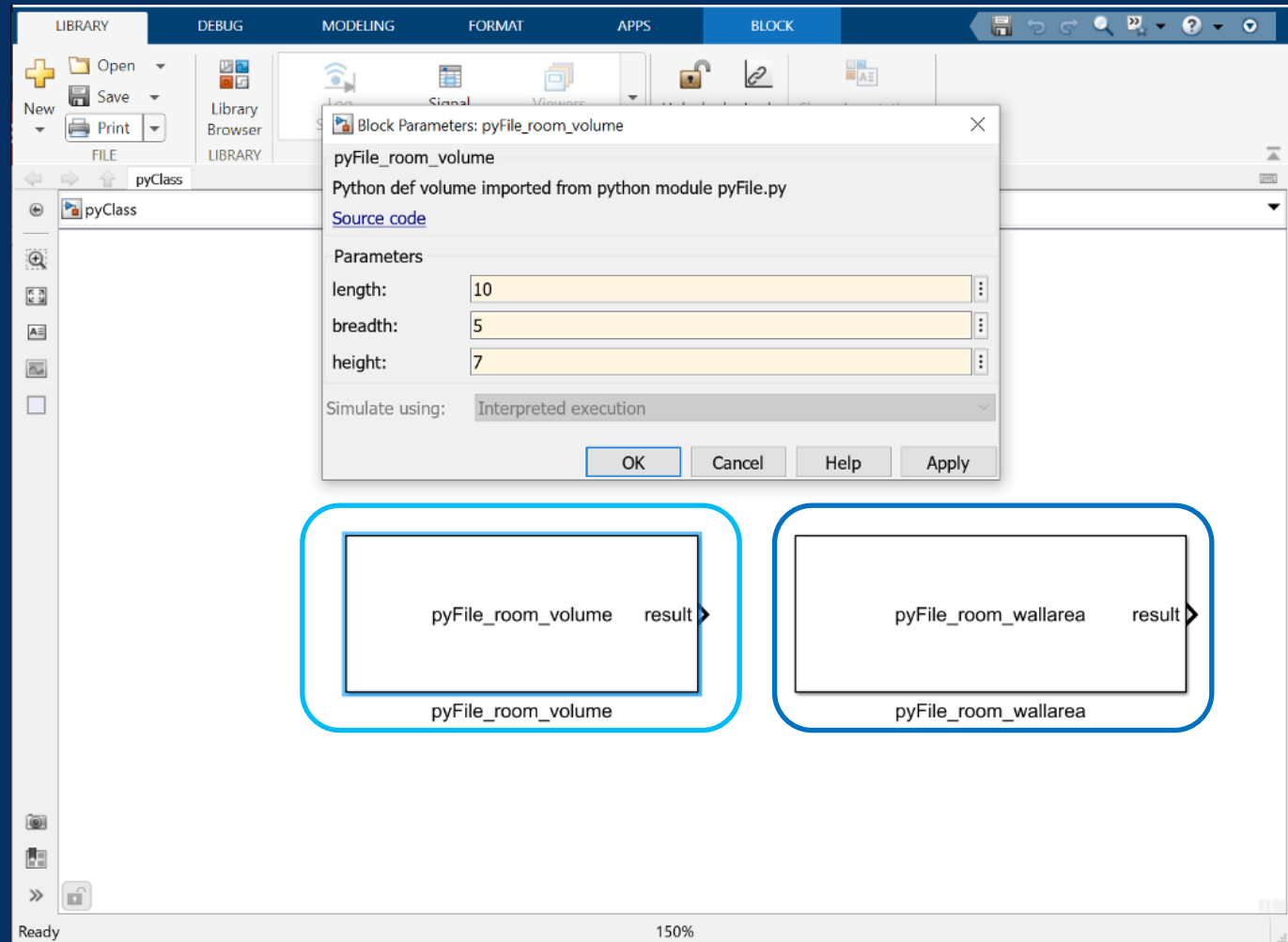
System blocks will be generated with default port specifications unless you specify otherwise. The block port specifications can also be edited in the generated System object.

Back Next



# Import Python functions within classes

Export as **custom blocksets** for simulations

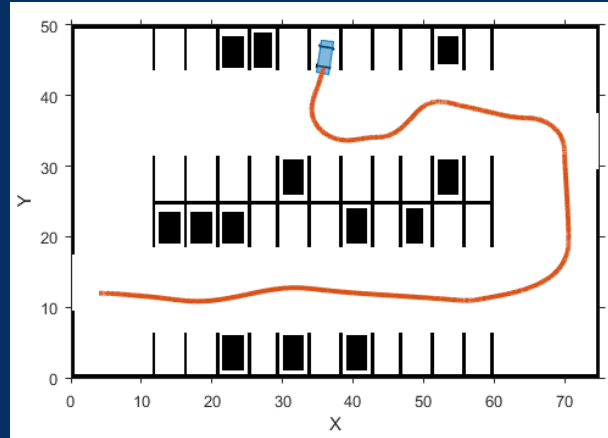




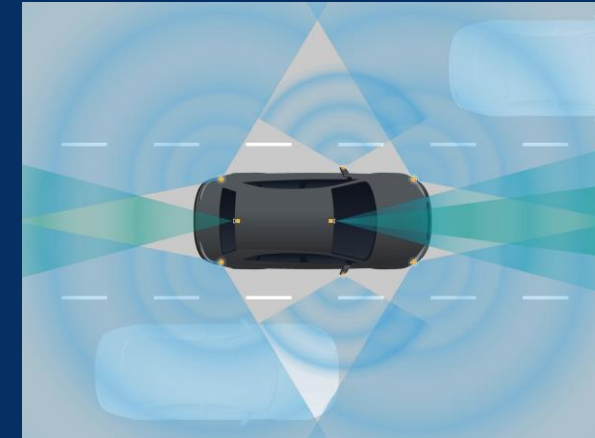


# Support for **unbounded** variable-size signals

**Flexibility** to model signals without specifying a finite signal size



Autonomous parking maneuver system



Radar system

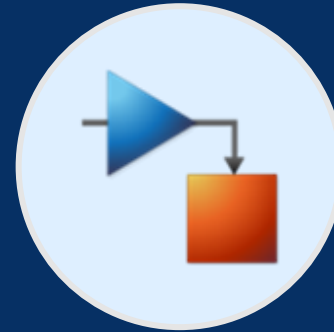
Sizes of signals are **unknown at compilation** and can grow/shrink during simulation



# Support for **unbounded** variable-size signals

Provide a **mapping** between Simulink signals and dynamic arrays in C++

Easily **exchange data** with other external software components



**Unbounded** variable-size signals

**Dynamic** arrays

Signal size as **Inf**

**Resizable** data

Memory allocated **at run time**

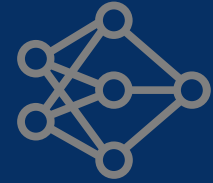
**Dynamic** memory



MATLAB®  
& SIMULINK®



**Integrations**



AI



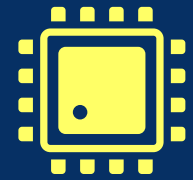
**Languages**



**Simulation**



**Visualization**



**Hardware**



# Import FMI 3.0 Function Mockup Units (FMUs)

FMU block supports  
FMI version 3.0

The screenshot shows the Simulink environment with two FMU blocks. The top block is labeled 'FMI 3.0 FMU - Co-Simulation' and has four inputs (in1 to in4) and four outputs (out1 to out4). The bottom block is labeled 'FMI 3.0 FMU - Model-Exchange' and has four inputs (in1 to in4) and eight outputs (out1 to out4). A dialog box titled 'Block Parameters: FMI 3.0 FMU - Co-Simulation' is open, showing the block name 'array\_double\_2d [Co-Simulation, v3.0]' and a table of parameters.

Parameter	Value
param1	
param2	
param3	
R [1,1]	0
R [1,2]	-1
R [2,1]	-2
R [2,2]	-3
param4	



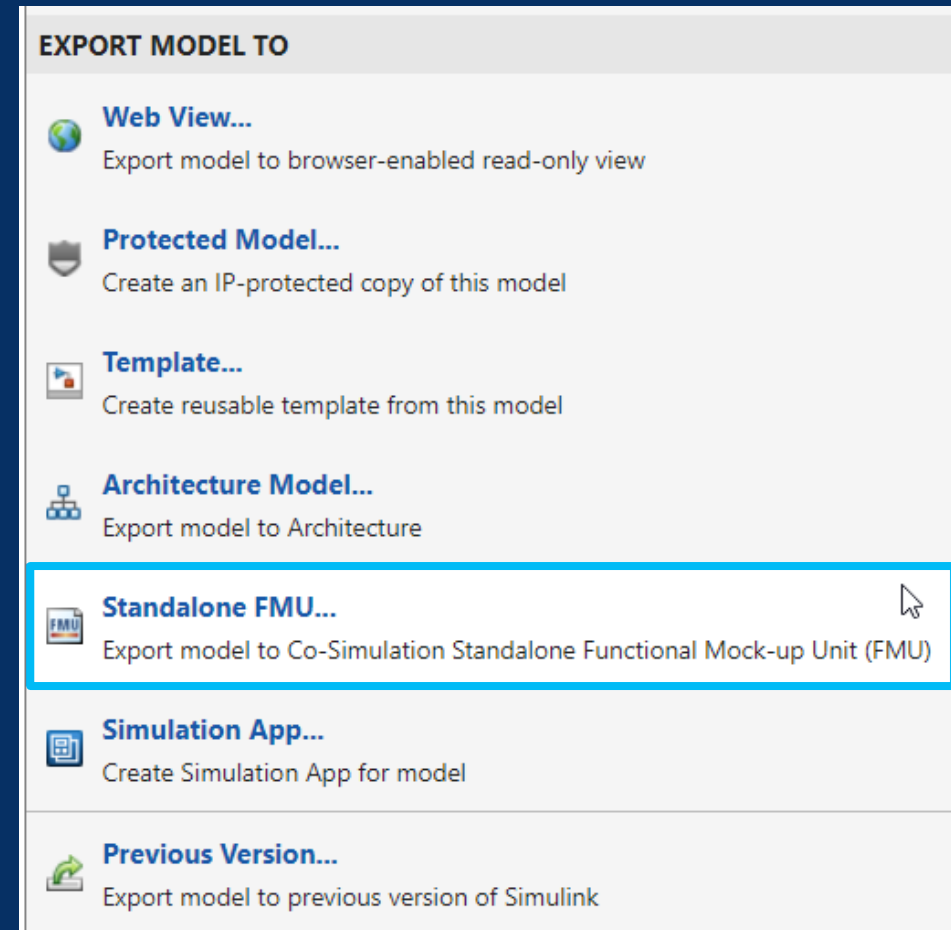
# Export simulations as FMI 3.0 FMUs

FMU Builder for Simulink Support Package

**Create** standalone FMUs from Simulink models or source code

**Validate** FMI 3.0 FMUs

**Export** FMUs to be used in other simulation environments





# Control scripted simulations using **Simulation** object

Configure, run, and interact with simulations, including stepping

Access in-simulation status and outputs

Deployable to other environments with compiler workflows

initialize  
start  
step  
pause  
resume  
stop  
terminate

```

Command Window
>> mdl = 'sldemo_suspn_3dof';
>> open_system(mdl);
>> sm = simulation(mdl);
>> initialize(sm)
>> start(sm);
>> pause(sm);
>> sm.Status

ans =

    "paused"

>> sm.Time

ans =

    16.8811

>> step(sm, PauseTime=40);
>> resume(sm)
>> stop(sm);
>> sm.Status

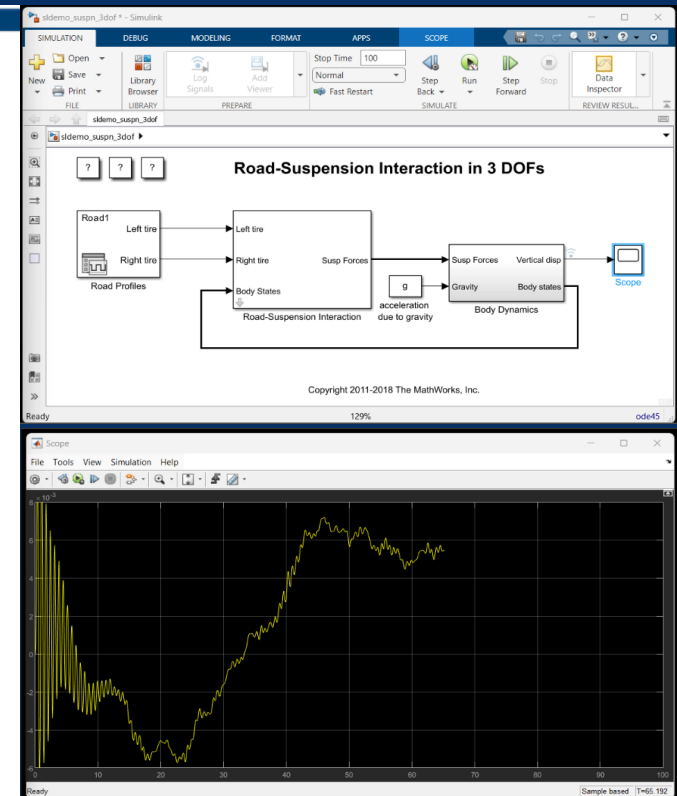
ans =

    "inactive"

```

Scripted simulation

Simulink model

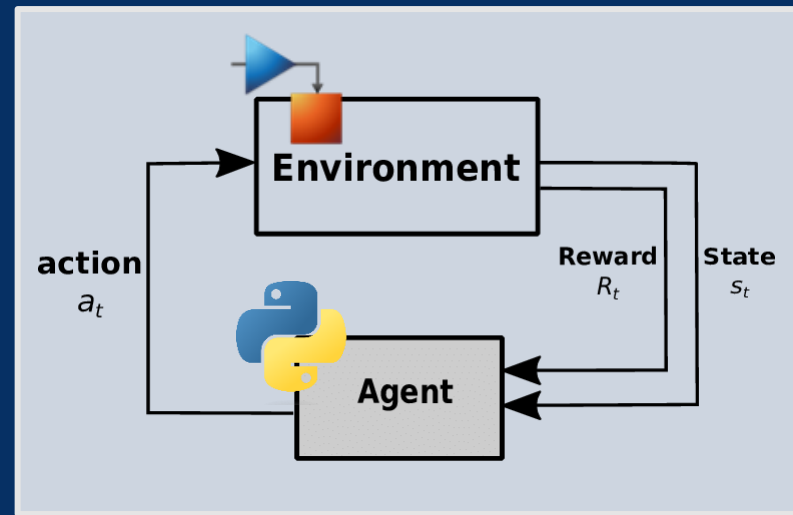


In-simulation outputs

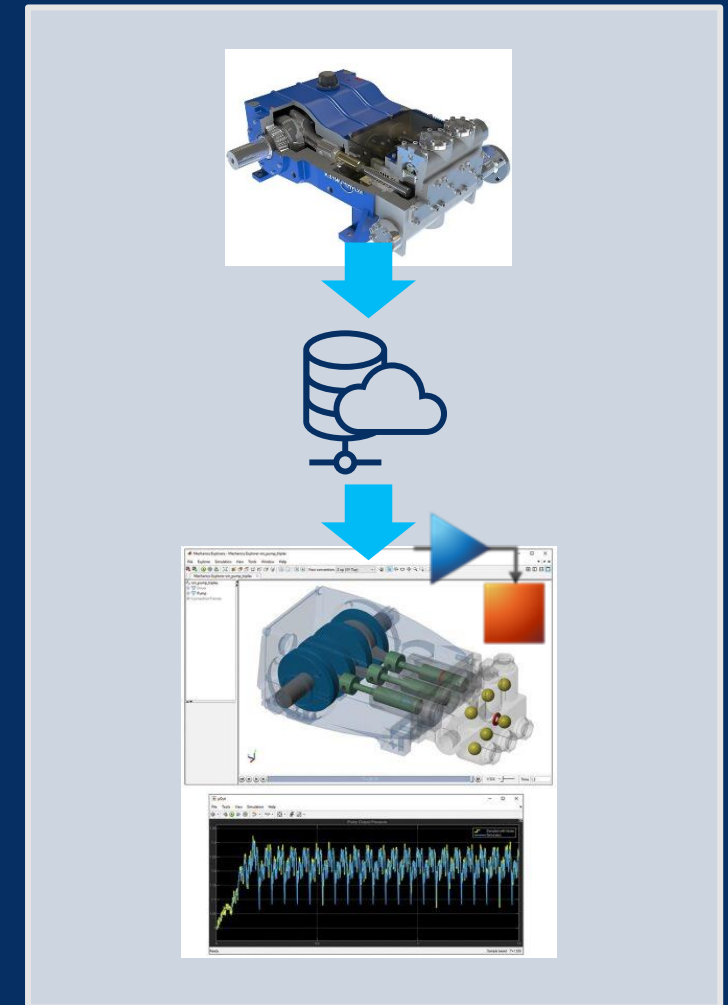


# Control **scripted simulations** using **Simulation** object

Enable **simulation integration** in new applications such as reinforcement learning and digital twins



Reinforcement Learning



Digital Twins



MATLAB®  
& SIMULINK®



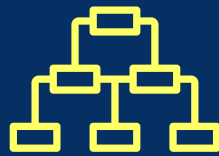
**Integrations**



AI



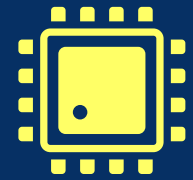
**Languages**



**Simulation**



**Visualization**



**Hardware**





# Visualize 3D simulations

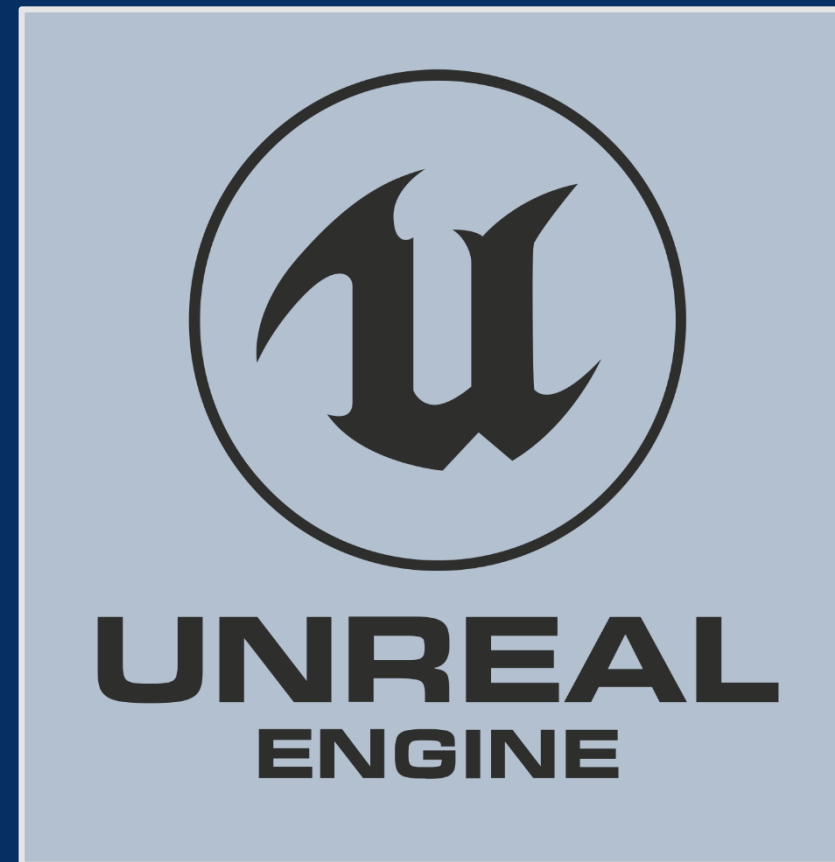
## Simulink 3D

Animation provides foundational 3D assets, platform, and **integration to the Unreal Engine** for vertical product

## Simulink 3D

Animation supports Unreal Engine 5.1

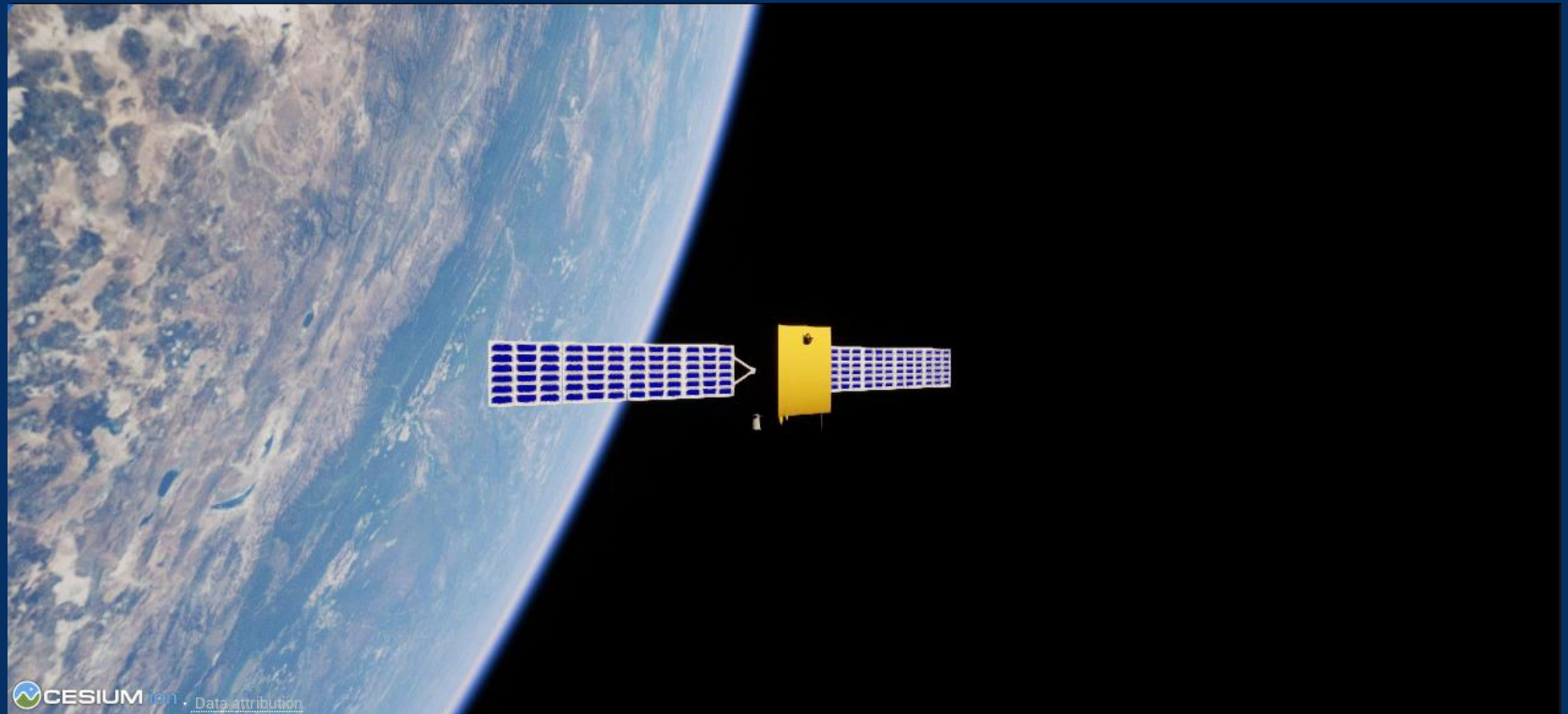
## 3D Visualizations





# Visualize 3D simulations

Photorealistic 3D scenes, actions, and sensors for simulating dynamic system behavior



Aerospace Blockset



# Visualize 3D simulations

Photorealistic 3D scenes, actions, and sensors for simulating dynamic system behavior



UAV Toolbox



# Visualize 3D simulations

Photorealistic 3D scenes, actions, and sensors for simulating **dynamic system behavior**



Vehicle Dynamics Blockset



# Visualize 3D simulations

Photorealistic 3D scenes, actions, and sensors for simulating dynamic system behavior



Automated Driving Toolbox



# Visualize 3D simulations

Photorealistic 3D scenes, actions, and sensors for simulating dynamic system behavior



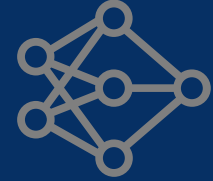
Robotics System Toolbox



MATLAB®  
& SIMULINK®



**Integrations**



AI



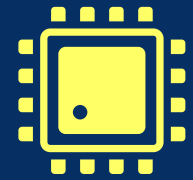
**Languages**



**Simulation**



**Visualization**



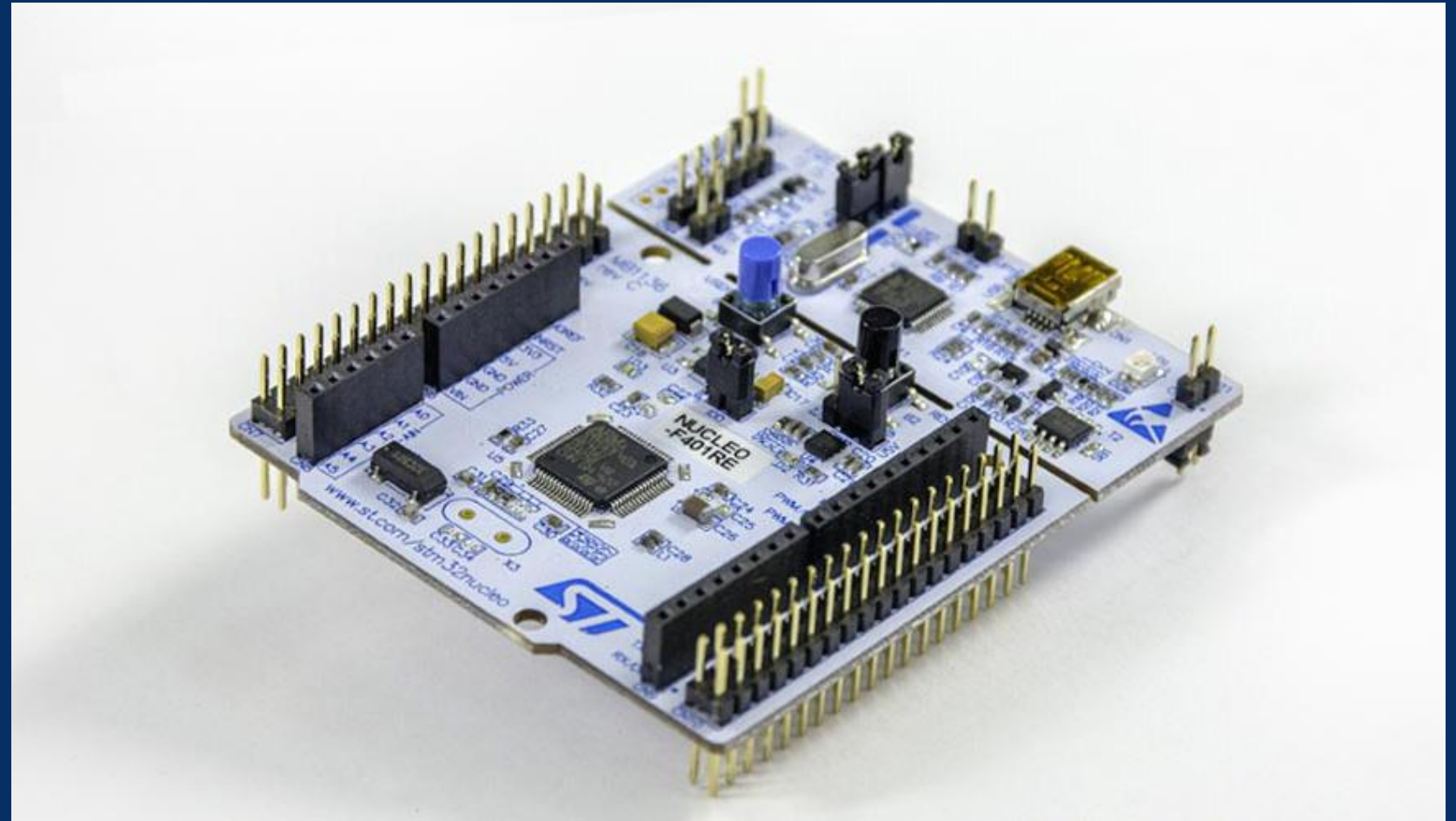
**Hardware**

# ST Microelectronics



Support for **dual-core** devices

Support for 4 new device families:  
**U5, L4, L5, WB**







Support for **AURIX  
TC3x**

Support for **AURIX  
TC4x PPU  
accelerator**



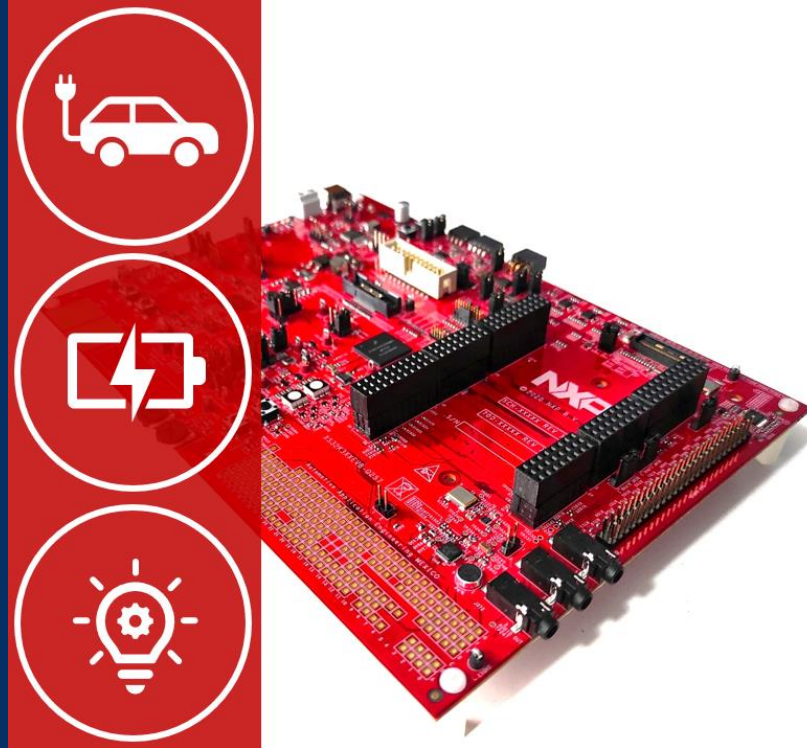
Embedded Coder

Embedded Coder Support Package for Infineon® AURIX™ TC3x Microcontrollers



Support for S32M2,  
S32K396, LAX  
(S32R45), LPC553x,  
and BMS

Support for S32K3,  
S32ZE, and HCP



S32K3xx



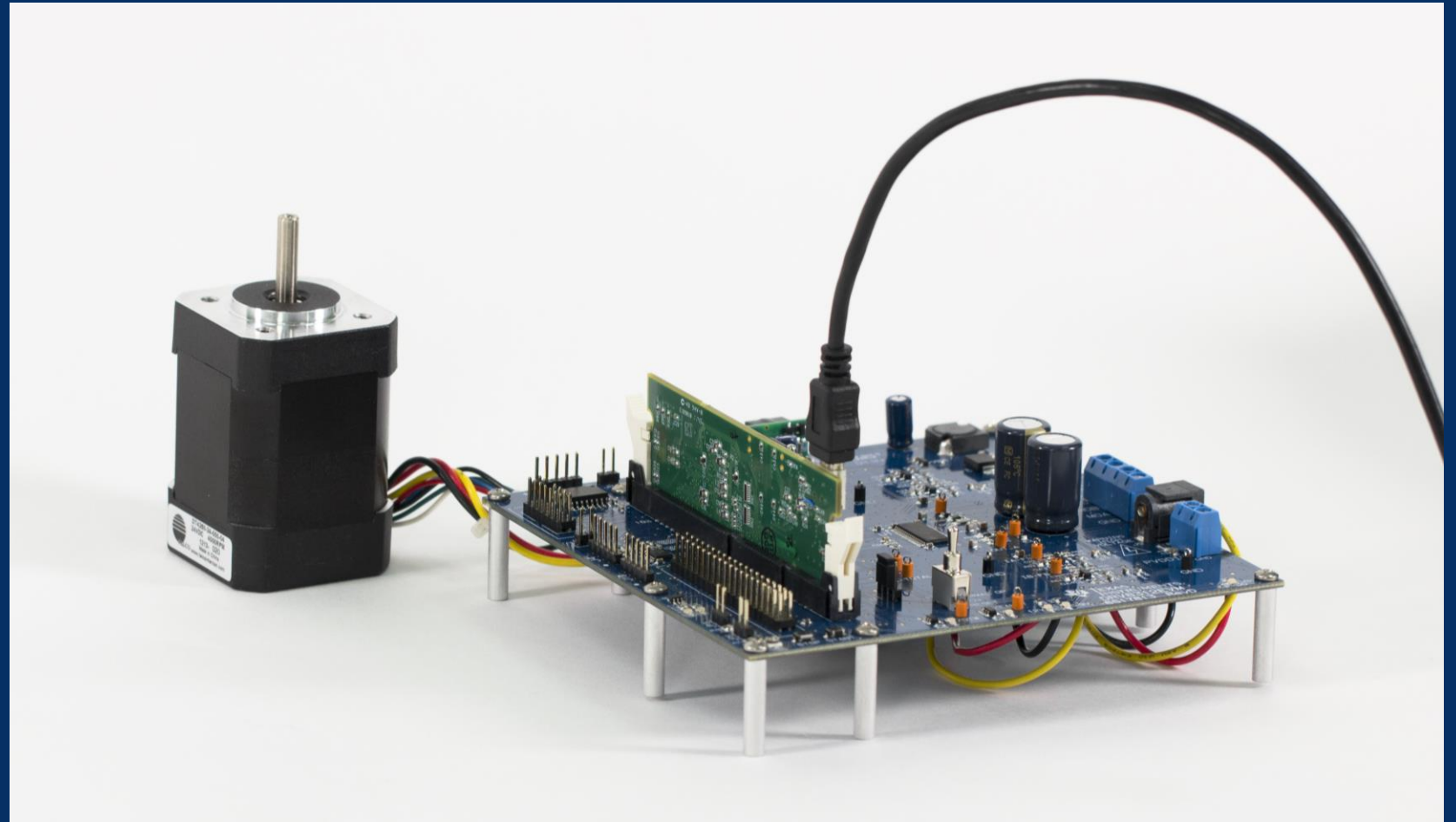
HCP



# Texas Instruments (TI)

## C2000 Microcontroller Blockset

Design, simulate,  
and implement  
applications for TI  
C2000  
Microcontrollers

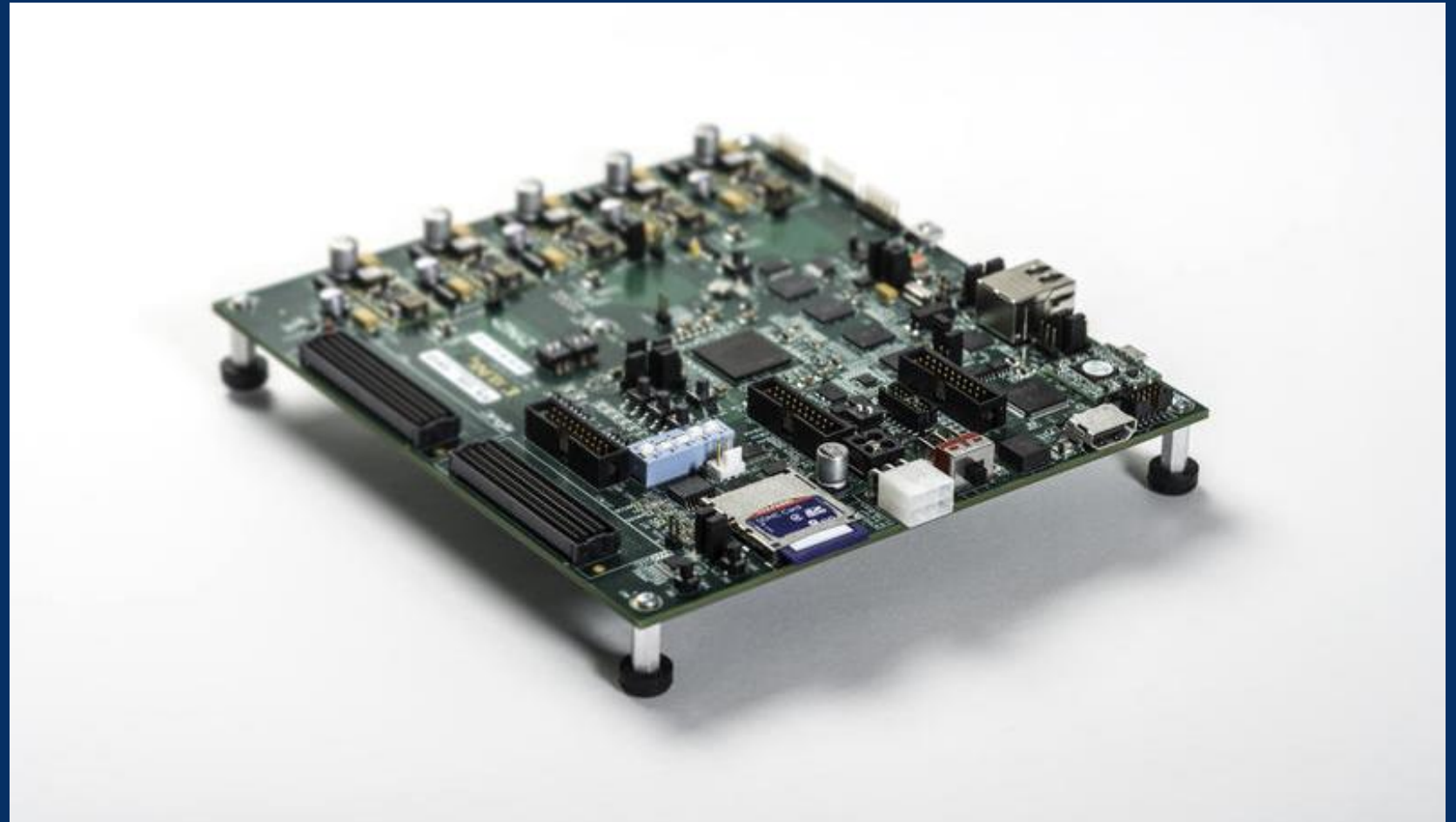


# AMD



Design, analyze, and prototype for Versal Adaptive SoCs, Zynq SoCs, and Xilinx FPGA devices

Generate and deploy HDL code and Embedded Software for Xilinx FPGA and SoC devices





MATLAB®  
& SIMULINK®



**Integrations**



**AI**





# New capabilities across the entire AI workflow

**Data  
Preparation**

**AI Modeling**

**Simulation  
& Test**

**Deployment**

# Collaborative, multiuser, team-based labeling

Data Preparation

AI Modeling

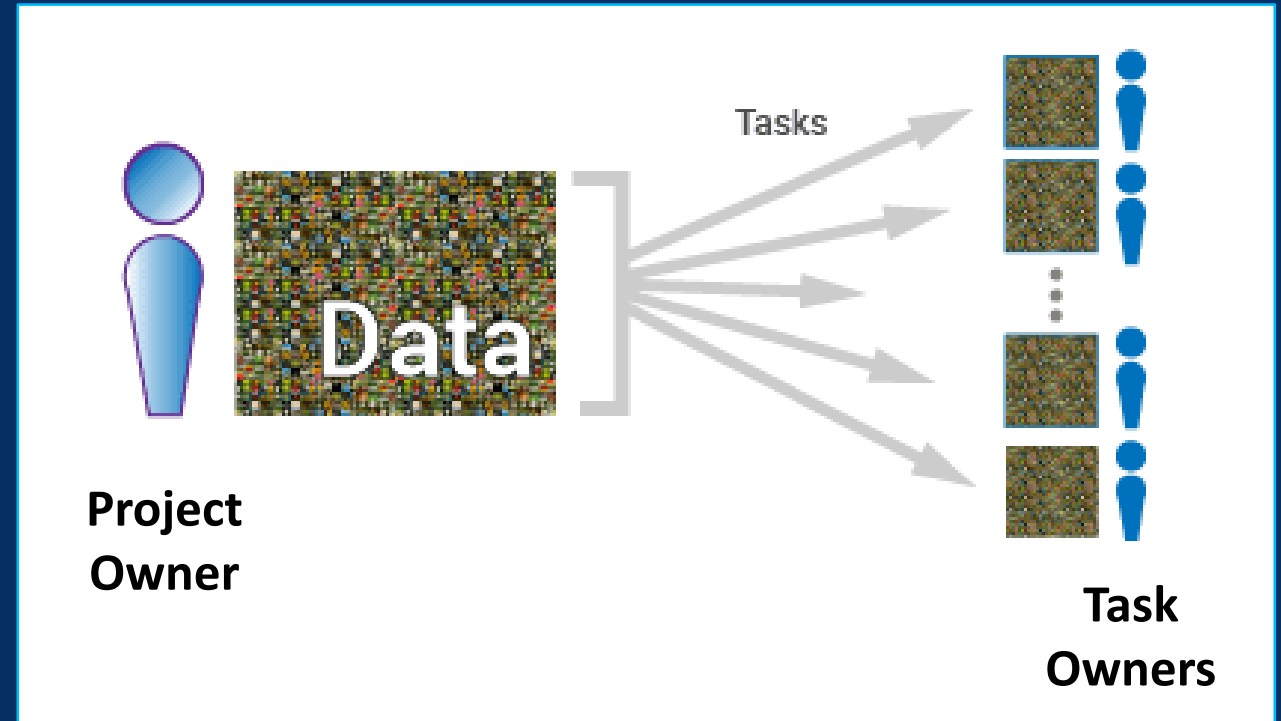
Simulation & Test

Deployment



Distribute, monitor, and review labeling tasks across a team

Create an executable labeling app, which team members can use to label or review tasks without a MATLAB license



# Detect objects in images using a YOLOX deep learning network

Data Preparation

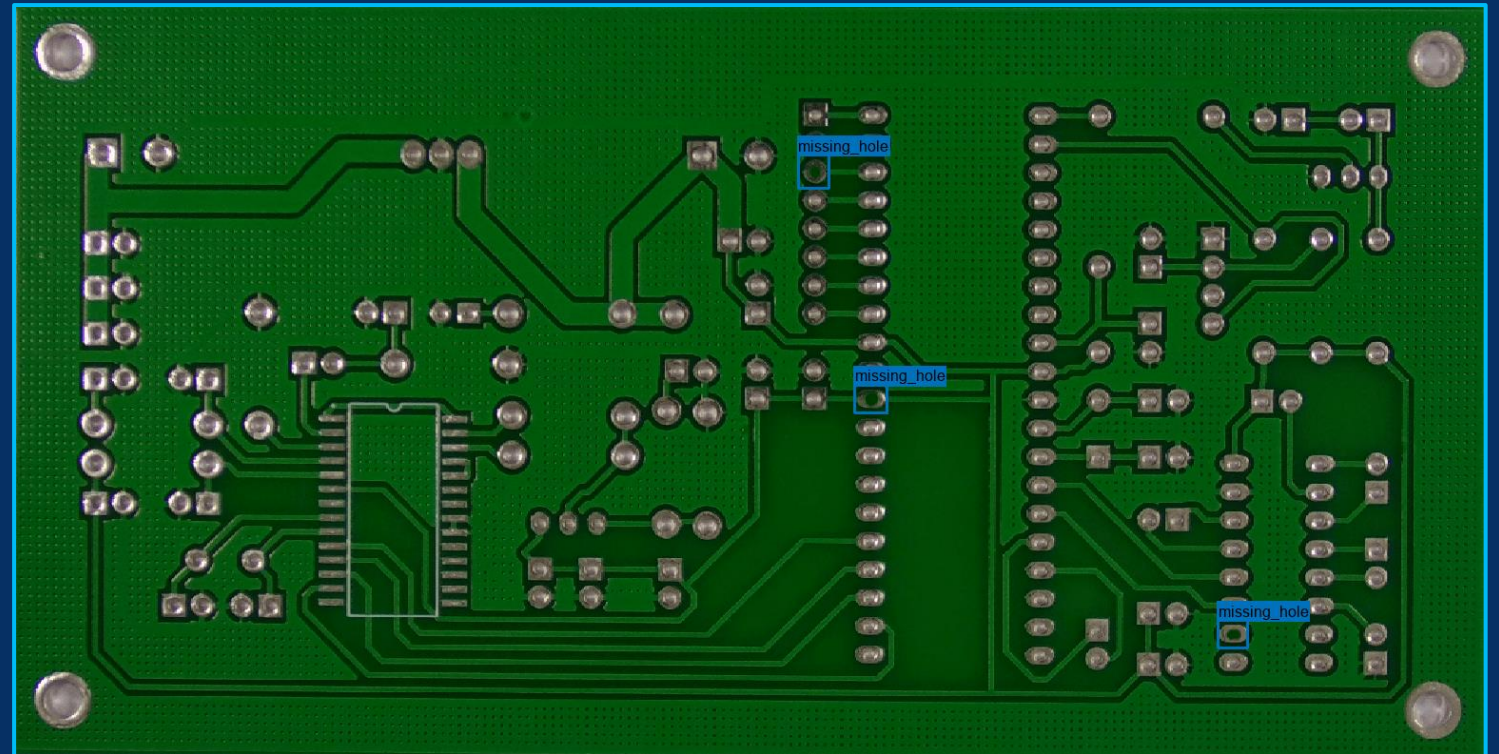
AI Modeling

Simulation & Test

Deployment



Create pretrained or custom YOLOX object detectors



## Automated Visual Inspection Example

*Detect, localize, and classify defects in printed circuit boards (PCBs) using a YOLOX object detector.*



# Execute Python deep learning models in Simulink

Coexecute TensorFlow and PyTorch models in Simulink together with other deep learning and machine learning blocks



# Explain object detection network predictions using D-RISE

Data  
Preparation

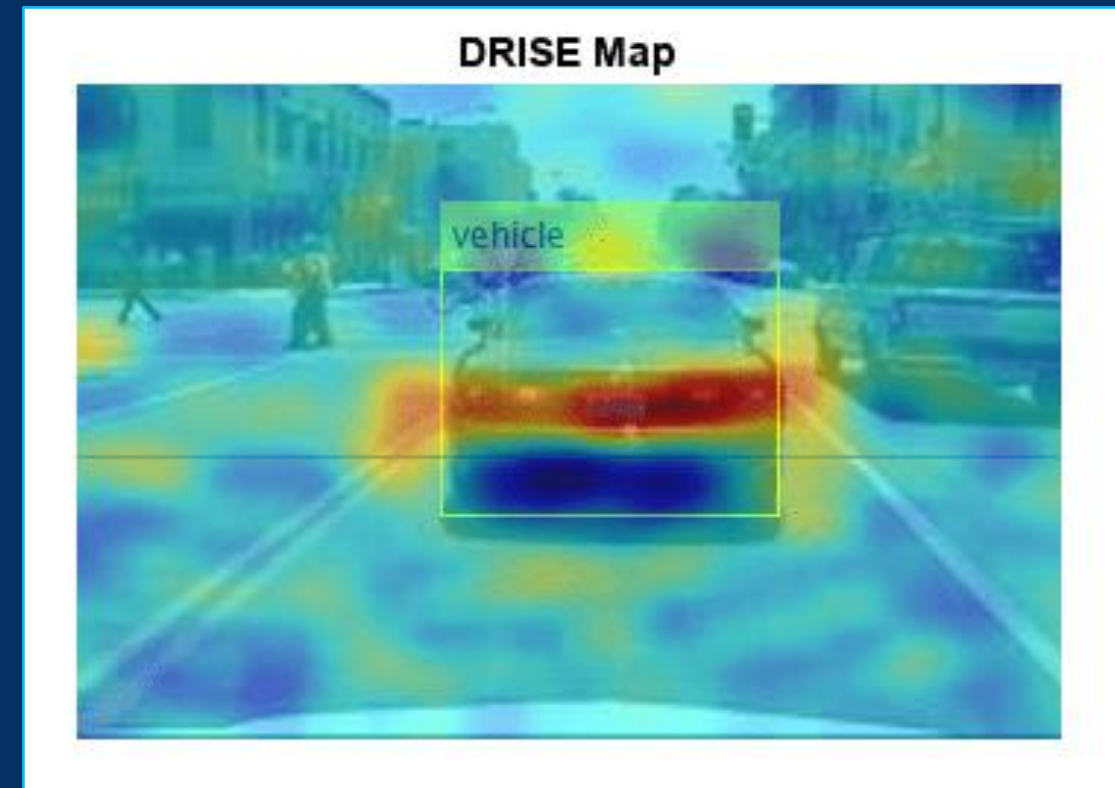
AI Modeling

Simulation  
& Test

Deployment



Generate visual explanations for the prediction results of object detection networks



# Generate generic CUDA code for deep learning

Generate deep learning CUDA code that does not require NVIDIA deep learning libraries





MATLAB®  
& SIMULINK®



Integrations



AI



Ease of Use



Performance



Verification



Languages



Simulation



Visualization



Hardware



# Learn more on mathworks.com

R2024a リリースハイライト – MATLAB および Simulink サイト内検索 

## リリースハイライト

最新リリースをダウンロードして、MATLAB と Simulink の価値をご実感ください。

[今すぐダウンロード](#)



メジャーアップデート

- **Computer Vision Toolbox** – YOLOX オブジェクト検出器を展開。チームベースのラベル付けを実施。リアルタイムの Visual SLAM を実行。
- **Deep Learning Toolbox** – Transformer などのアーキテクチャをサポート。PyTorch と TensorFlow モデルのインポートおよびコシミュレーションを実行。
- **GPU Coder** – ディープラーニング向け汎用 CUDA コードを生成。MEX コード生成向けシングル メモリ マネージャーの使用とコードのプロファイリング。
- **Instrument Control Toolbox** – 計測器エクスプローラー アプリを使用して、コードを記述することなく IVI および VXIplug&play ドライバーでデバイスを管理。
- **Satellite Communications Toolbox** – マルチプラットフォーム シナリオをモデル化し、それらに対する可視化および通信リンク解析を実行。
- **UAV Toolbox** – PX4 ハードウェアインザループ シミュレーションにより、垂直離着陸 (VTOL) UAV 向けのフライト コントローラーを設計および展開。PX4 Cube Orange Plus および Pixhawk 6c Autopilot によるインターフェイス接続。

# Learn more on mathworks.com

The screenshot displays the MATLAB Help Center interface. At the top, there is a blue header with the 'Help Center' logo on the left, a search bar in the center, and 'Help Center' with a search icon on the right. Below the header, a navigation bar contains links for 'Documentation', 'Examples', 'Functions', 'Blocks', 'Apps', 'Videos', and 'Answers'. On the far right of this bar are links for 'Trial Software' and 'Product Updates'. A left-hand sidebar titled 'CONTENTS' includes a 'Documentation Home' link and a 'Category' list with items like MATLAB, Simulink, and various toolboxes. The main content area features a 'Release Notes' section with a 'Select a Product' dropdown and a 'Resources' list. Below these is a 'New Products and Major Updates' section with eight product cards, each with a title and a brief description.

**Help Center** Search Help Center Help Center

Documentation Examples Functions Blocks Apps Videos Answers Trial Software Product Updates

## Release Notes R2024a

**Explore Release Notes**

Select a Product ▾

**Resources**

- Installation and Licensing Changes
- System Requirements
- Bug Reports
- Bug Fixes

### New Products and Major Updates

<b>Bioinformatics Toolbox</b> Read, analyze, and visualize genomic and proteomic data	<b>Computer Vision Toolbox</b> Design and test computer vision systems	<b>Deep Learning Toolbox</b> Design, train, analyze, and simulate deep learning networks	<b>GPU Coder</b> Generate CUDA <sup>®</sup> code for NVIDIA <sup>®</sup> GPUs
<b>Instrument Control Toolbox</b> Control test and measurement instruments and communicate with computer peripherals	<b>Satellite Communications Toolbox</b> Simulate, analyze, and test satellite communications systems and links	<b>Simulink 3D Animation</b> Simulate and visualize dynamic systems in a 3D environment	<b>UAV Toolbox</b> Design, simulate, and deploy UAV applications

# MATLAB EXPO

# Thank you!



© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

