MATLAB EXPO

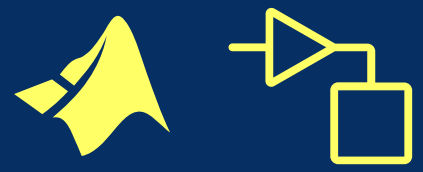
July 11,2024, Bengaluru

What's New in MATLAB & Simulink R2024a

Vijayalayan R, MathWorks

MathWorks

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
```

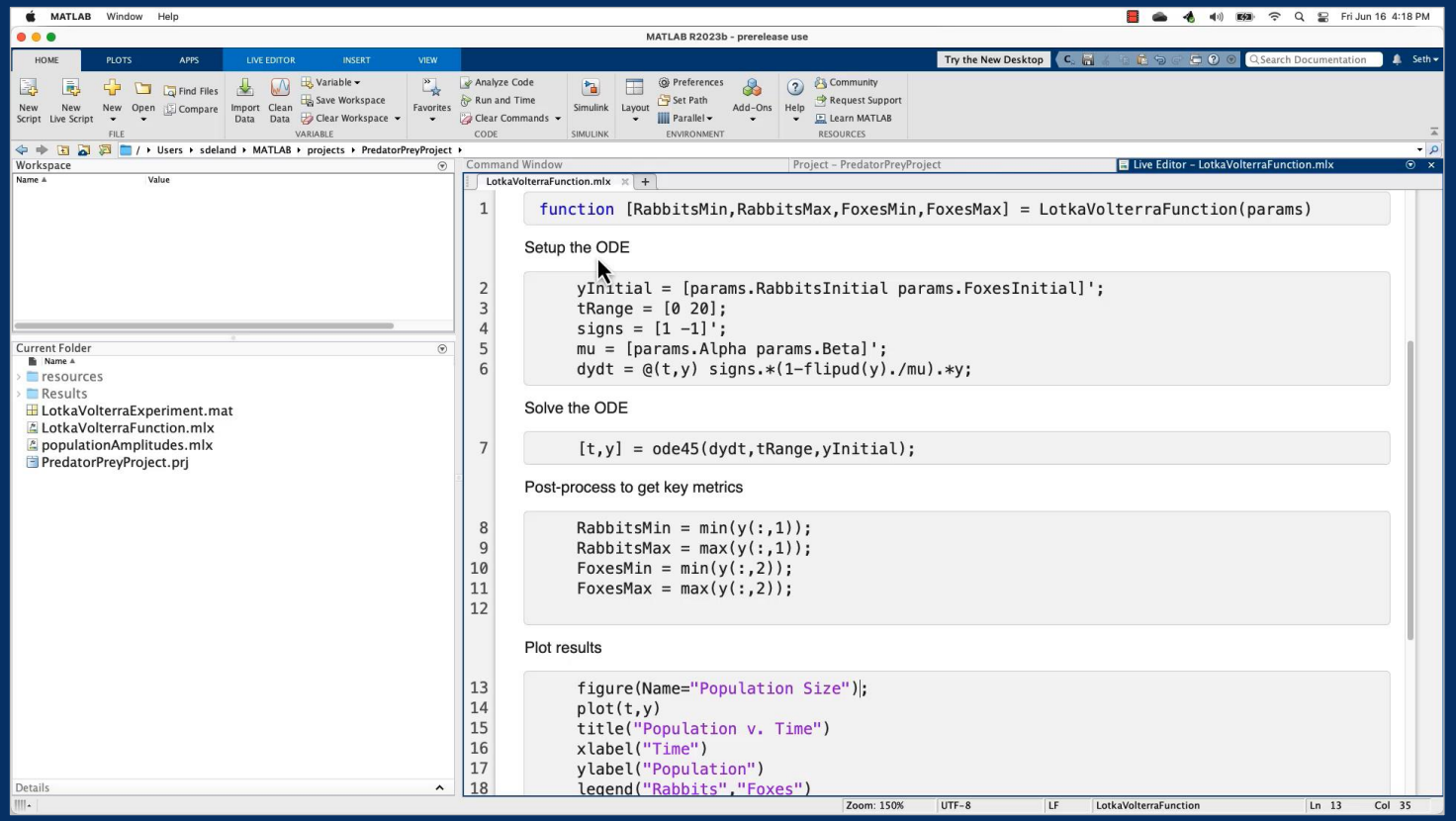
Local Functions: Define functions anywhere in scripts

```
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

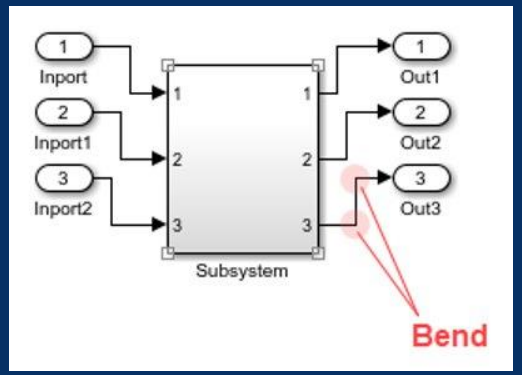
– Visualize, filter, and compare results

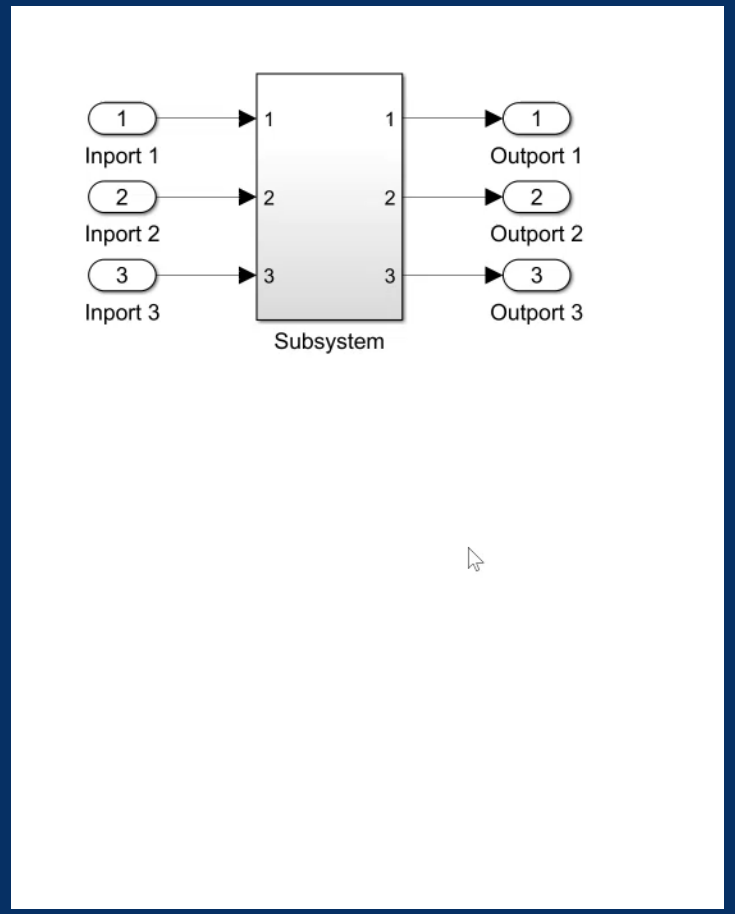


Experiment Manager App

Edit models at the speed of thought

Automatically preserve signal line shape when moving or resizing blocks





Create interactive simulation controls using dashboards

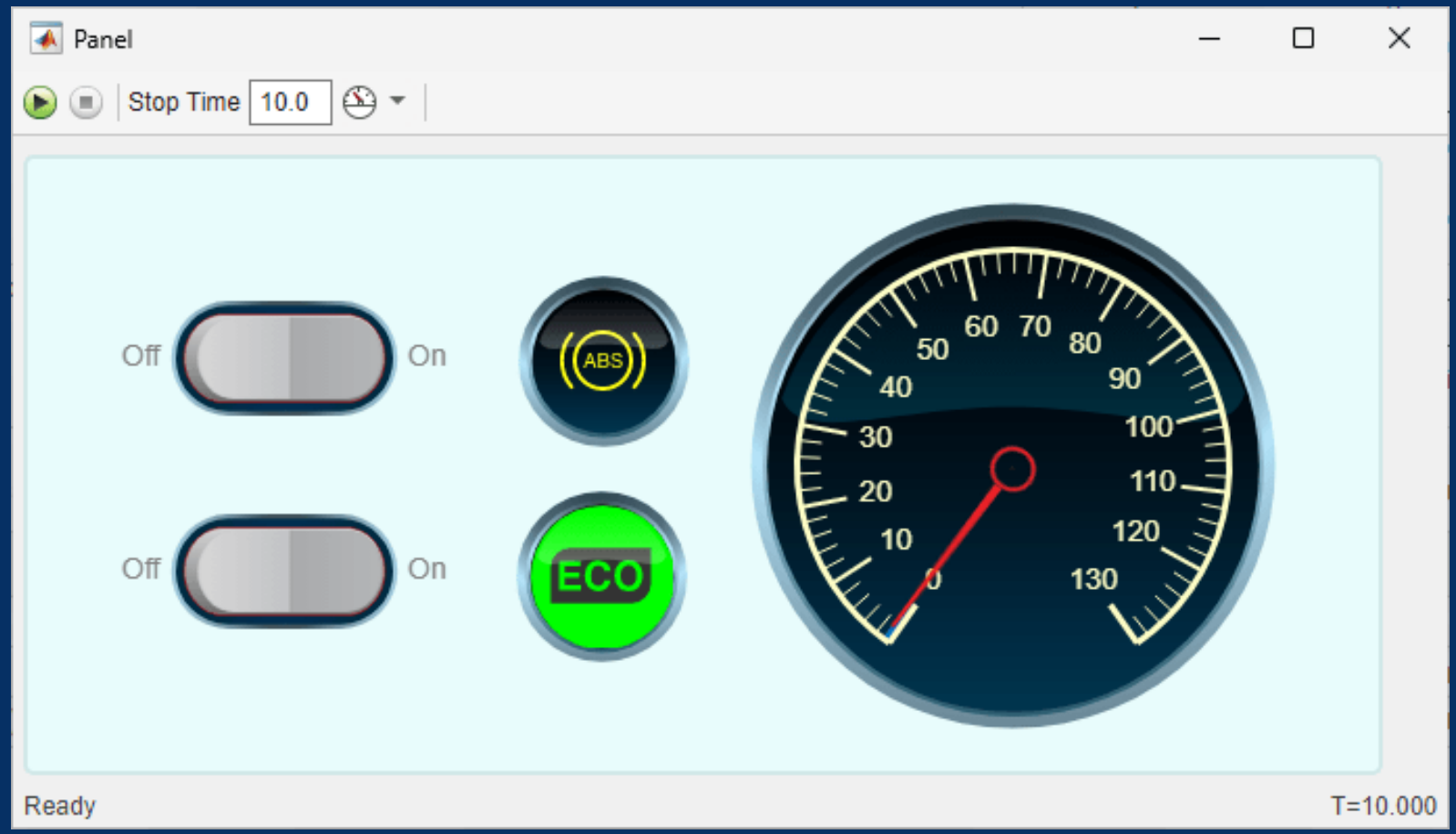
Dashboard blocks

Group and promote dashboard blocks to a panel

Dashboard panel

Deploy interactive simulation controls as apps

Convert dashboard panels into apps



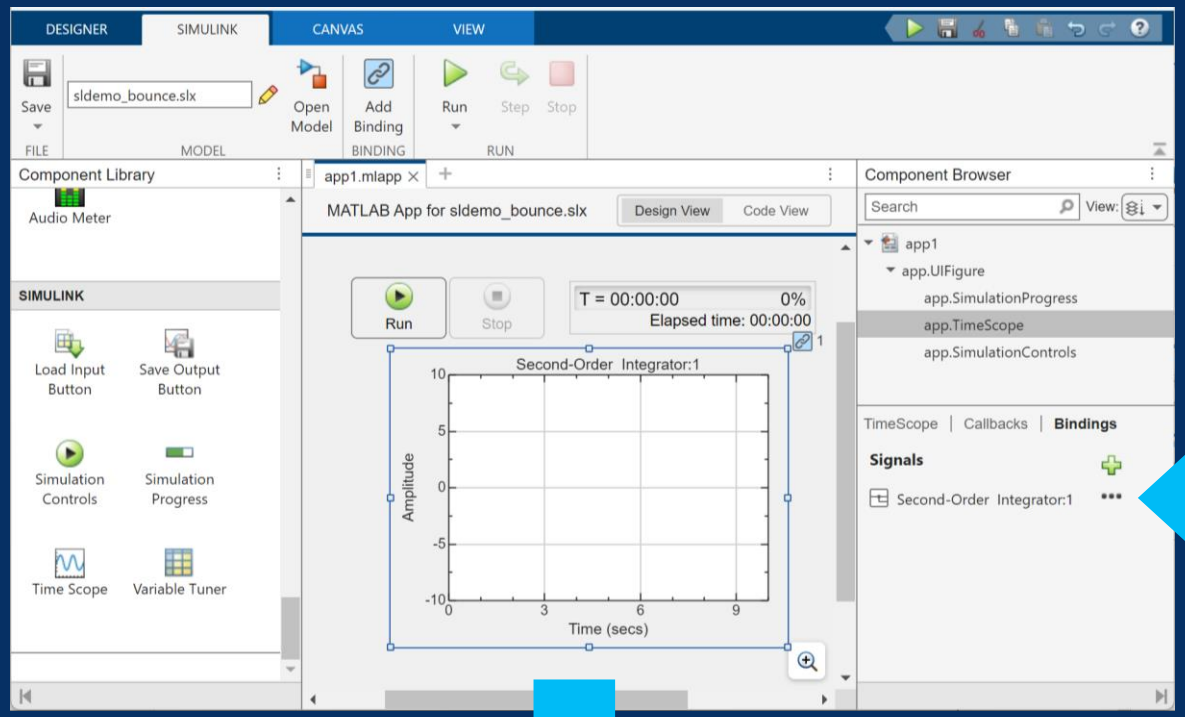
Simulink Compiler

Simulink

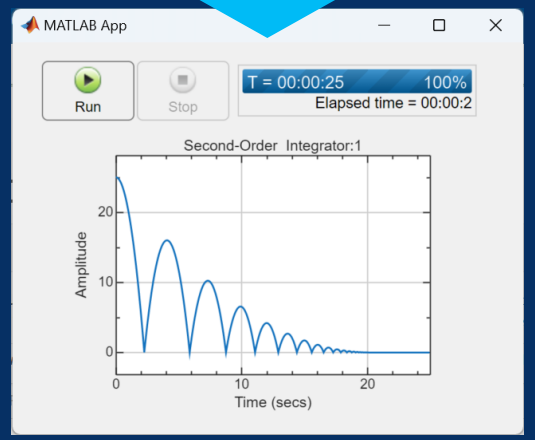
Design simulation apps without writing code

App integration with Simulink models, including signal binding

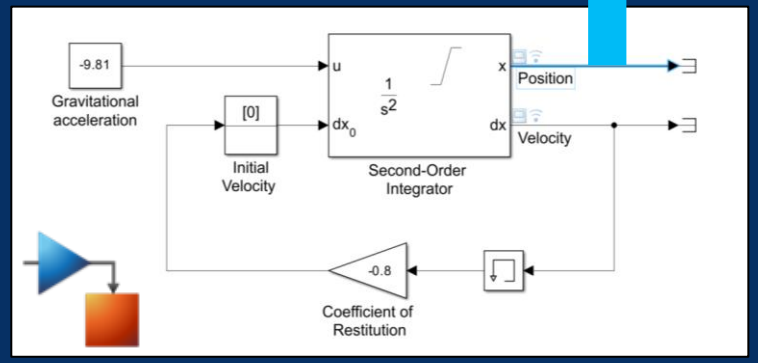
Out-of-the-box, Simulink-specific graphical components



Signal binding with model



Simulation app



MATLAB® & SIMULINK®

Integrations

AI

Ease of Use

Performance

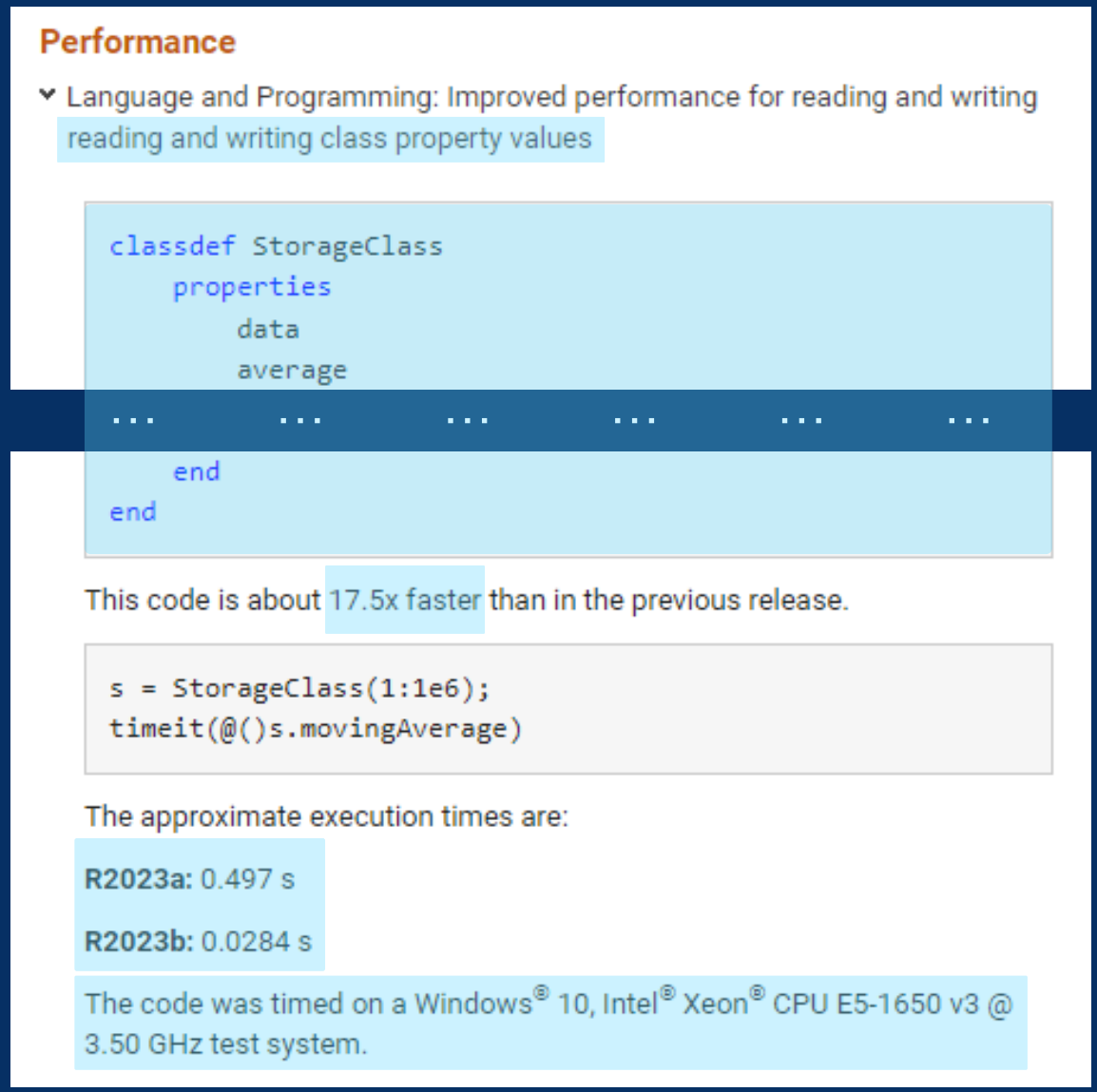
Verification

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.

Apply workflow-specific techniques to improve performance



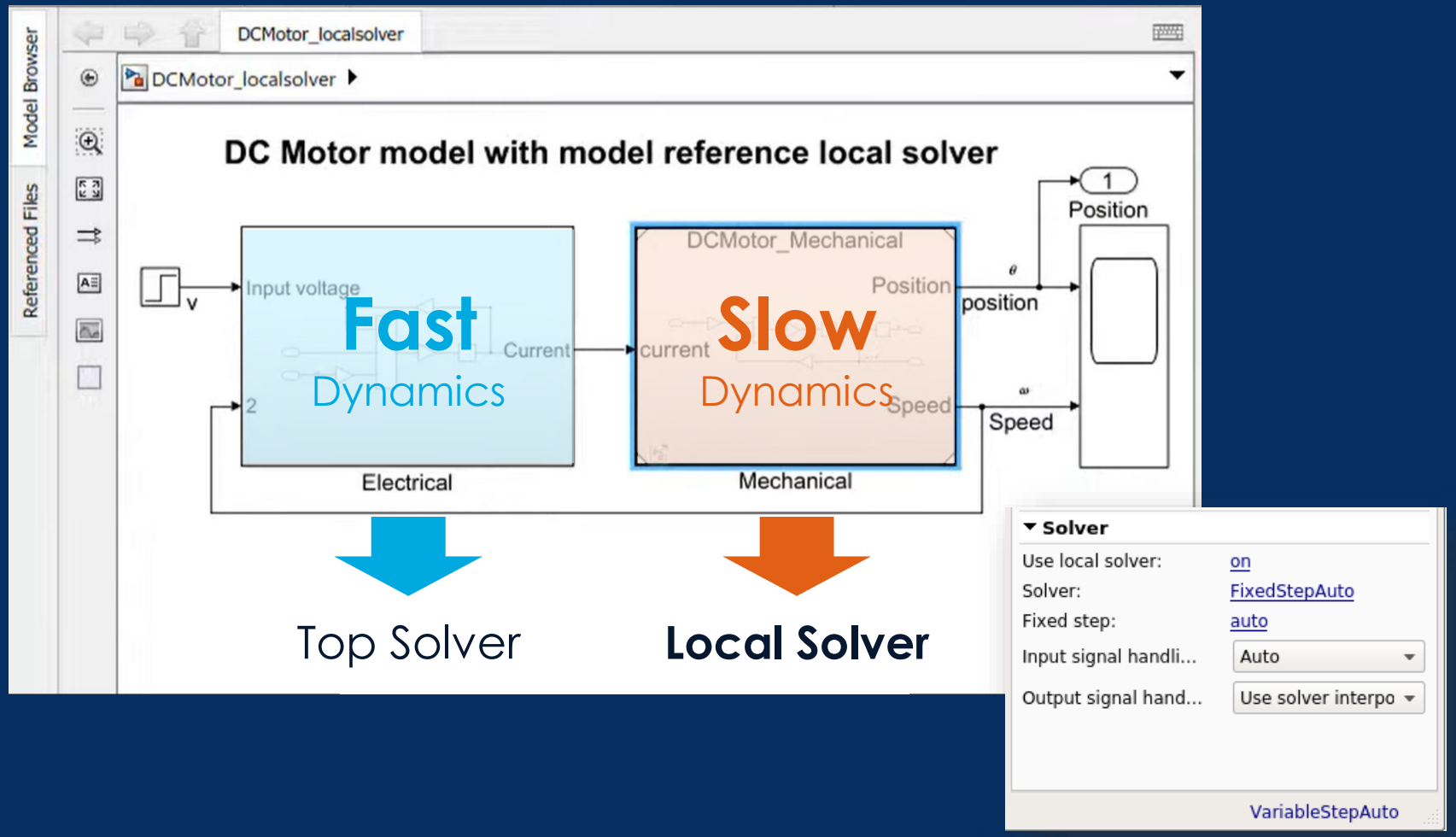




Techniques	Workflow			
	Edit-Update-Repeat	Edit-Sim-Repeat	Tune-Sim-Repeat	Multiple-Sim
Simulation Mode		x	x	x
Fast Restart			x	x
Simulation Cache	x	x	x	x
Model Reference - Parallel Build	x	x		
Model Reference - Incremental Loading & Rebuilding	x	x		
Simulink Profiler	x	x	x	x
Solver Profiler		x	x	x
Modify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

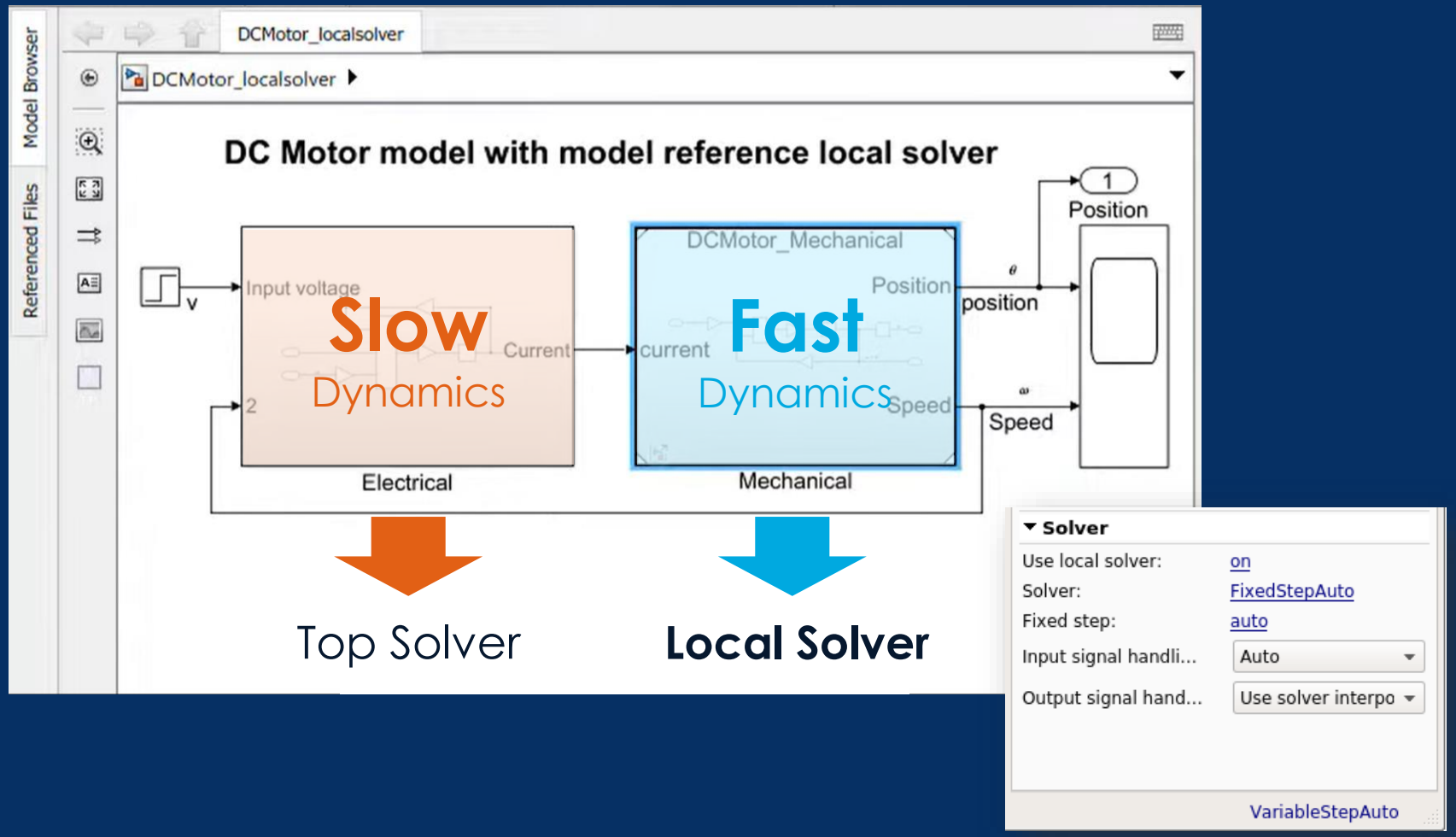
Speed up simulations using the local solver

Decouple systems of different dynamics with the local solver



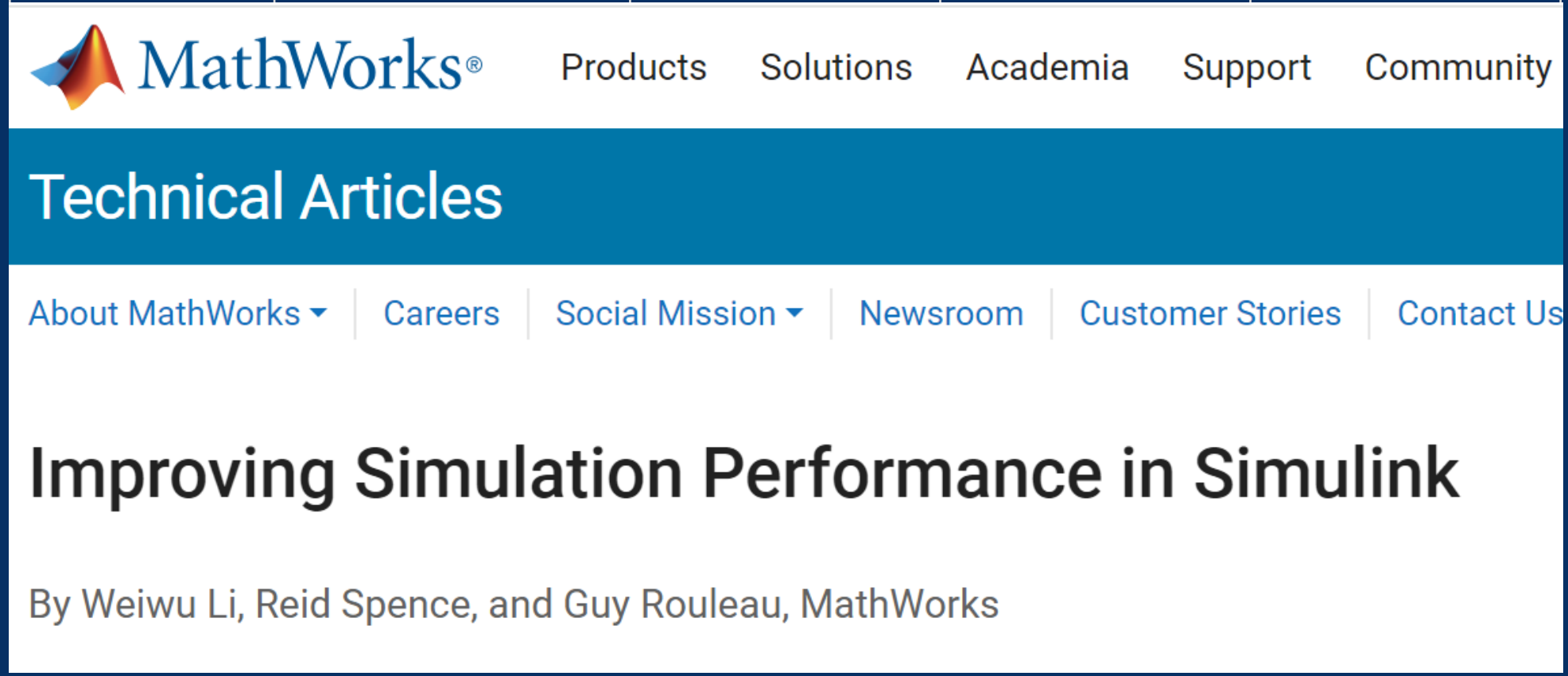
Speed up simulations using the local solver

Support faster local dynamics



Apply workflow-specific techniques to improve performance

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



MATLAB® & SIMULINK®

Integrations

AI

Ease of Use

Performance

Verification

New products

Major updates

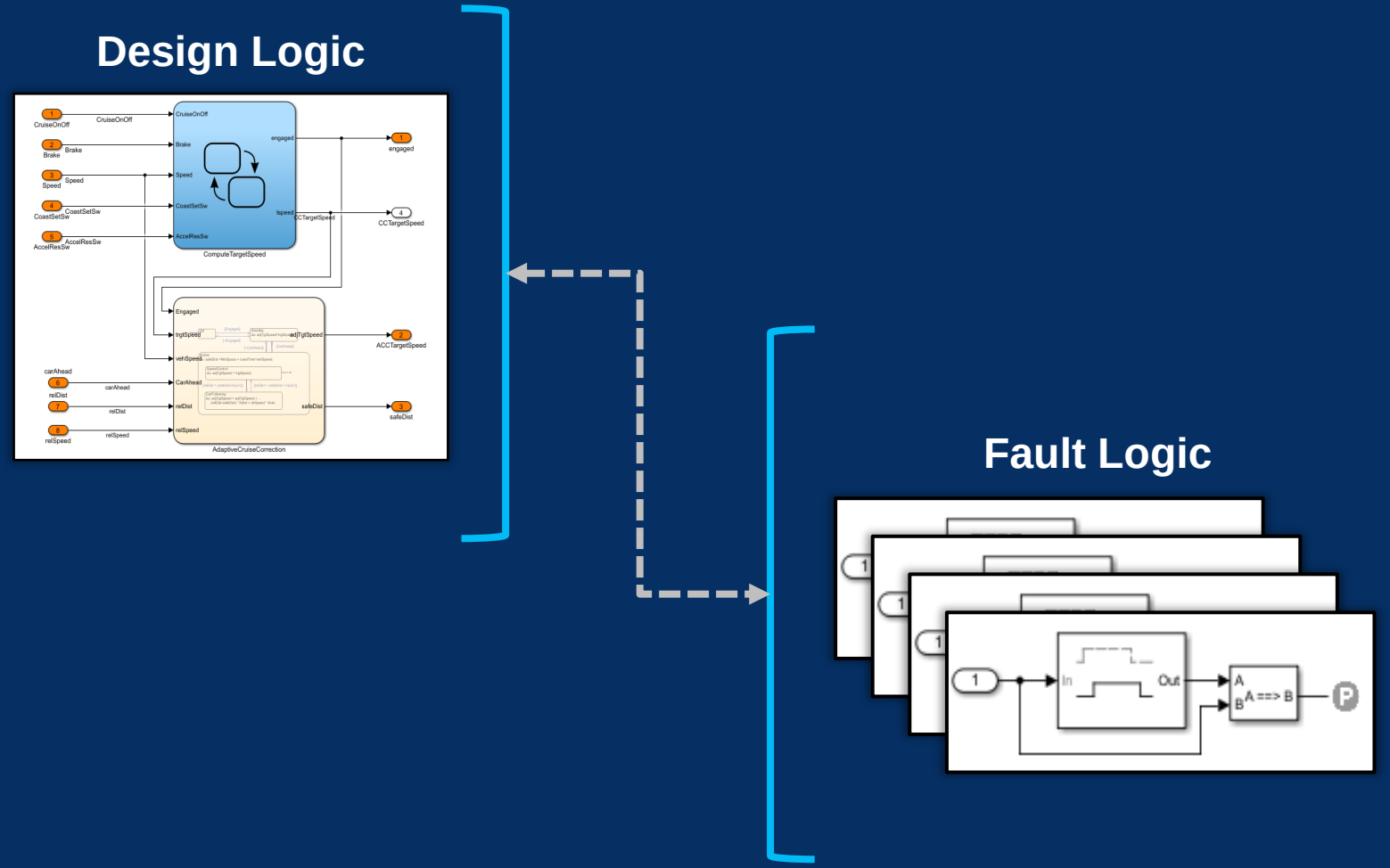
Find defects sooner

Design	Test	Code	Certify
Simulink Design Verifier	Simulink Coverage	Polyspace Bug Finder	DO Qualification Kit
Simulink Check	Simulink Test	Polyspace Code Prover	IEC Certification Kit
HDL Verifier	MATLAB Test	Polyspace Access	Simulink Code Inspector
Simulink Fault Analyzer	Polyspace Test		

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

Fault Table

Fault | Conditional

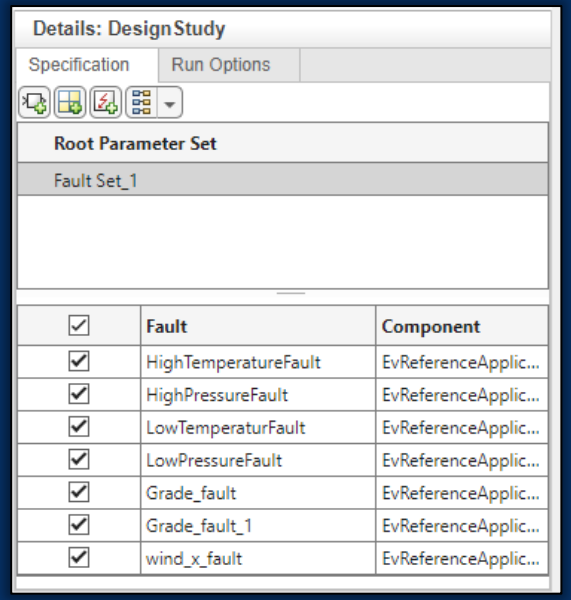
Enable	Model Element/Fault Name	Active Fault	Trigger
☑	Environment/Constant6/Outport/1		
	HighTemperatureFault	☐	Conditional: highSpeedCondition
	LowTemperaturFault	☑	Conditional: SampleConditional
☑	Environment/Constant7/Outport/1		
	HighPressureFault	☑	Timed: 20
	LowPressureFault	☐	Always On
☑	Environment/Constant2/Outport/1		
	Grade_fault	☑	Always On
	Grade_fault_1	☐	Always On
☑	Environment/Constant3/Outport/1		
	wind_x_fault	☑	Always On
☑	Passenger Car/Electric Plant/Simscape/Inductor1/Inductor		
	Inductor1_fault	☑	Behavioral

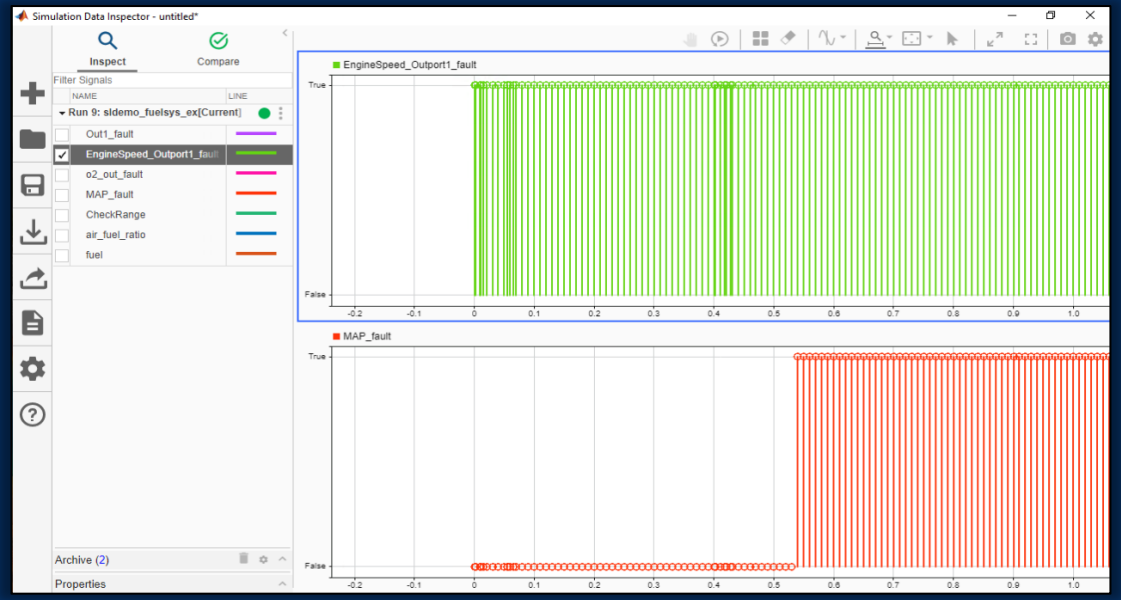
Model faults and analyze effects with Simulink Fault Analyzer

Model faults without modifying the design

Manage faults across multiple domains

Simulate, explore and analyze fault effects





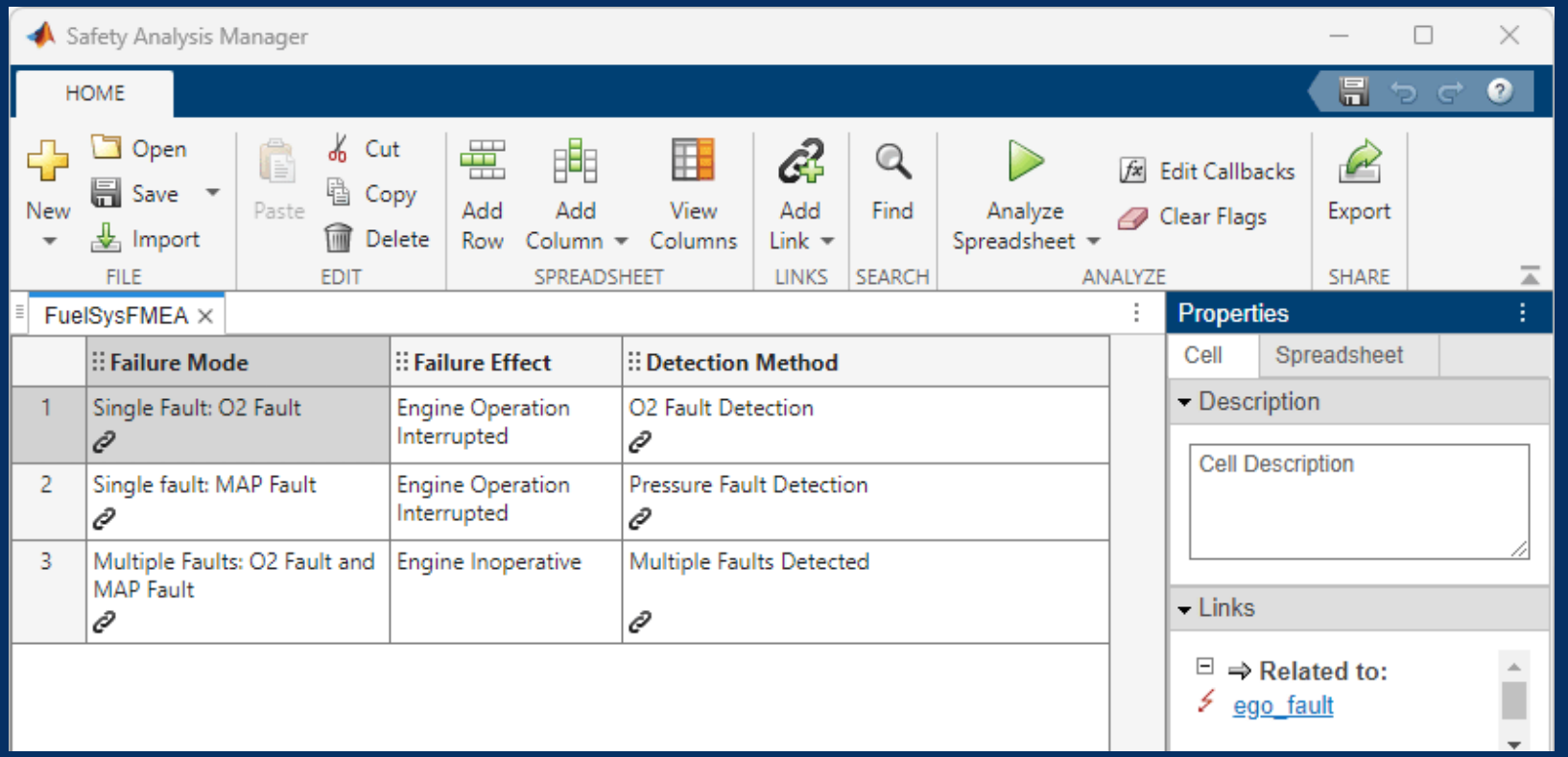
Model faults and analyze effects with Simulink Fault Analyzer

Model faults without modifying the design

Manage faults across multiple domains

Simulate, explore and analyze fault effects

Perform systematic safety analysis



New products

Major updates

Find defects sooner

Design	Test	Code	Certify
Simulink Design Verifier	Simulink Coverage	Polyspace Bug Finder	DO Qualification Kit
Simulink Check	Simulink Test	Polyspace Code Prover	IEC Certification Kit
HDL Verifier	MATLAB Test	Polyspace Access	Simulink Code Inspector
Simulink Fault Analyzer	Polyspace Test		

Requirements Toolbox

Test Product Family

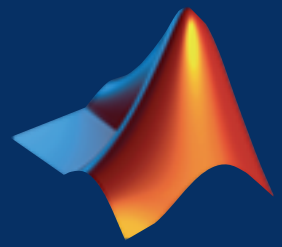
R2015a

Simulink Test

Model Centric

R2023a

MATLAB Test



MATLAB Code Centric

R2023b

Polyspace Test

C/C++ Code Centric

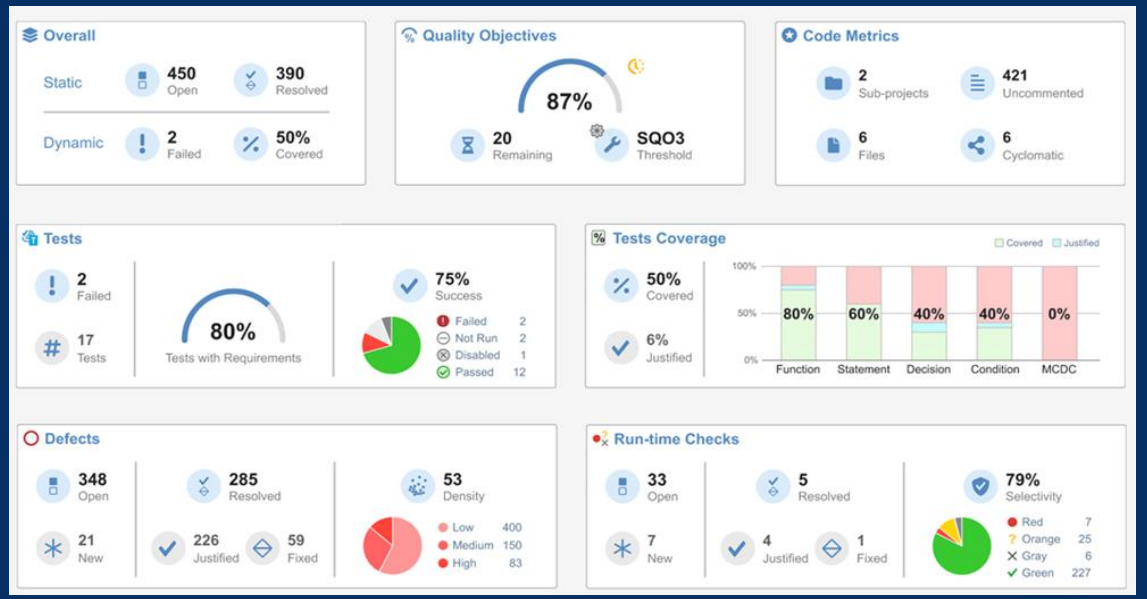
Develop, manage, and execute tests for C and C++ code in embedded systems

Polyspace

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

To know more about embedded software development and verification workflow

- Listen to our experts at Tech Talk on

"Shaping Future Software Factories: Leveraging Model-Based Design for Scalability from Desktop to Cloud" - 15:00–16:00 Hrs

- Engage with our technical experts at Technology Showcase Area:

Model-Based Systems Engineering and Software Engineering

- Model-Based Systems Engineering
- Design High-Performance Compute Applications Using Model-Based Design
-
- Code Verification for Modern Software Development Workflows

MATLAB & SIMULINK

Integrations

AI

Languages

Simulation

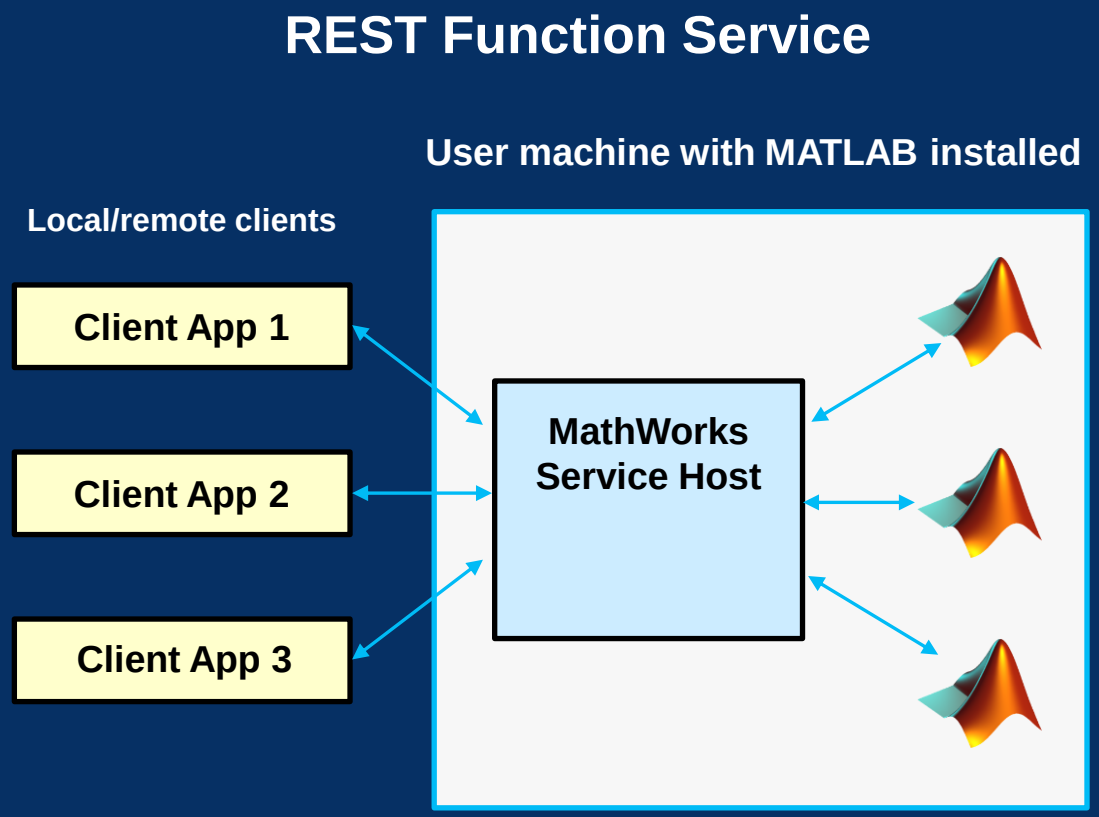
Visualization

Hardware

Call MATLAB from any local or remote client program using REST

Write client programs to call MATLAB using the MATLAB Function Service

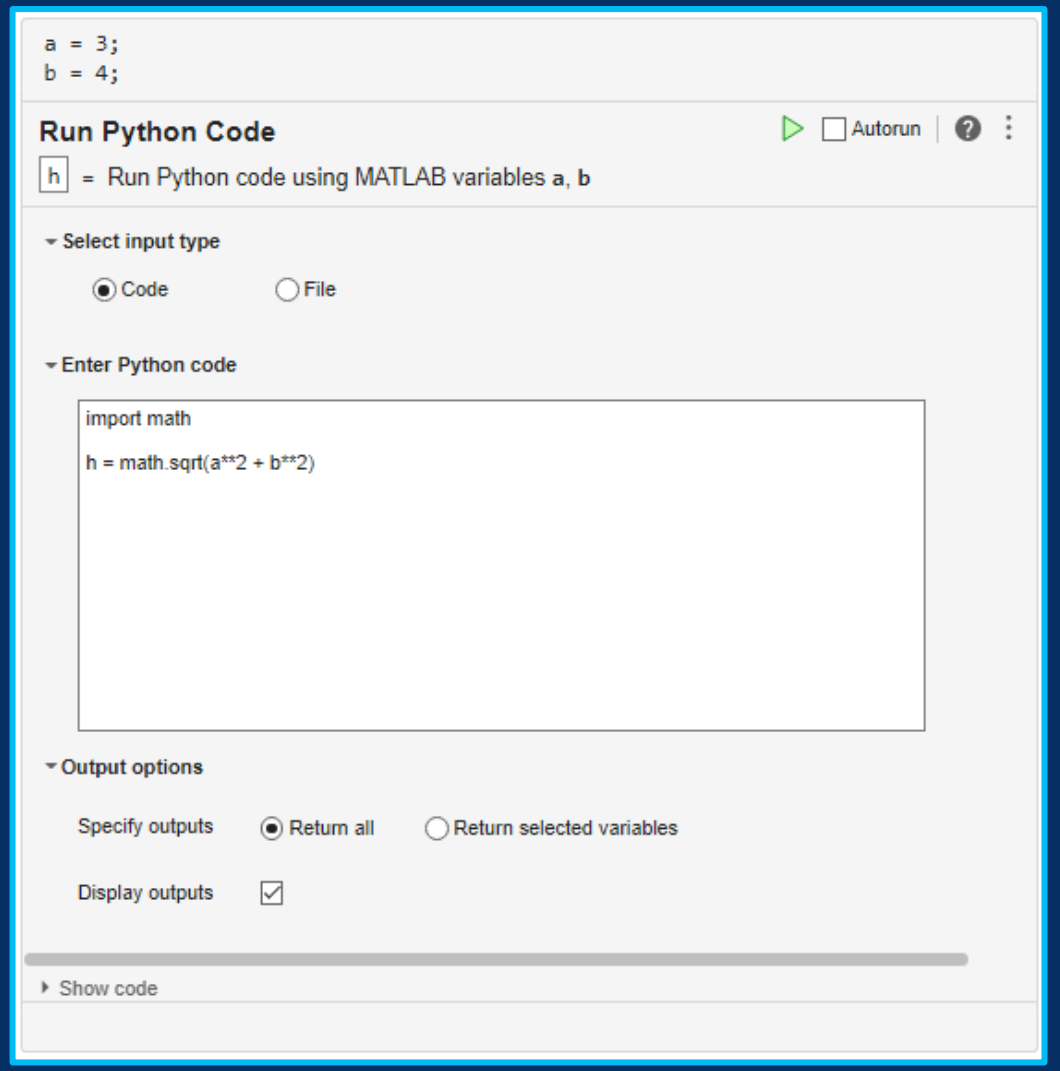
The service uses HTTPS protocol



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



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

R2018b

<FunctionName>

C Caller

R2020a

C

C Function

R2021a

Code Importer

R2022a

C

C Function Supports C++

R2023a

Python Importer

Import Python functions within classes

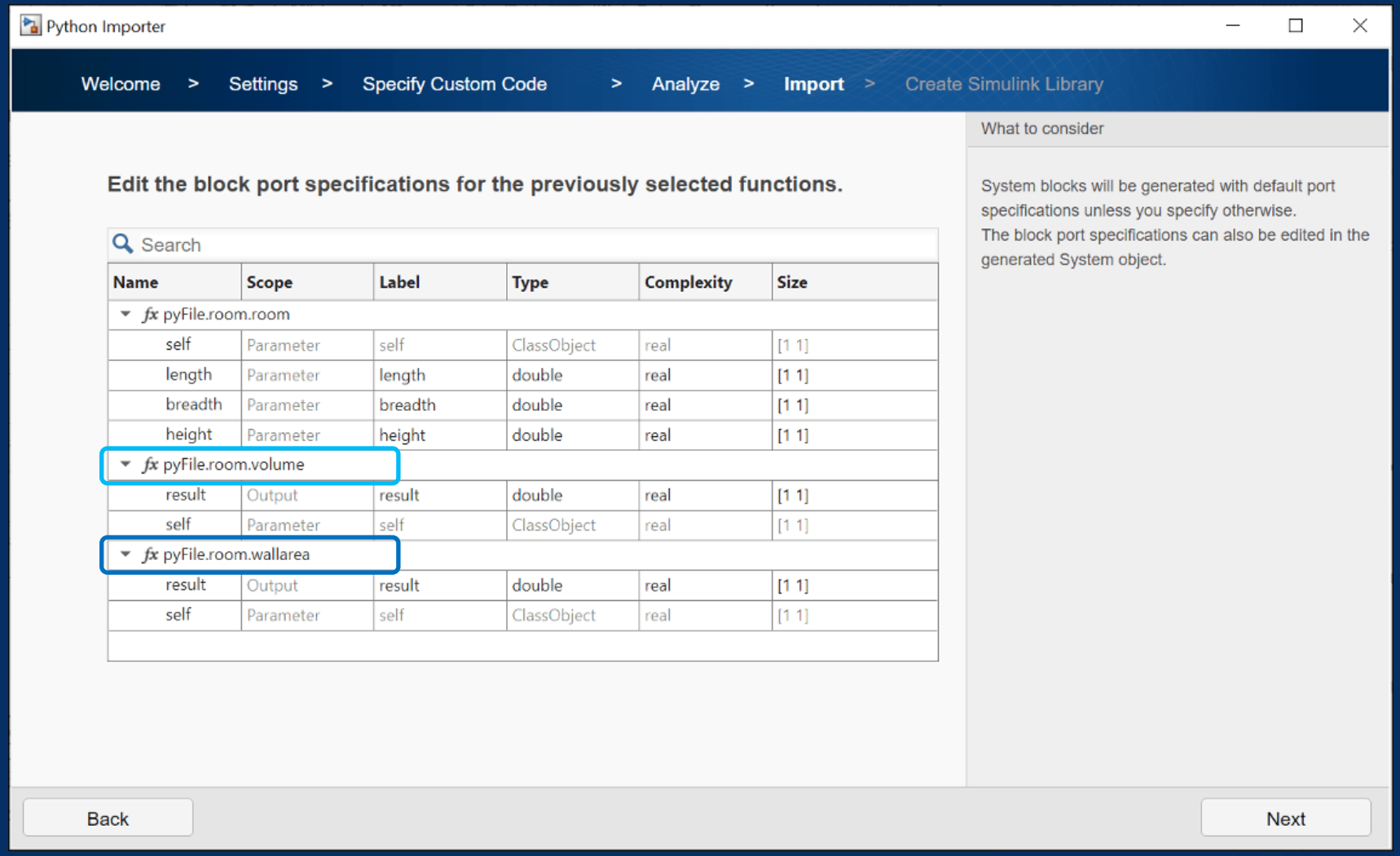
Python importer supports Python functions specified within Python classes

```
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

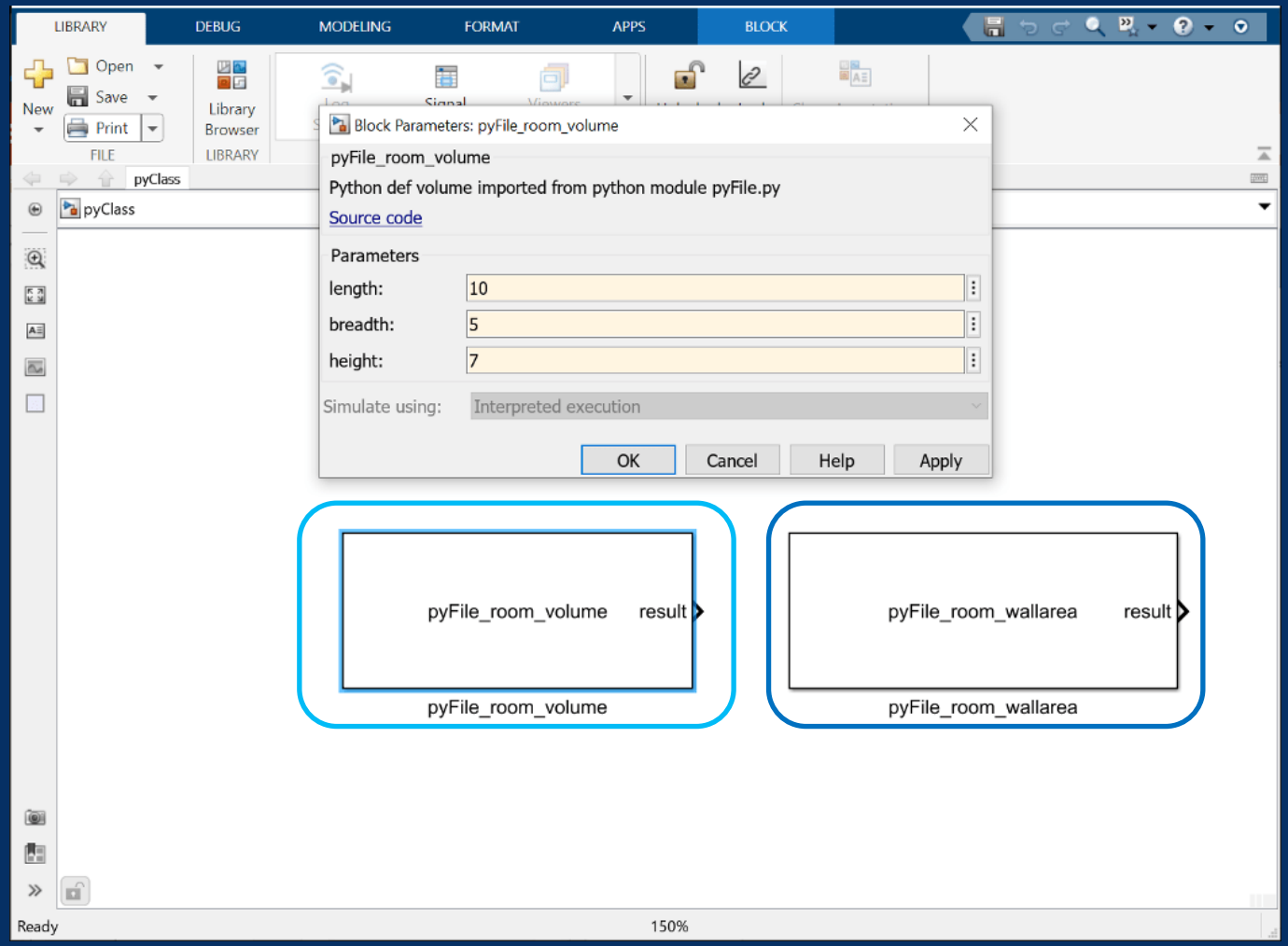
Import Python functions within classes

Wizard UI provides step-by-step import guidance



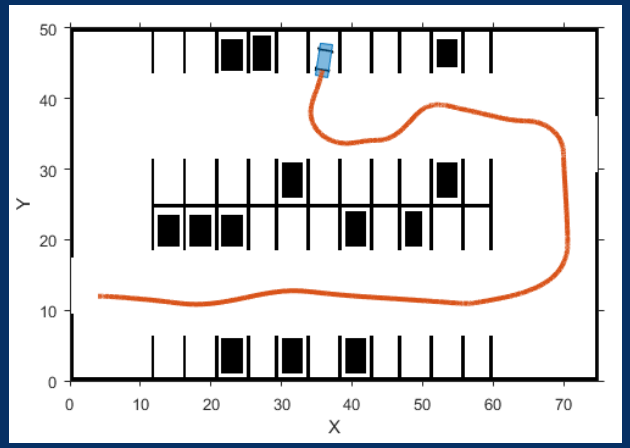
Import Python functions within classes

Export as custom blocksets for simulations

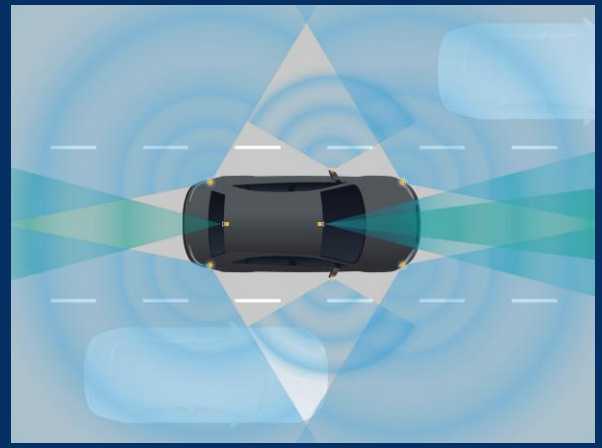


Support for unbounded variable-size signals

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

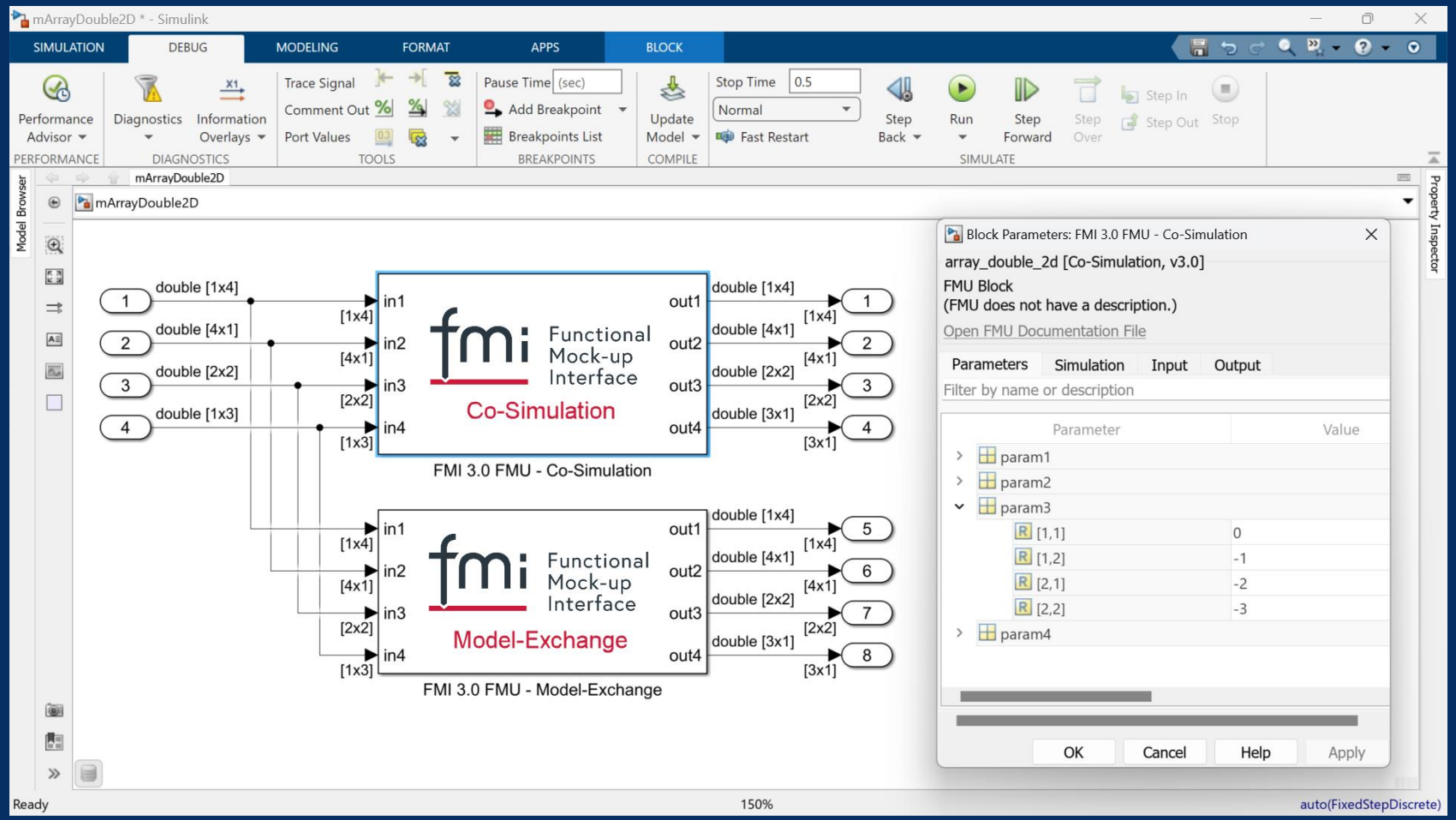
Simulation

Visualization

Hardware

Import FMI 3.0 Function Mockup Units (FMUs)

FMU block supports
FMI version 3.0



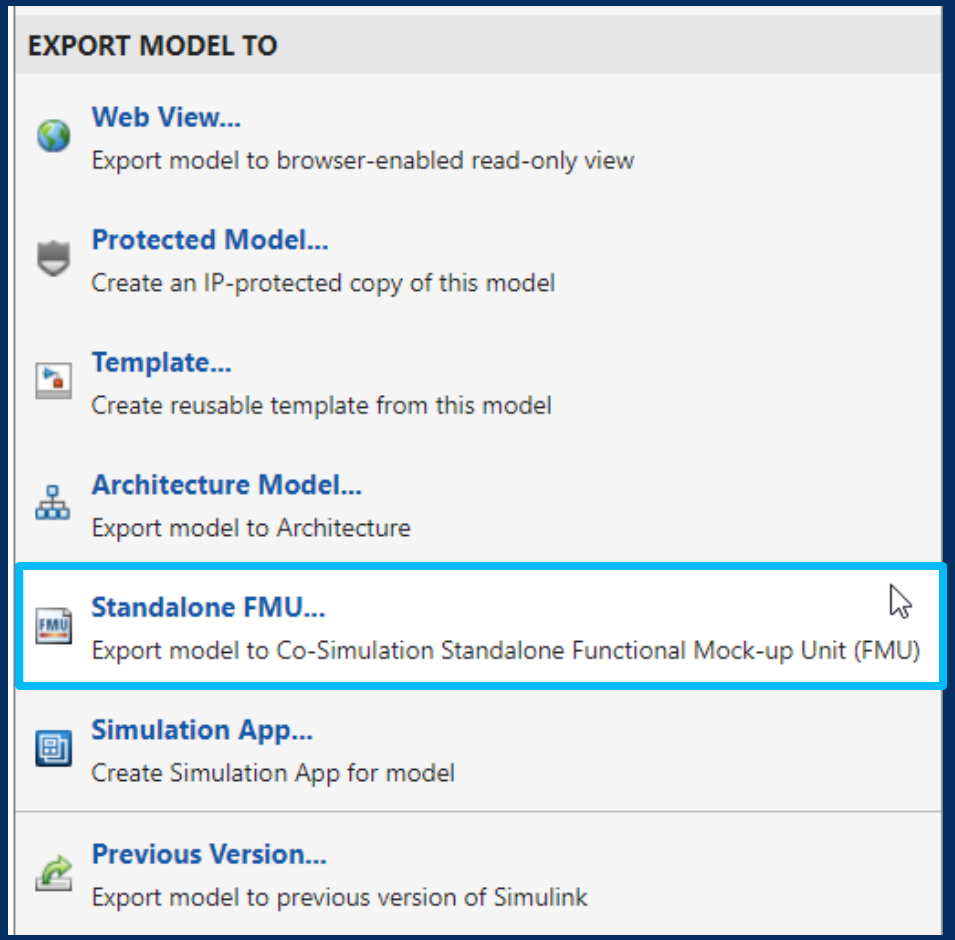
Export simulations as FMI 3.0 FMUs

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

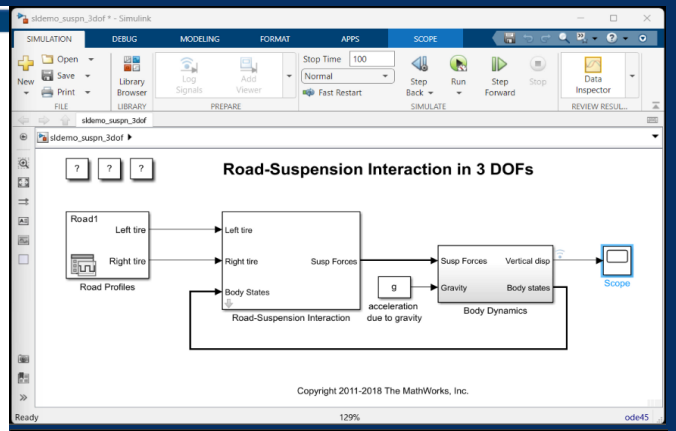
Deployable to other environments with compiler workflows

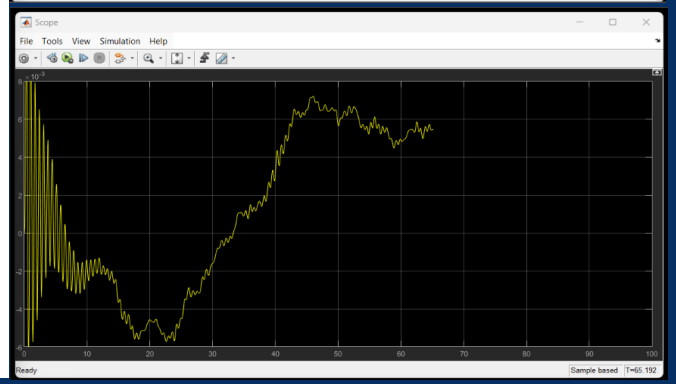
`initialize`
`start`
`step`
`pause`
`resume`
`stop`
`terminate`

```
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



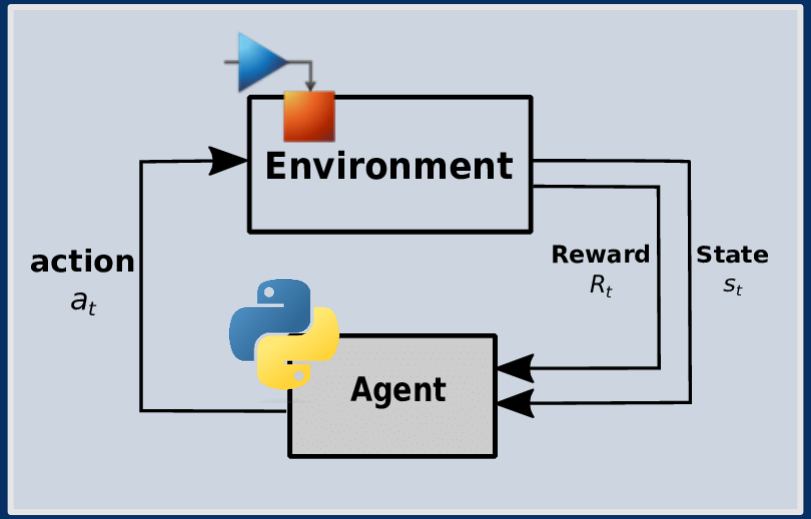


In-simulation outputs

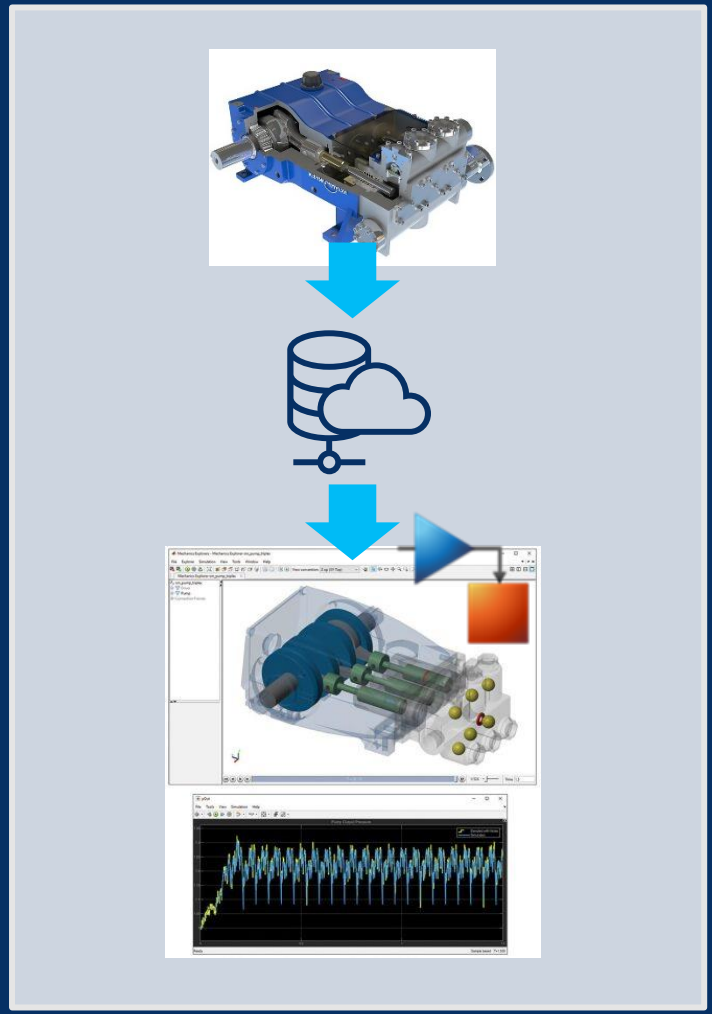
MATLAB EXPO

Control scripted simulations using `Simulation` object

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



Reinforcement Learning



Digital Twins

Simulink Compiler

Simulink

MATLAB®
&SIMULINK®

Integrations

AI

Languages

Simulation

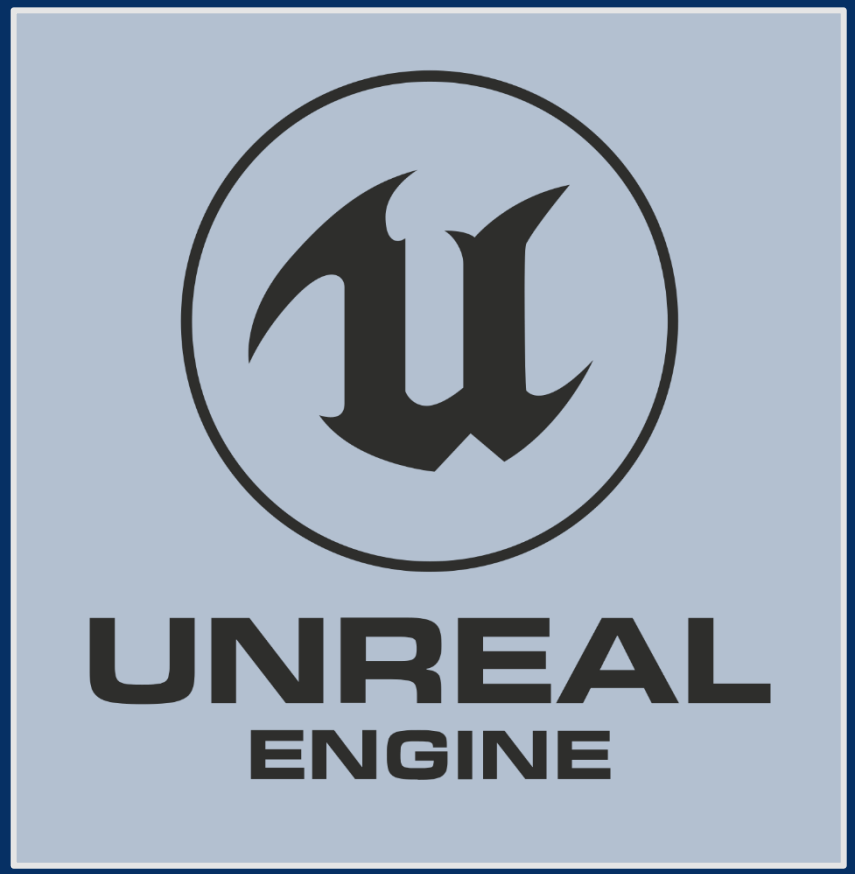
Visualization

Hardware

Visualize 3D simulations

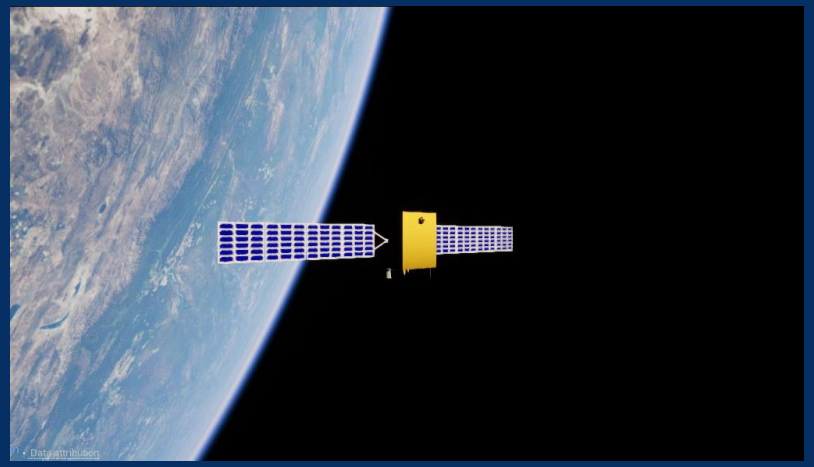
Simulink 3D Animation provides foundational 3D assets, platform, and integration to the Unreal Engine for vertical products

3D Visualizations



Visualize 3D simulations

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



Aerospace Blockset

UAV Toolbox

Automated Driving Toolbox

Vehicle Dynamics Blockset

Robotics System Toolbox

To know more about workflow for integration with simulation engines and visualization environments

- Listen to our experts at Tech Talk on

"Addressing Challenges of Meeting Net-Zero Goals with Simulation and Model-Based Design 14:00–15:00 Hrs"

- Engage with our technical experts at Technology Showcase Area:

Electrification: From Prototyping to Production

- Renewable System Design Using GFM Converter
- Powertrain Design for Transportation Electrification

Virtual Vehicles and Automated Driving

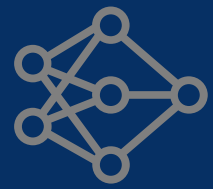
- Automated Driving: Validation and Development
- Virtual Environments for Simulation and Testing
- Virtual Vehicle EV Range Estimation

Communications and Radar Systems: From Design to Deployment

- Space mission modeling and analysis

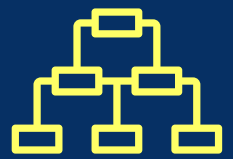
MATLAB®
&SIMULINK®

Integrations



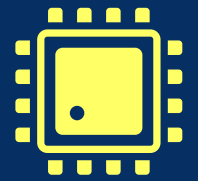
AI

Languages



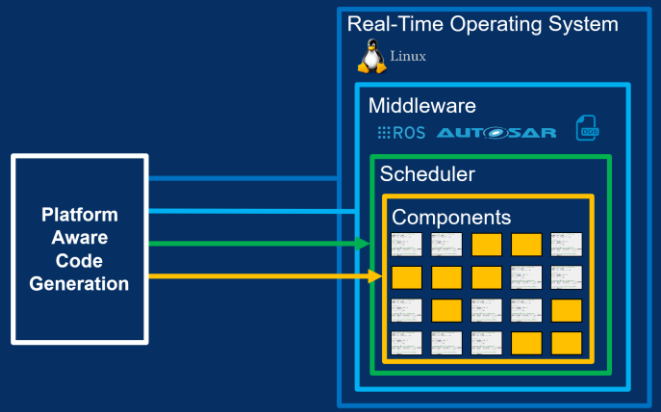
Simulation

Visualization

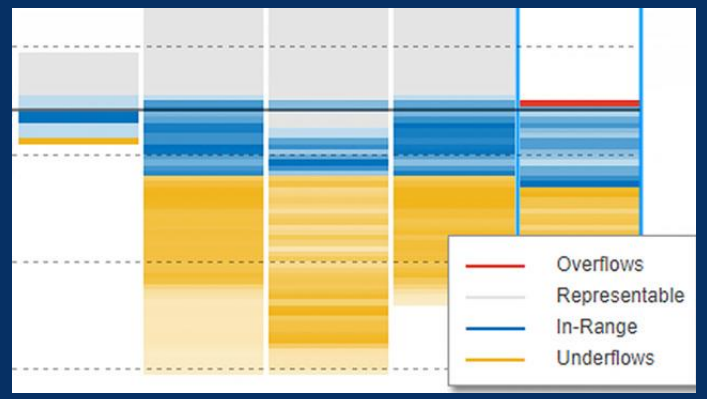


Hardware

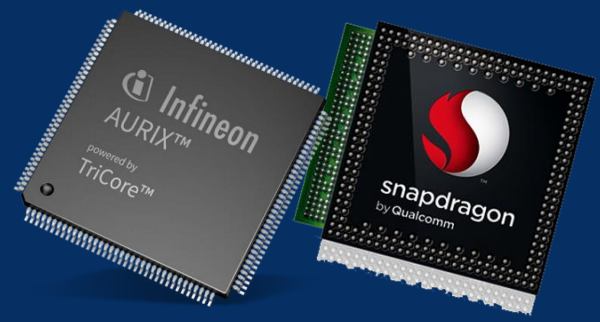
Simulink is for Production Software Development



Platform aware code generation

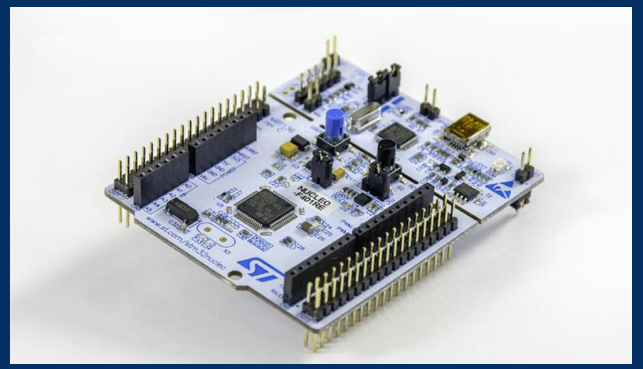


Numeric efficiency leveraging parallelism



Specialized hardware support

Specialized support for different hardware devices and architectures



ST Microelectronics

- Support for dual-core devices
- Support for 4 new device families:U5, L4, L5, WB

Embedded Coder

Embedded Coder Support Package for STMicroelectronics® STM32 Processors

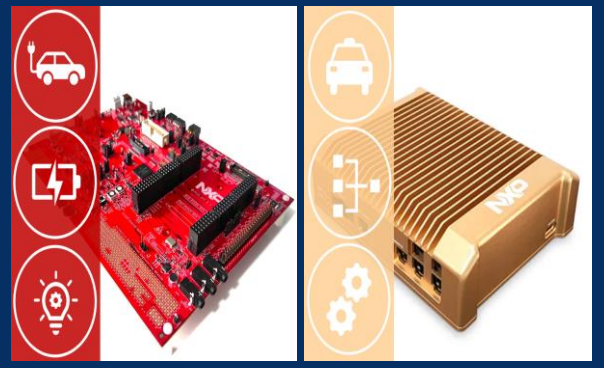
Infineon

- Support for AURIX TC3x
- Support for AURIX TC4x PPU accelerator

Embedded Coder

Embedded Coder Support Package for Infineon® AURIX™ TC3x Microcontrollers

Embedded Coder Support Package for Infineon® AURIX™ TC4x Microcontrollers

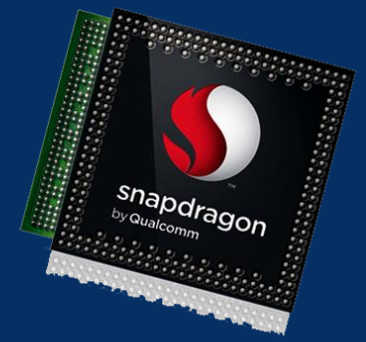


NXP

- Support for S32M2, S32K396, LAX (S32R45), LPC553x, and BMS
- Support for S32K3, S32ZE, and HCP

Embedded Coder

NXP® Model-Based Design Toolbox™

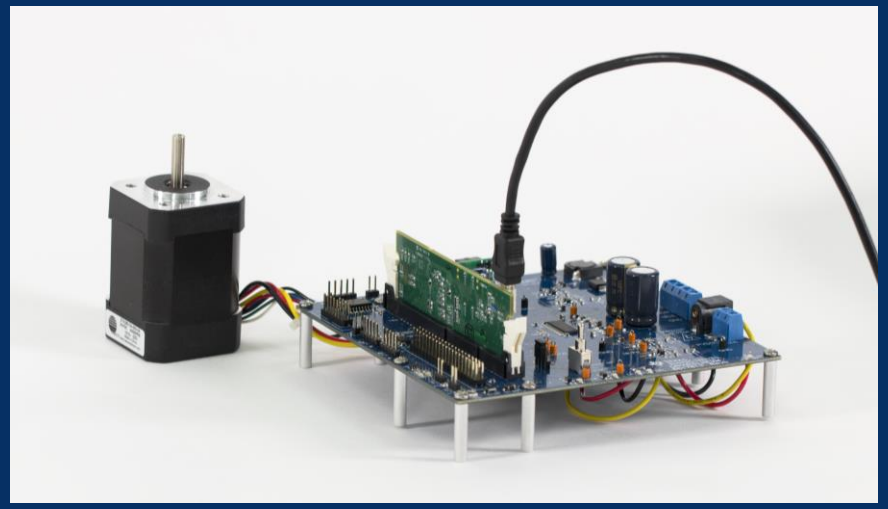


Qualcomm

- Hardware Support Package for Qualcomm Snapdragon Hexagon DSP
- Run Processor-in-Loop (PiL) simulations

Embedded Coder

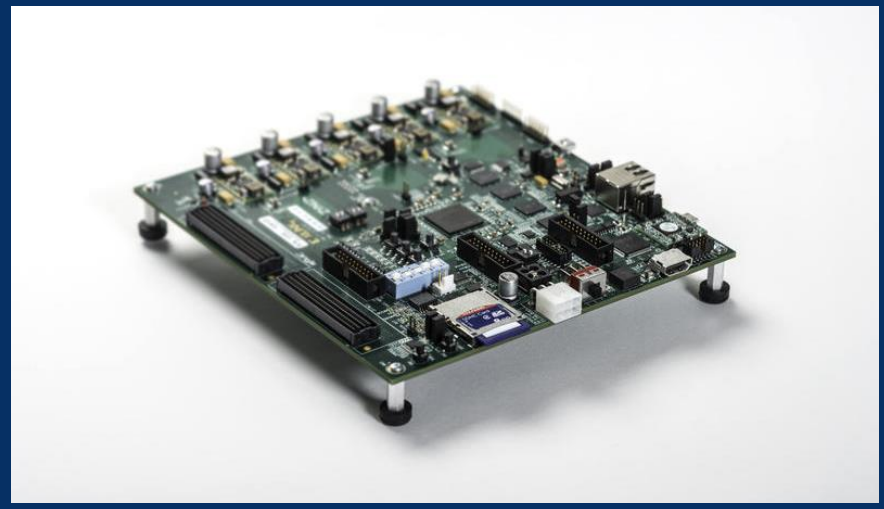
Specialized support for different hardware devices and architectures



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

C2000 Microcontroller Blockset

SoC Blockset, HDL Coder

SoC Blockset Support Package for Xilinx® Devices, HDL Coder Support Package for Xilinx® FPGA and SoC Devices

To know more about integration workflows with hardware platforms

Experience and Engage with our Experts at the below Technology Showcase Areas

Electrification: From Prototyping to Production

- Accelerate Motor Control Development: From Modeling to Deployment
- Accelerate AI-Based Software Development on Infineon AURIX TC4x
- Design, Prototype, and Test Your Brushless DC Motor Controls Using Simulink and Speedgoat Hardware

Artificial Intelligence - From Design to Deployment

- Visual Inspection with Beckhoff PLC

Communications and Radar Systems: From Design to Deployment

- Live Stream Radar Data to MATLAB from TI mmWave Radars for Signal Processing and Tracking Applications
- Rapid Prototyping of Radar and Wireless Communications Systems on RFSoCs

MATLAB®
&SIMULINK®

Integrations

AI

New capabilities across the entire AI workflow

Data Preparation

AI Modeling

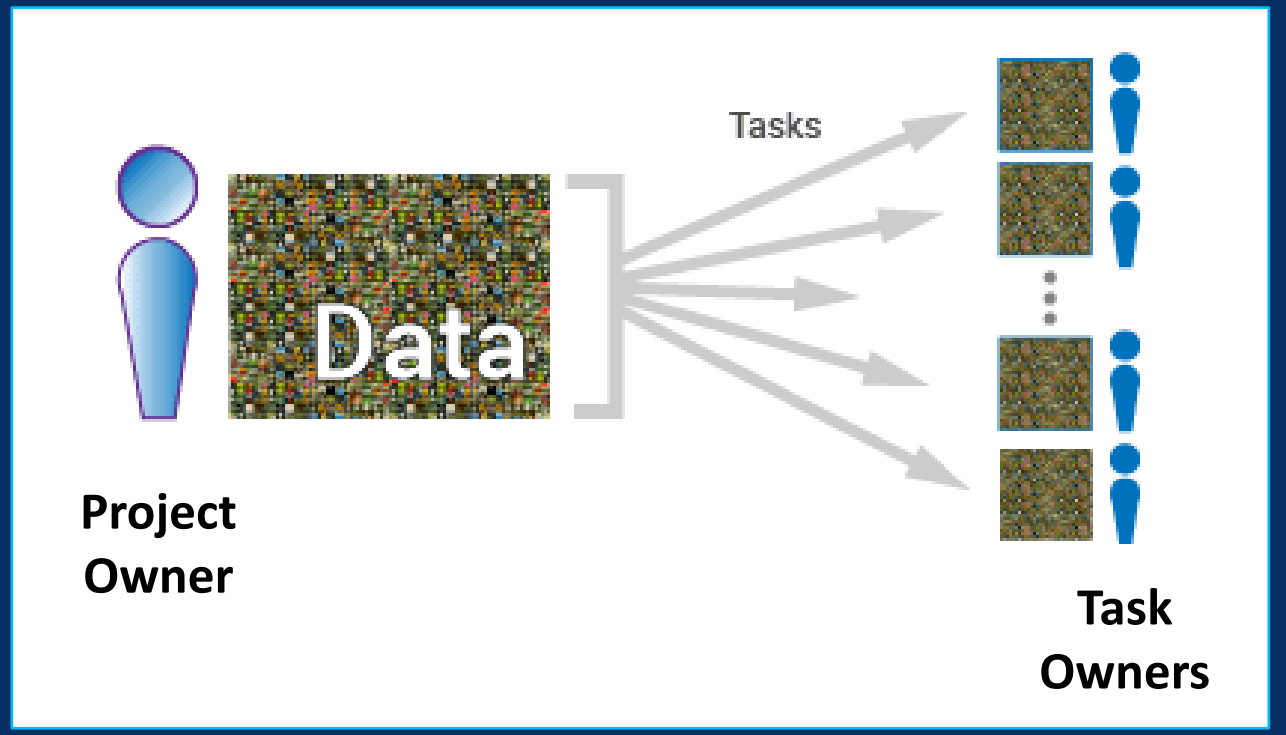
Simulation & Test

Deployment

Collaborative, multiuser, team-based labeling

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

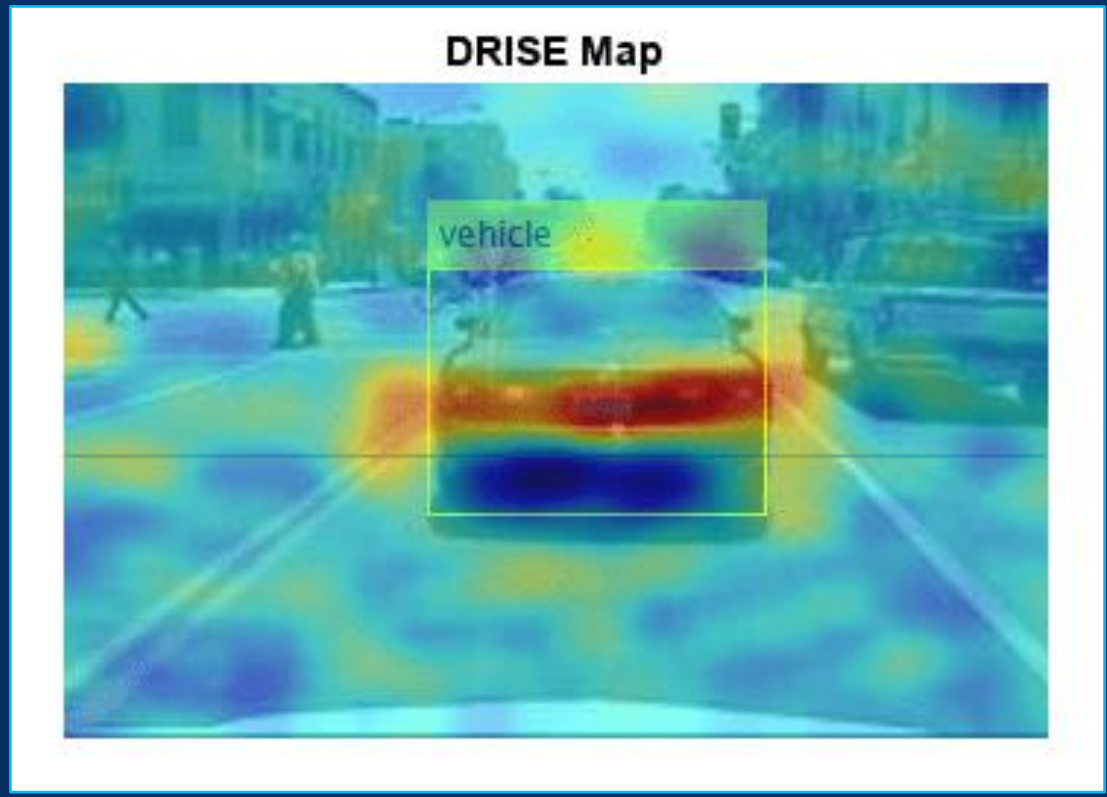


Execute Python deep learning models in Simulink

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

Explain object detection network predictions using D-RISE

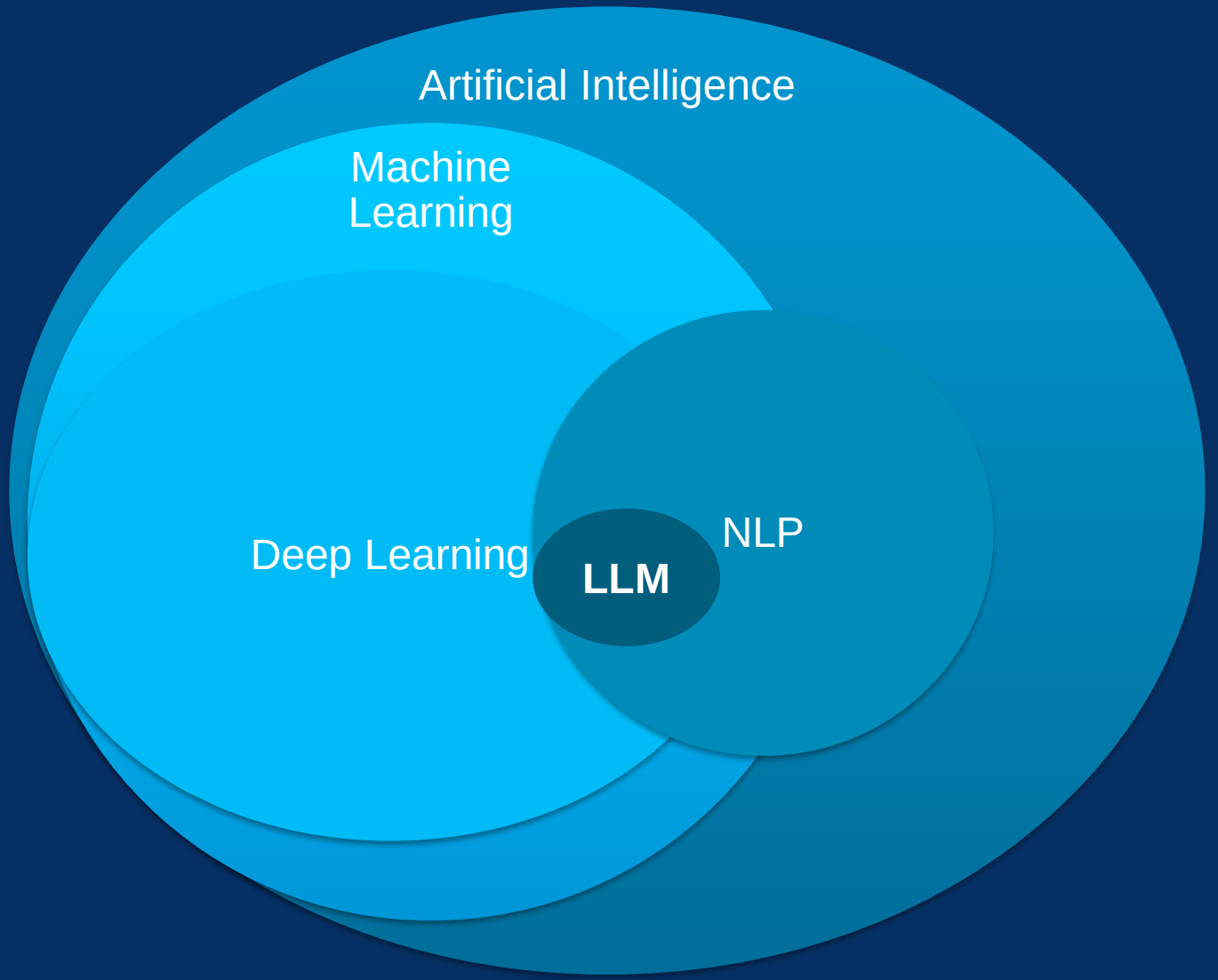
Generate visual explanations for the prediction results of object detection networks



Generate generic CUDA code for deep learning

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

NLP, LLMs, and how these relate to AI



AI Chat Playground - An experimental generative AI offering from MathWorks

Playground

Hello! Whether you're checking out MATLAB for the first time or years, I'm here to answer your questions and provide coding tips

Ran in R2023a

DF Deborah Ferreira

How can I compute the eigenvalues for a matrix?

AI

To compute eigenvalues for a matrix in MATLAB, you can use the `eig()` function. Here's an example code:

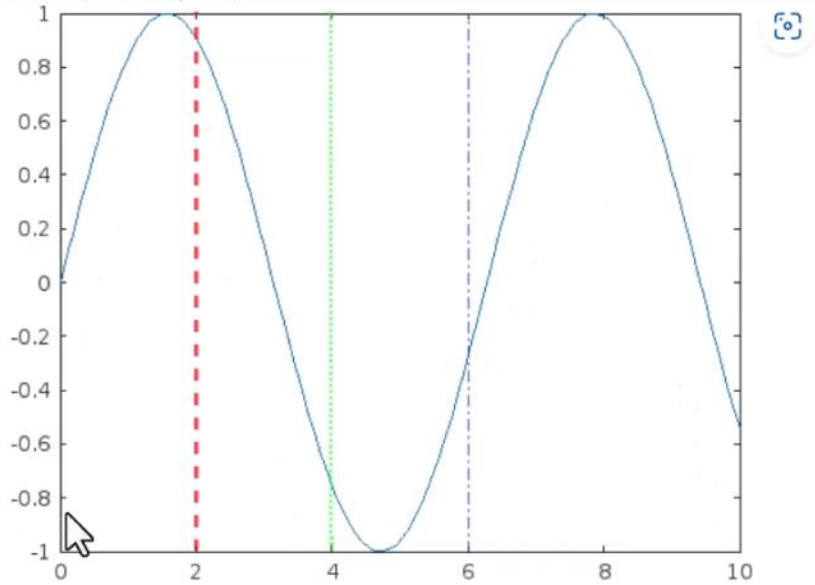
```
A = [1 2; 3 4];
eigenvalues = eig(A);
disp(eigenvalues);
```

This will output the eigenvalues for matrix `A`. What happens when you try to compute eigenvalues for a non-square matrixfp>

How can you verify the correctness of your code?

Send a message...

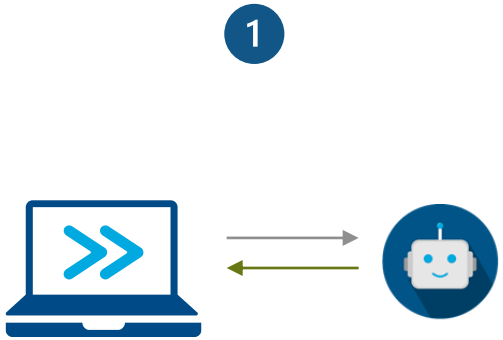
Generated code and text might be inaccurate. Validate before use.



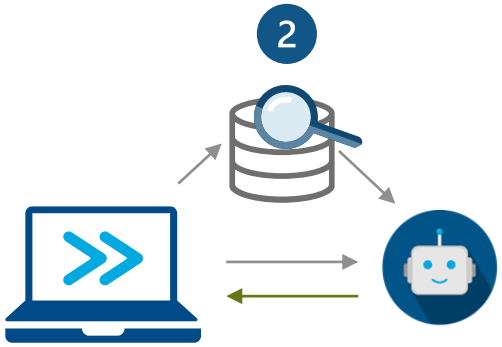

```
A = [1 2; 3 4];
eigenvalues = eig(A);
disp(eigenvalues);
```

```
-0.3723
 5.3723
```

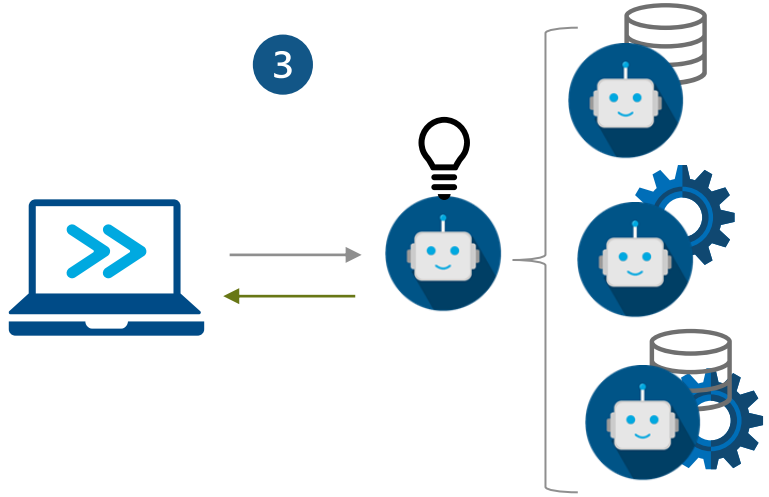
Incorporating LLMs in your workflow



Prompt Engineering



Retrieval-Augmented Generation



Continue pretraining or fine-tune

GitHub Repo: [llms-with-matlab]

To know more about AI workflows

Listen to our experts at Tech Talk on :

“The Industrial AI Life Cycle: Dreaming, Designing, and Delivering in the Digital Age 14:00–15:00 Hrs

Experience and Engage with our Experts at the Technology Showcase Areas on :

Artificial Intelligence - From Design to Deployment

- Predictive Maintenance Using Data Acquisition Systems
- AI for Design Optimization and Model Calibration
- Visual Inspection with Beckhoff PLC

MATLAB®
&SIMULINK®

- Ease of Use
- Performance
- Verification

Integrations

- Languages
- Simulation
- Visualization
- Hardware

AI

What's new in Training Services

Onramps
(Total - 22)

- Power Systems Simulation Onramp
- System Composer Onramp
- Simscape Battery Onramp

Online Training Courses

- Individual self-paced courses to Online Training Suite
  - 11 Learning paths, 36 smaller courses

Instructor Led Trainings
(Total - 62)

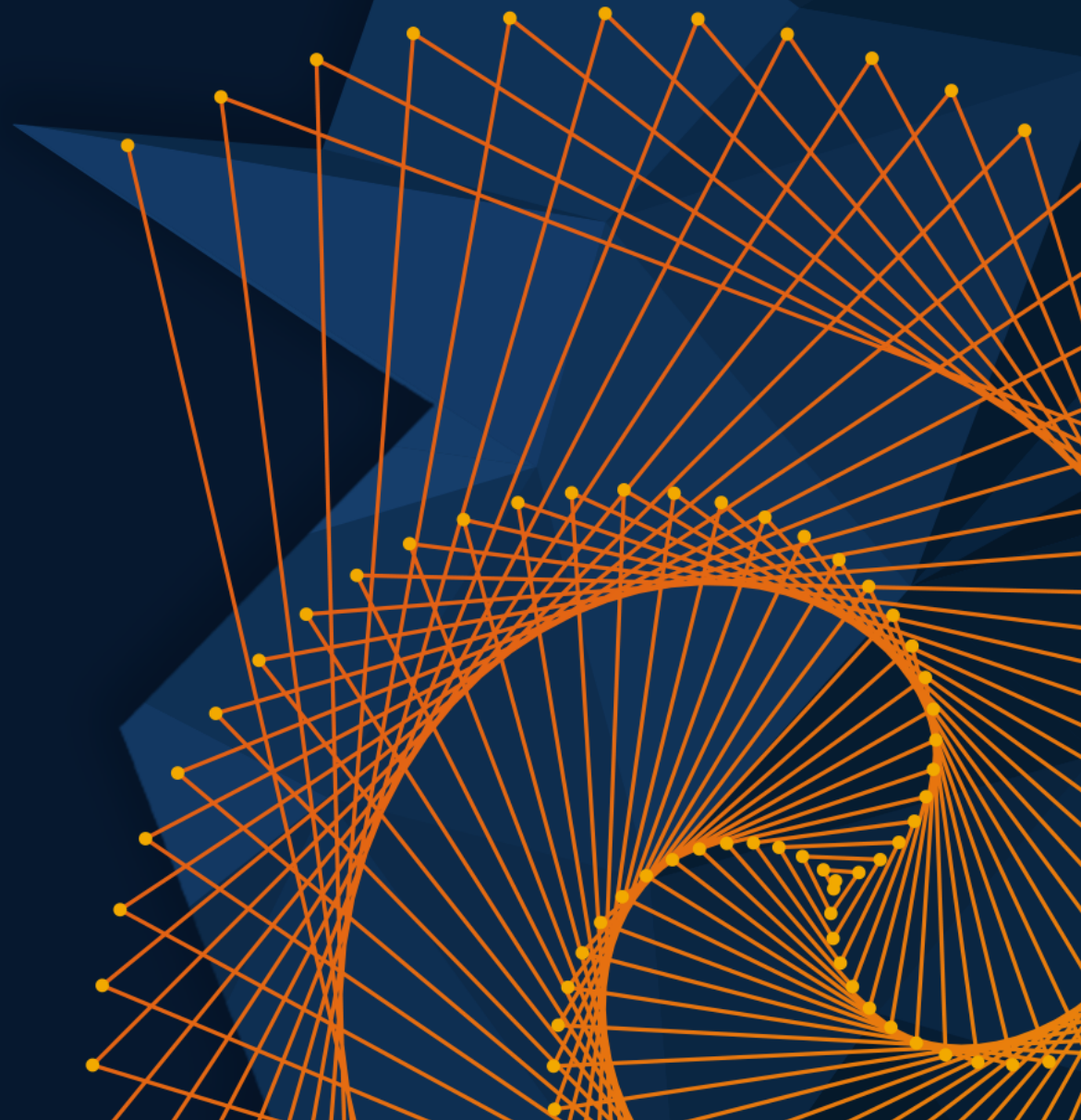
- MATLAB and Python Integration
- Deep Learning for Signals in MATLAB
- Motor Control with Simulink and Simscape
- Simulink for Analog and Mixed-Signal Verification
- Polyspace for C/C++ Code Verification (new tool)
- Reviewing Polyspace Results

Certification Existing MATLAB Associate & Professional)

- Simulink Associate

MATLAB EXPO

Thank you!



MathWorks®