Building Your Virtual Vehicle Simulation with Simulink

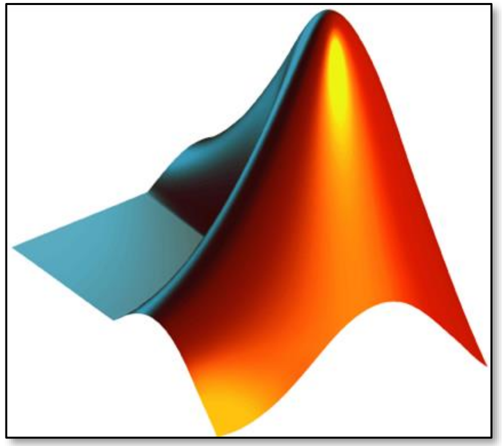
Brad Hieb
Application Engineer
bhieb@mathworks.com

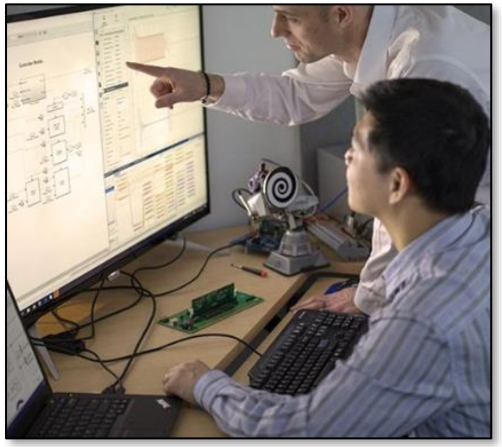
Mike Sasena, PhD
Automotive Product Manager
msasena@mathworks.com

Key Takeaways

MathWorks provides a **powerful platform** for building your **Virtual Vehicle**

Our platform is very **flexible**, and we can help you **customize** it for your needs

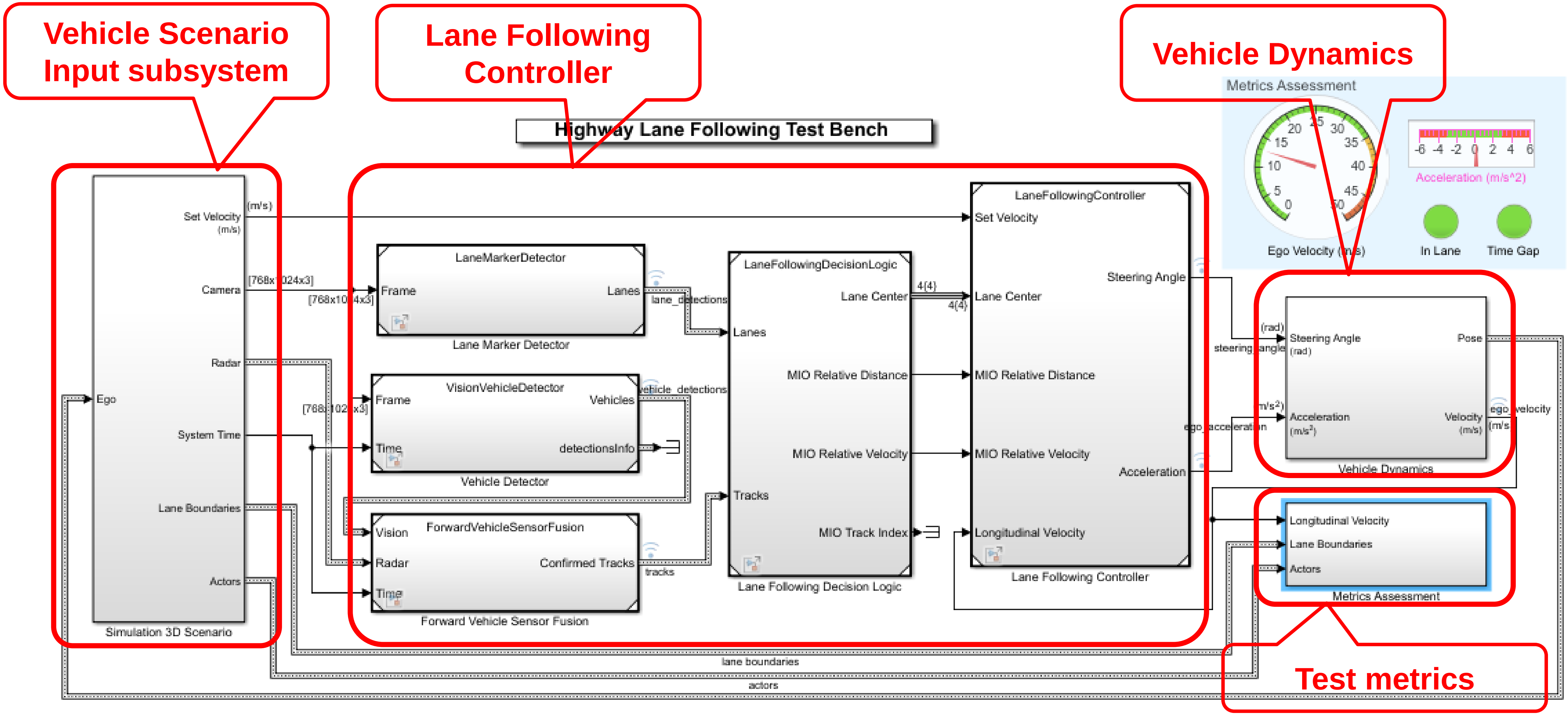




Out-of-the-box capability

Custom virtual vehicle solution

Demo: Regression Testing of Highway Lane Following Controller



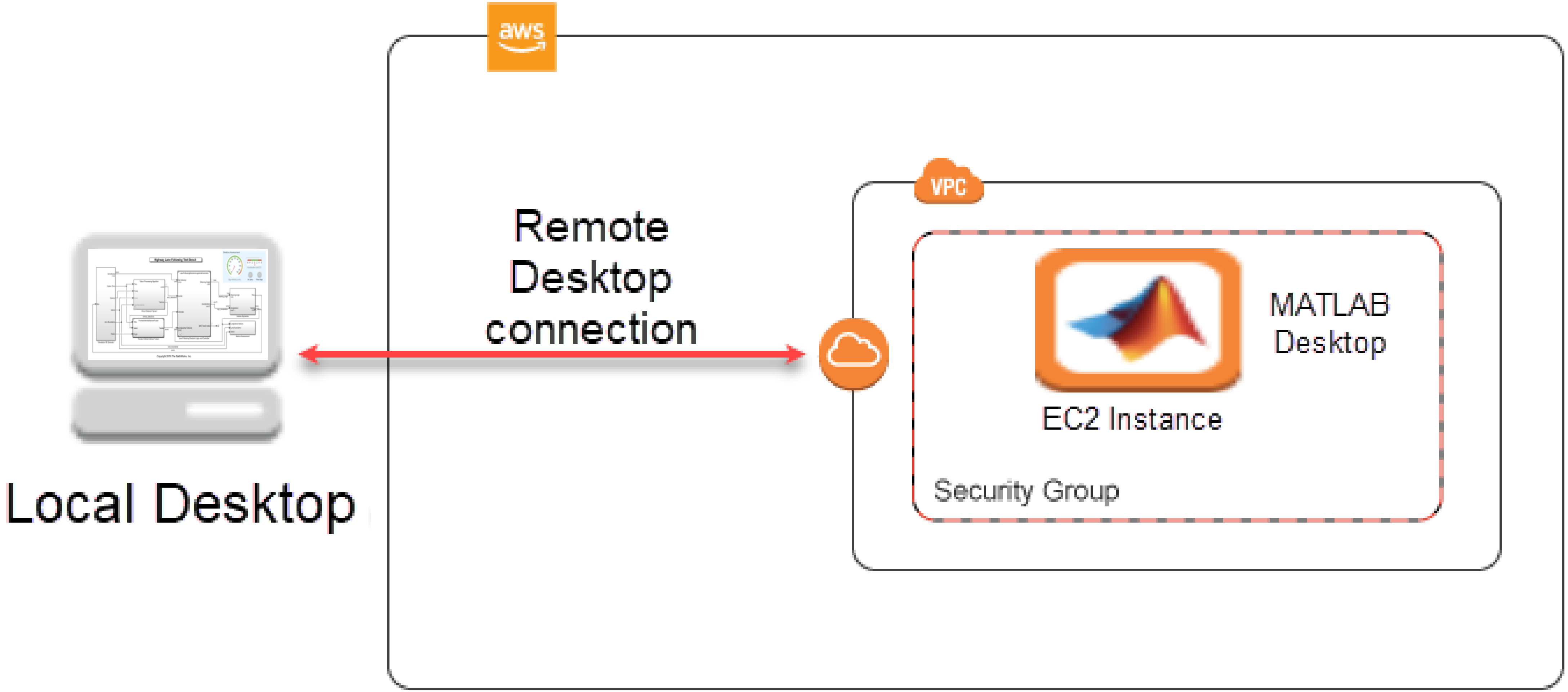
Vehicle Scenario Input subsystem

Lane Following Controller

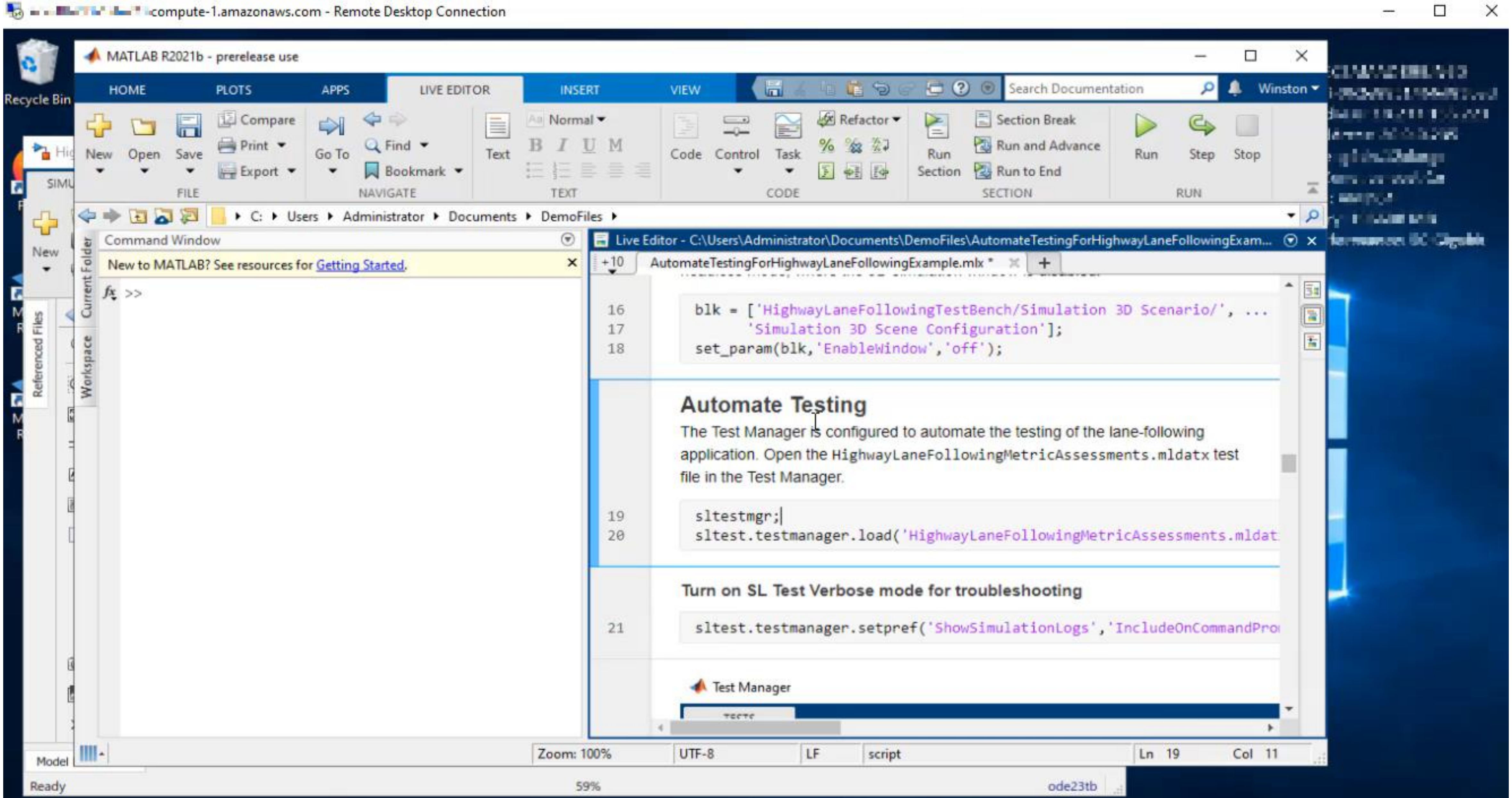
Vehicle Dynamics

Test metrics

Leveraging a Prebuilt Cloud Configuration via Reference Architecture



Running Regression Tests on Cloud using Parallel Computing Toolbox



```
blk = ['HighwayLaneFollowingTestBench/Simulation 3D Scenario/', ...
       'Simulation 3D Scene Configuration'];
set_param(blk,'EnableWindow','off');
```

Automate Testing

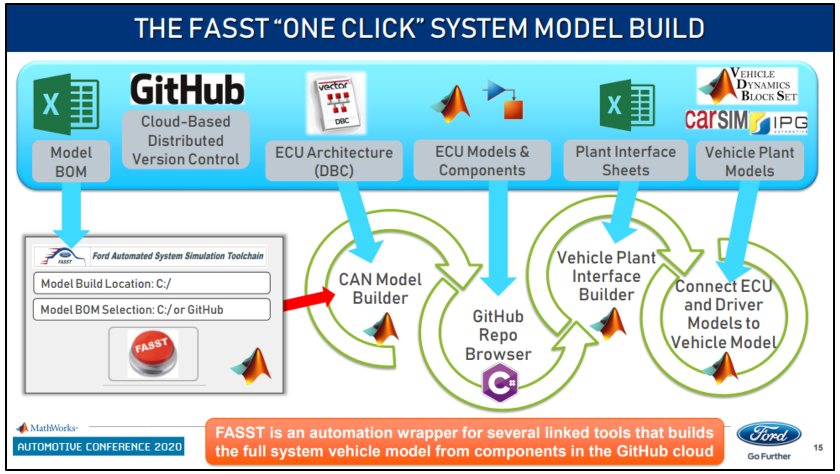
The Test Manager is configured to automate the testing of the lane-following application. Open the `HighwayLaneFollowingMetricAssessments.mldatx` test file in the Test Manager.

```
sltestmgr;
sltest.testmanager.load('HighwayLaneFollowingMetricAssessments.mldat
```

Turn on SL Test Verbose mode for troubleshooting

```
sltest.testmanager.setpref('ShowSimulationLogs','IncludeOnCommandPro
```

How Are Companies Building Virtual Vehicles with MathWorks?



THE FASST "ONE CLICK" SYSTEM MODEL BUILD

Model BOM
GitHub
Cloud-Based Distributed Version Control
ECU Architecture (DBC)
ECU Models & Components
Plant Interface Sheets
Vehicle Dynamics Block Set
CarSim
IPG
Vehicle Plant Models
Ford Automated System Simulation Toolchain
Model Build Location: C:/
Model BOM Selection: C:/ or GitHub
FASST
CAN Model Builder
GitHub Repo Browser
Vehicle Plant Interface Builder
Connect ECU and Driver Models to Vehicle Model

FASST is an automation wrapper for several linked tools that builds the full system vehicle model from components in the GitHub cloud

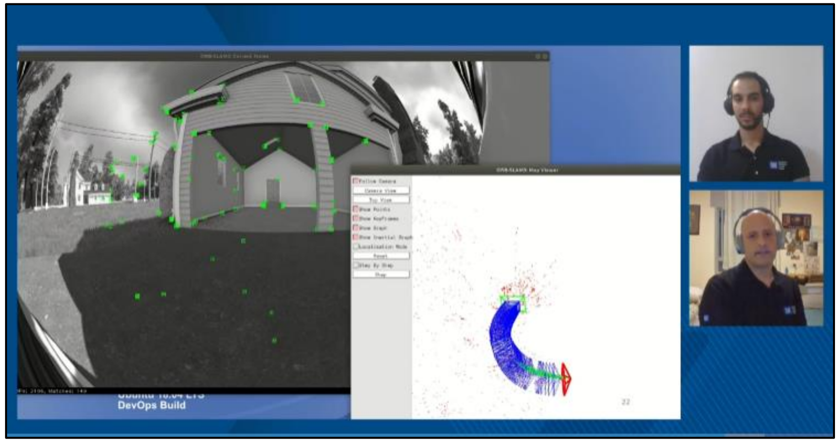
MathWorks AUTOMOTIVE CONFERENCE 2020

Ford Go Further

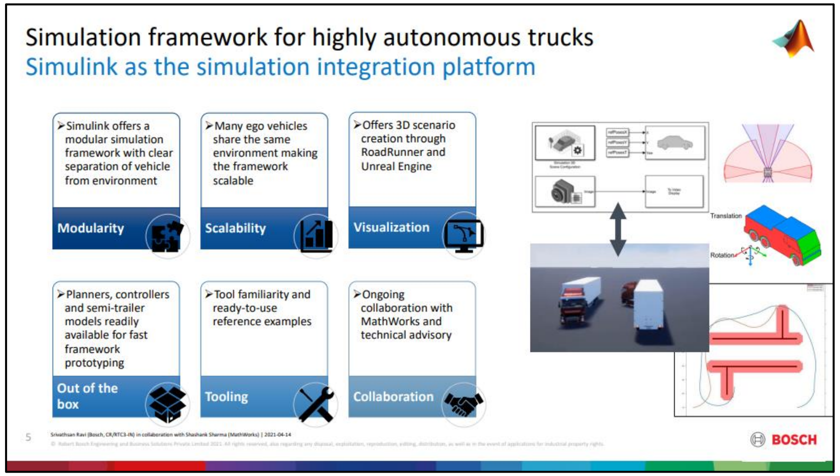
15

Ford: Build Virtual Vehicle in minutes

Different virtual vehicles are built for different use cases



GM: Autonomous parking development



Simulation framework for highly autonomous trucks

Simulink as the simulation integration platform

➢Simulink offers a modular simulation framework with clear separation of vehicle from environment **Modularity**	➢Many ego vehicles share the same environment making the framework scalable **Scalability**	➢Offers 3D scenario creation through RoadRunner and Unreal Engine **Visualization**
➢Planners, controllers and semi-trailer models readily available for fast framework prototyping **Out of the box**	➢Tool familiarity and ready-to-use reference examples **Tooling**	➢Ongoing collaboration with MathWorks and technical advisory **Collaboration**

5 Srivathsan Ravi (Bosch, CR/RTC3-IN) in collaboration with Shashank Sharma (MathWorks) | 2021-04-14

© Robert Bosch Engineering and Business Solutions Private Limited 2021. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

BOSCH

Bosch: Autonomous truck development

Common themes are the automation of model creation, simulation and analysis

Agenda

- Common challenges

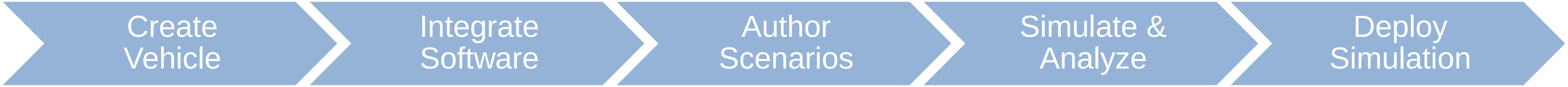
- MathWorks solutions

Agenda

- **Common challenges**

- MathWorks solutions

Challenges to early system-level testing

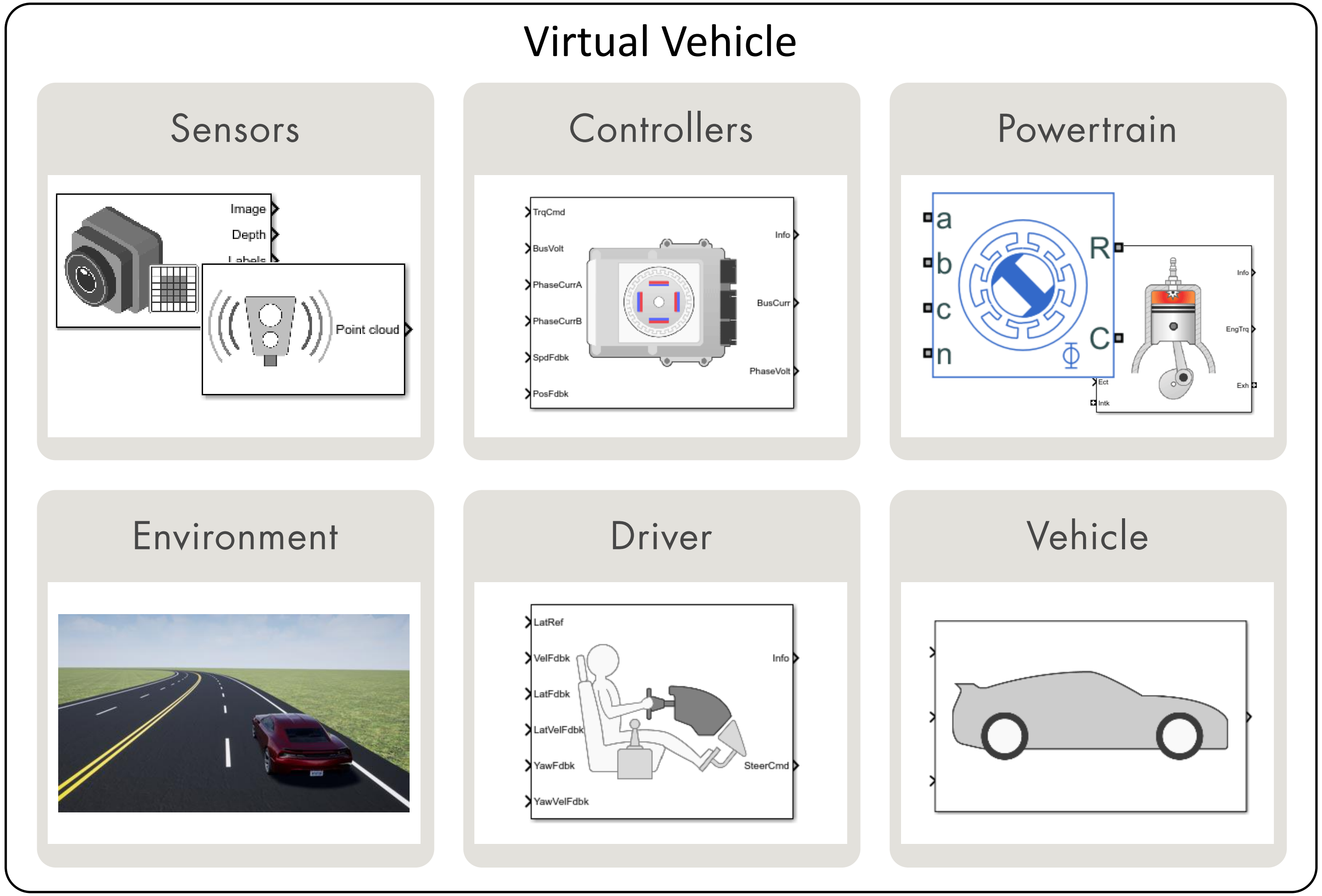

Create Vehicle

Integrate Software

Author Scenarios

Simulate & Analyze

Deploy Simulation


Virtual Vehicle

Sensors

Controllers

Powertrain

Environment

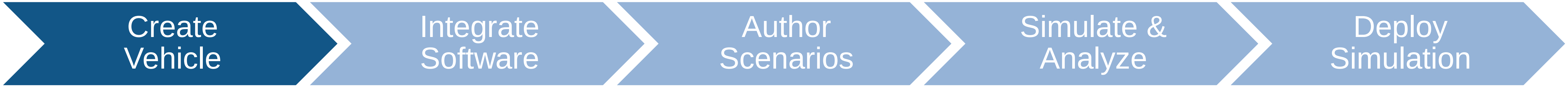
Driver

Vehicle

Using a virtual vehicle for systems integration testing early in development can **save time / money**

What are the **challenges** to building one?

Challenges to early system-level testing



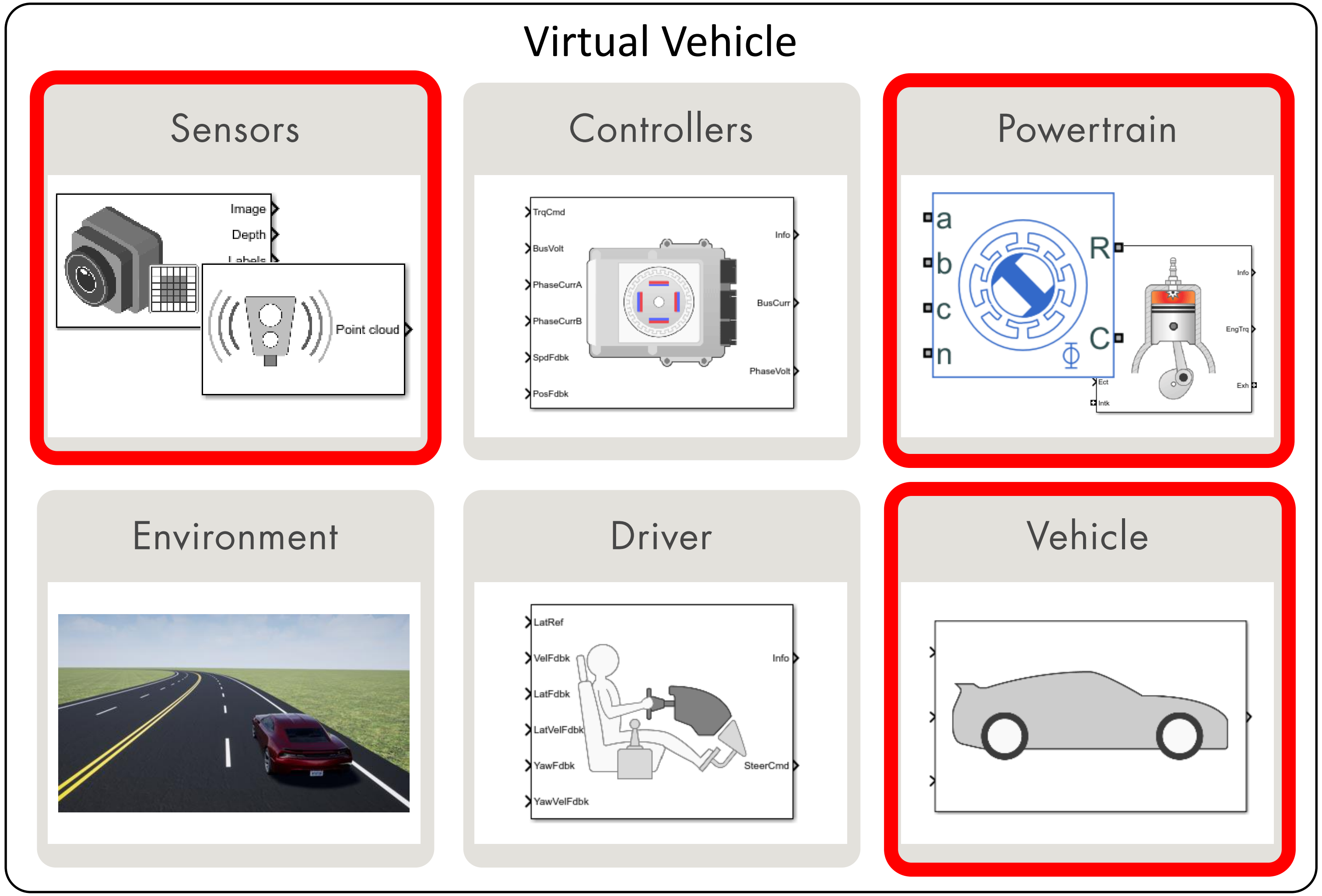
Create Vehicle

Integrate Software

Author Scenarios

Simulate & Analyze

Deploy Simulation



Virtual Vehicle

Sensors

Controllers

Powertrain

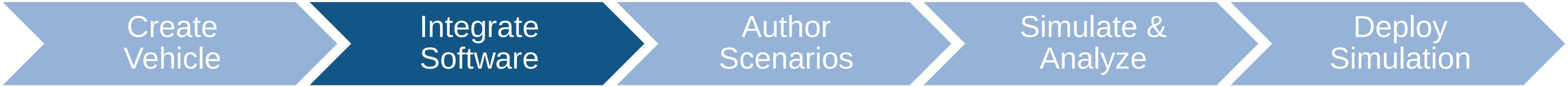
Environment

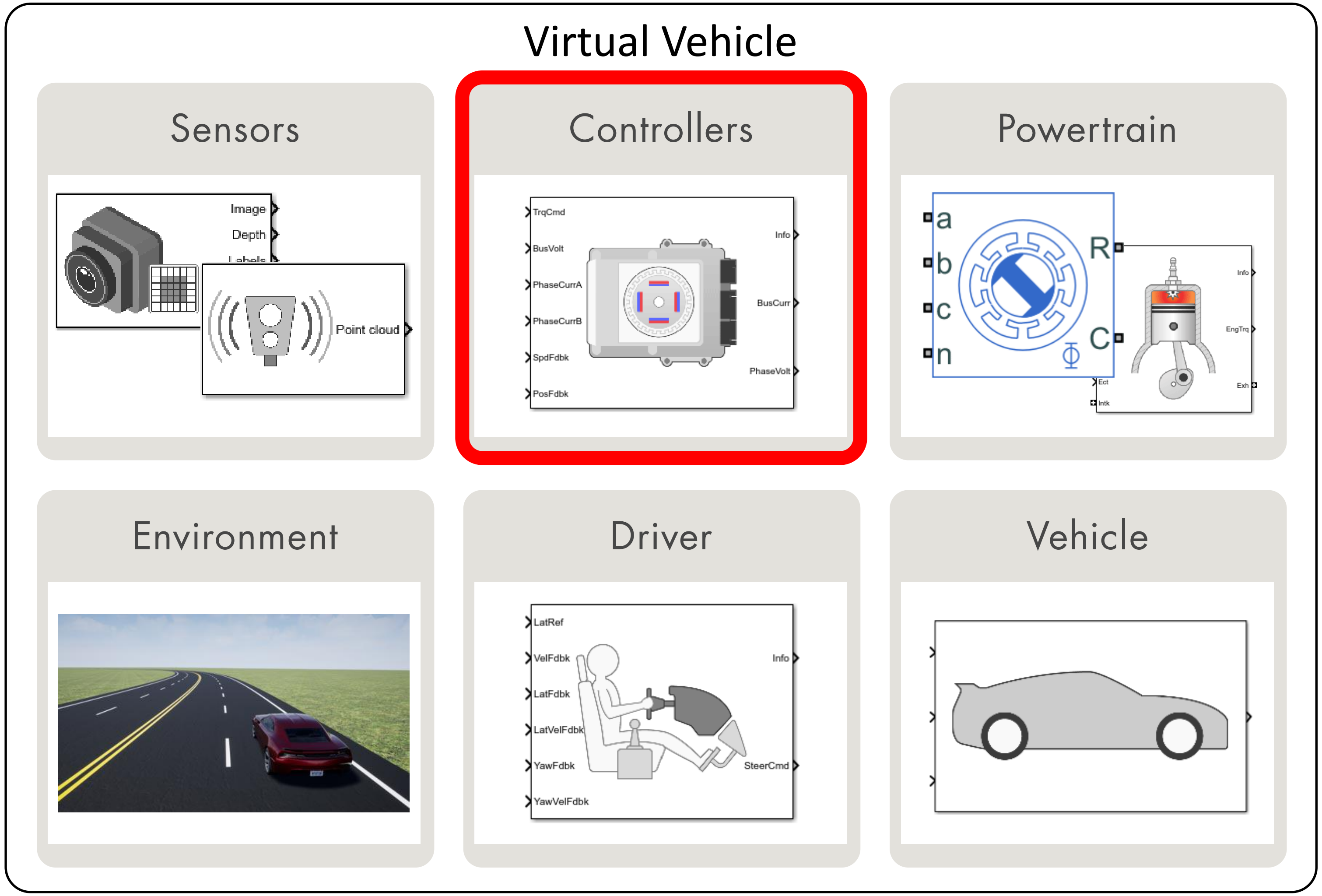
Driver

Vehicle

- Availability of appropriate vehicle level model
- Access to plant and sensor models with “right” level of fidelity
- Model calibration

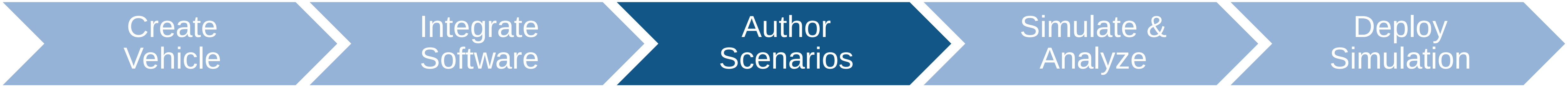
Challenges to early system-level testing





- Standardizing interfaces and data management
- Access to software components across different teams
- Assembly of software components from multiple sources

Challenges to early system-level testing



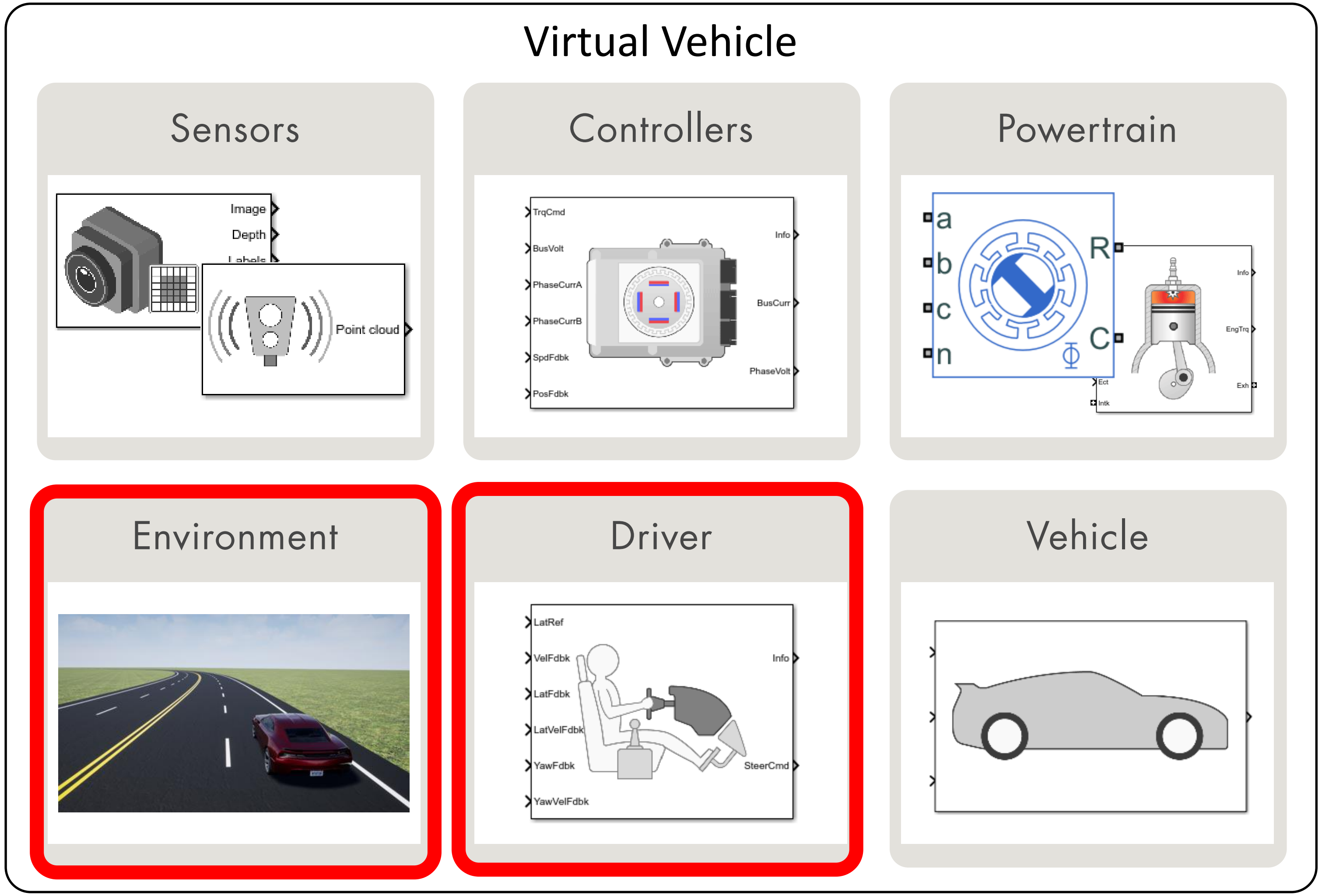
Create Vehicle

Integrate Software

Author Scenarios

Simulate & Analyze

Deploy Simulation



Virtual Vehicle

Sensors

Controllers

Powertrain

Environment

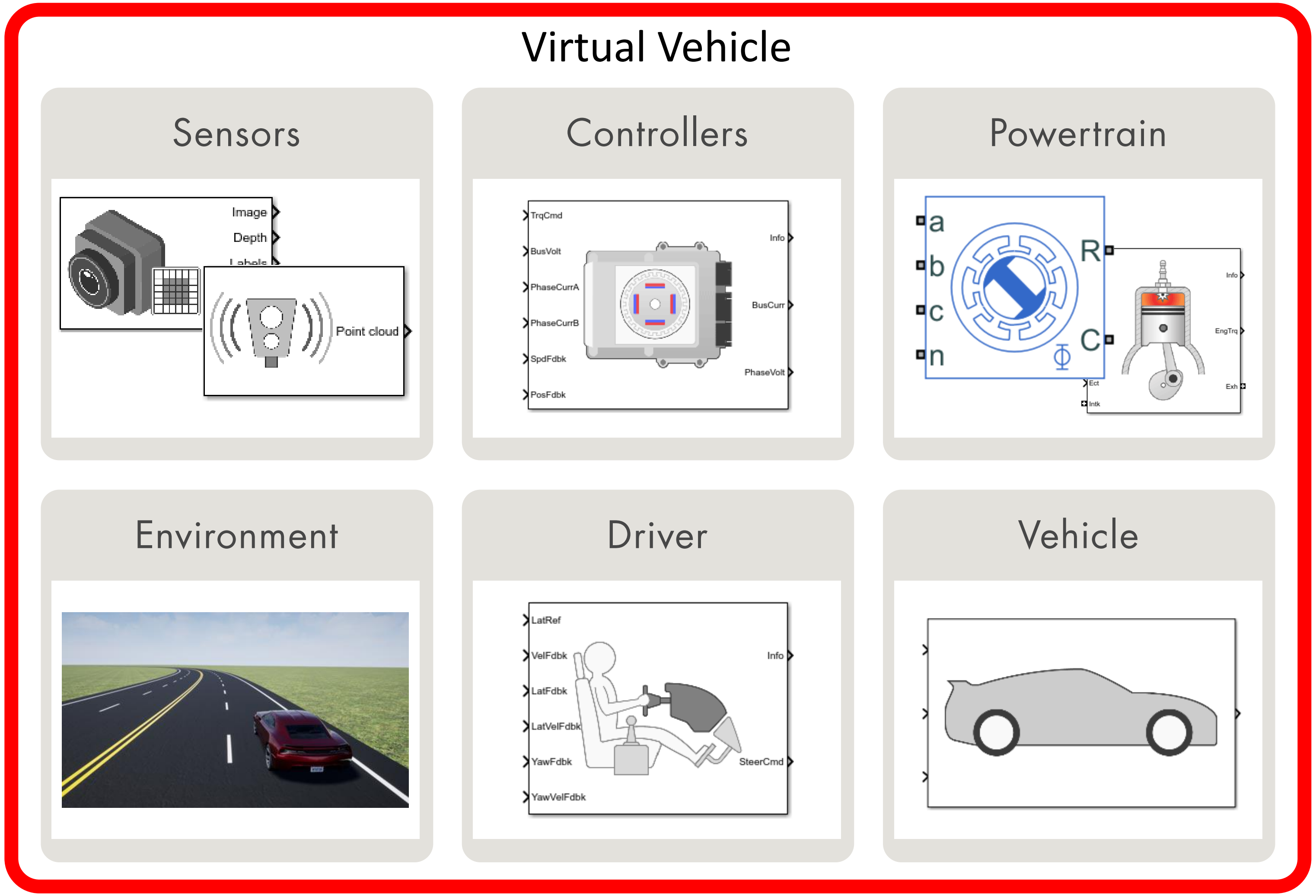
Driver

Vehicle

- Creation of virtual 3D environment
- Definition of scenarios to test
- Linking test cases to requirements

Challenges to early system-level testing

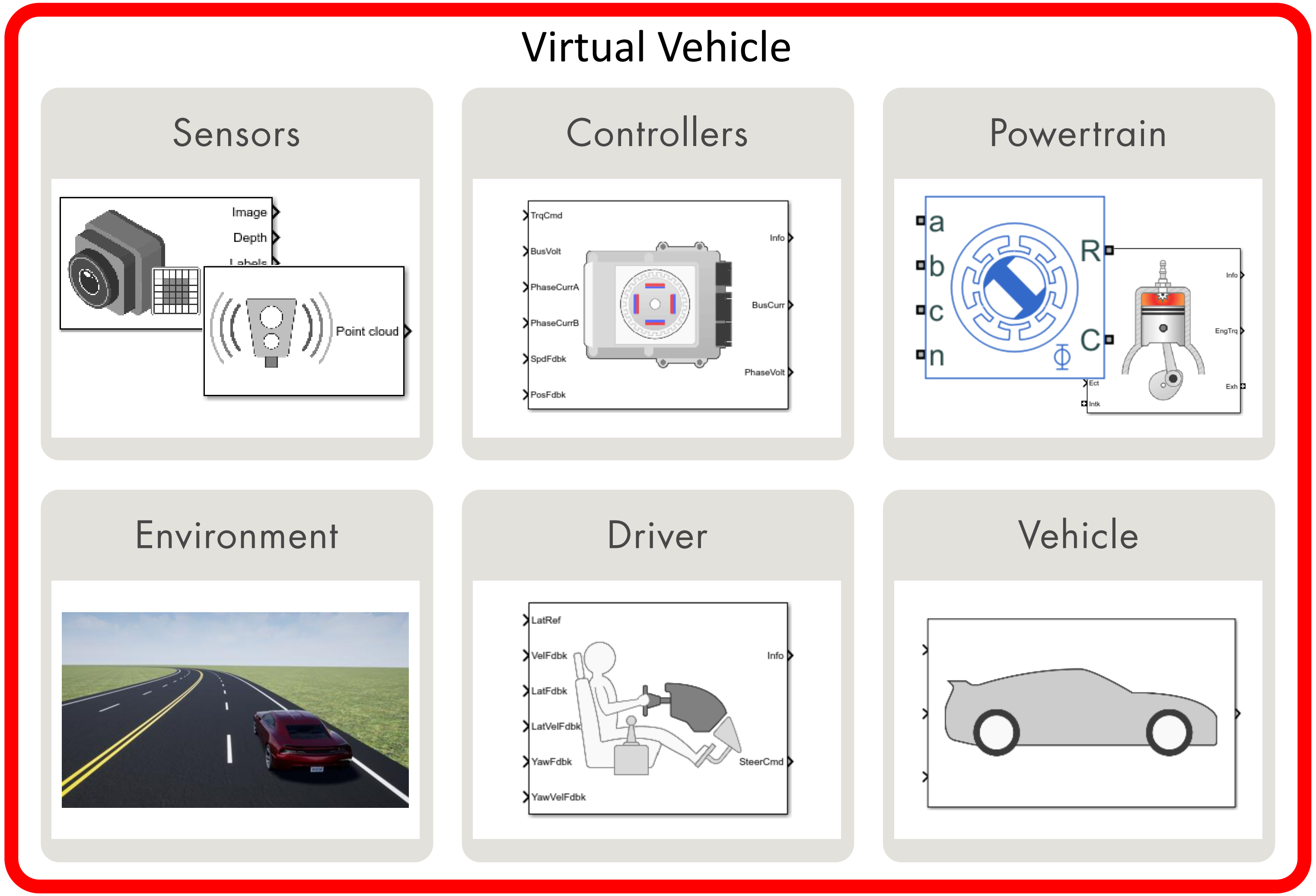
Create Vehicle → Integrate Software → Author Scenarios → **Simulate & Analyze** → Deploy Simulation



- Post-processing and visualizing results
- Automatically generating reports
- Running large numbers of simulations efficiently

Challenges to early system-level testing

Create Vehicle → Integrate Software → Author Scenarios → Simulate & Analyze → **Deploy Simulation**



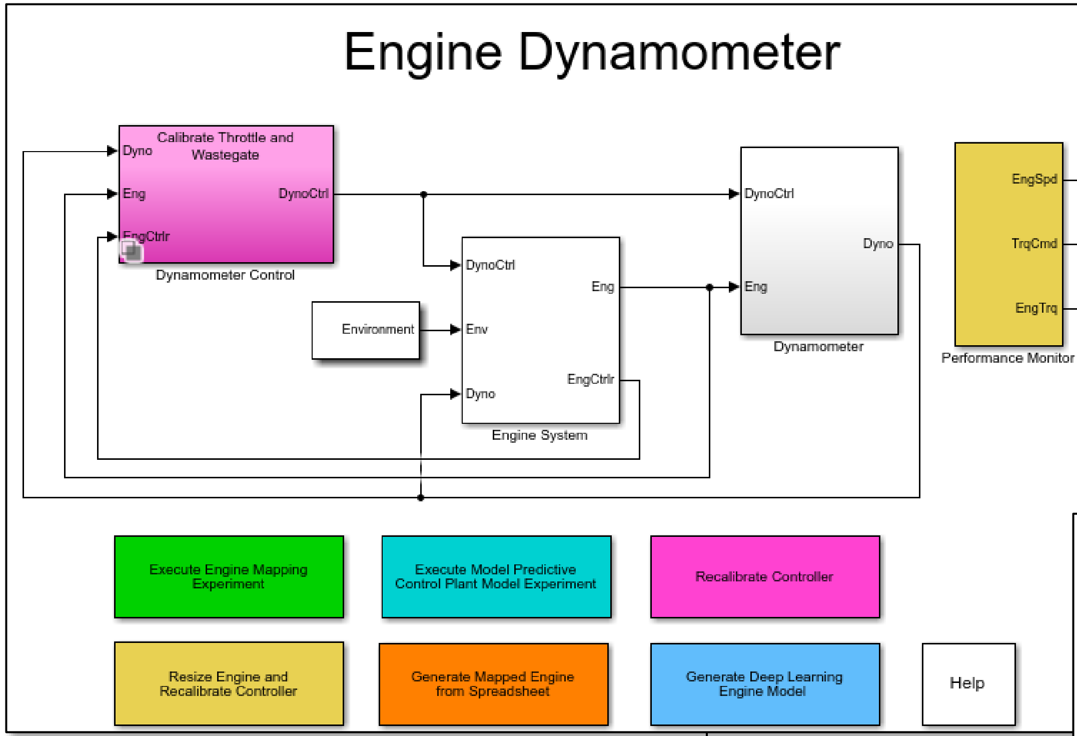
- Sharing models across the organization
- Deploying models to users who aren’t tool experts
- Deploying models for SIL, HIL, etc.

Agenda

- Common challenges
- **MathWorks solutions**

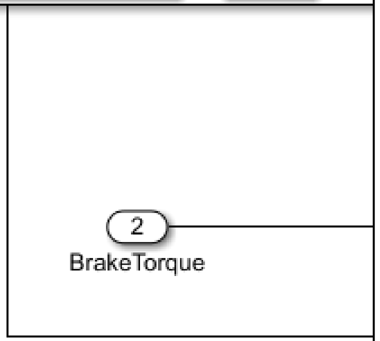
Create Vehicle: Reference Applications

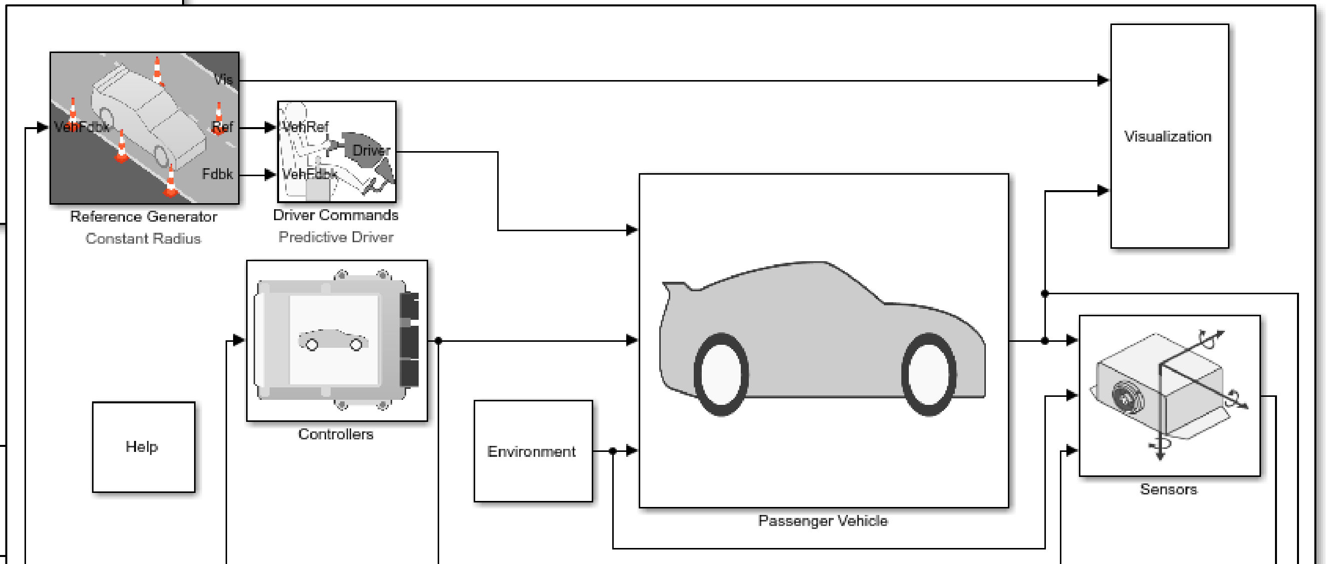
Create Vehicle → Integrate Software → Author Scenarios → Simulate & Analyze → Deploy Simulation



Engine Dynamometer
Calibrate Throttle and Wastegate
Dyno
Eng
DynoCtrl
EngCtrlr
Dynamometer Control
DynoCtrl
Eng
Environment
Env
Dyno
EngCtrlr
Engine System
DynoCtrl
Dyno
Eng
Dynamometer
EngSpd
TrqCmd
EngTrq
Performance Monitor
Execute Engine Mapping Experiment
Execute Model Predictive Control Plant Model Experiment
Recalibrate Controller
Resize Engine and Recalibrate Controller
Generate Mapped Engine from Spreadsheet
Generate Deep Learning Engine Model
Help

- Start with in-house vehicle models
  - We can help you customize it and apply best practices for Model-Based Design
- Start with our reference applications
  - Detailed system and vehicle level models for powertrain, vehicle dynamics, ADAS and other applications





Learn more:

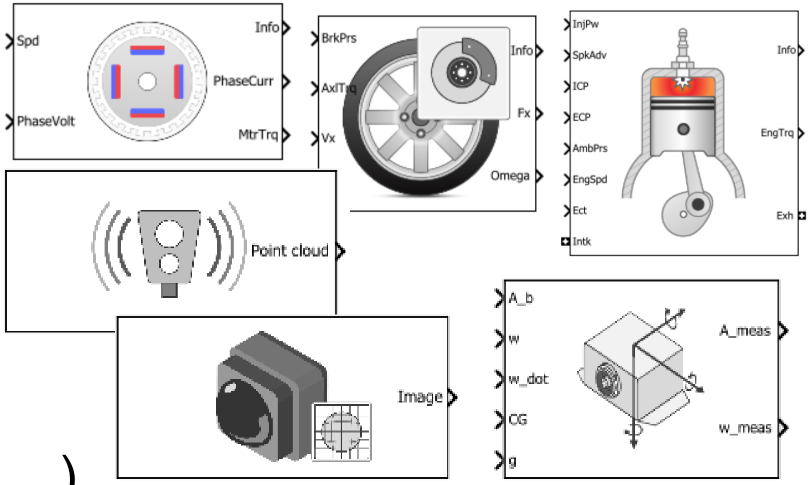
[Powertrain Blockset](#)

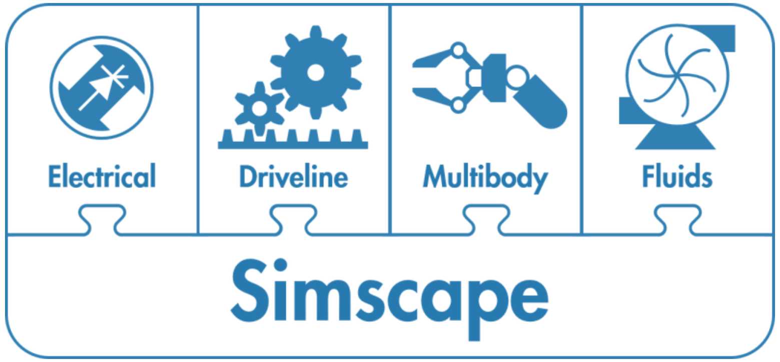
[Vehicle Dynamics Blockset](#)

[Automated Driving Toolbox](#)

Create Vehicle: Model Customization

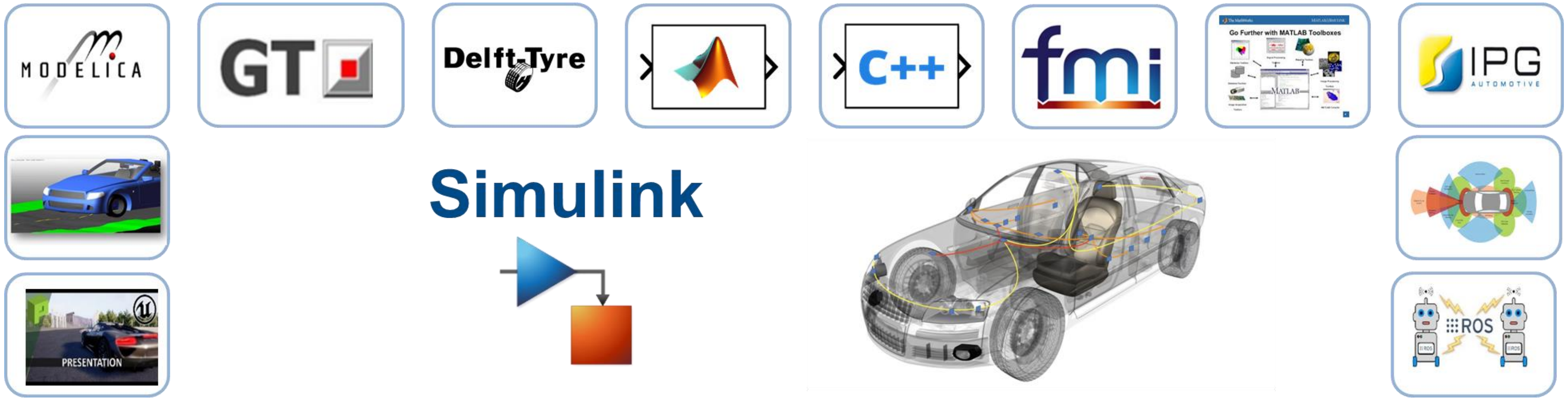
Create Vehicle → Integrate Software → Author Scenarios → Simulate & Analyze → Deploy Simulation





Add detail where needed using:

- In-house Simulink models
- Simulink and Simscape libraries
- 3rd party tools (S-function, FMU, …)



Learn more:

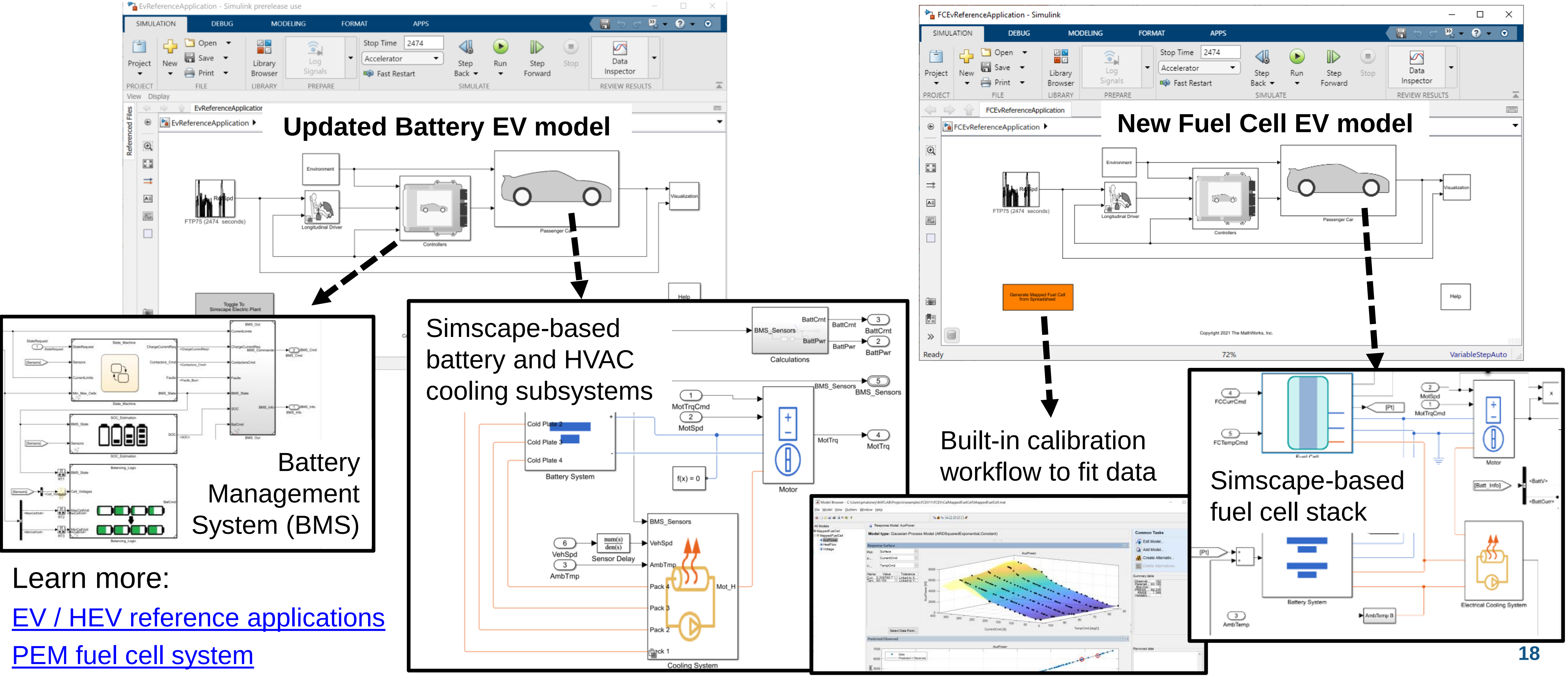
[Simscape](#)

[Multi-core cosim](#)

[Integrate with existing sims](#)

Create Vehicle: Electrified Powertrain Modeling

Create Vehicle → Integrate Software → Author Scenarios → Simulate & Analyze → Deploy Simulation



Updated Battery EV model

New Fuel Cell EV model

Battery Management System (BMS)

Simscape-based battery and HVAC cooling subsystems

Built-in calibration workflow to fit data

Simscape-based fuel cell stack

Learn more:

[EV / HEV reference applications]

[PEM fuel cell system]

Create Vehicle: Non-Passenger Vehicle Modeling

Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation

Motorcycle braking test

- New blocks
  - Motorcycle body
  - Chain drive
- New reference application
  - 5-state ABS algorithm

Tractor / trailer steering test

- New blocks
  - 3 DOF / 6 DOF bodies
  - 1, 2 or 3 axle trailers
- New reference application
  - Test system response, swept path, etc.

Learn more:

[Motorcycle braking test]

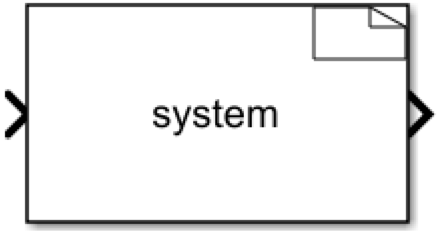
[One-axle trailer example]

Integrate Software: C Code Integration

Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation

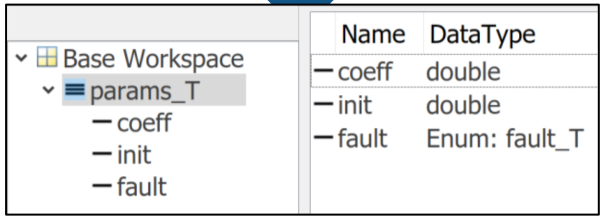
Integrate controller algorithms:

- Native Simulink models
- 3rd party tools (S-function, FMU, …)
- C / C++ code

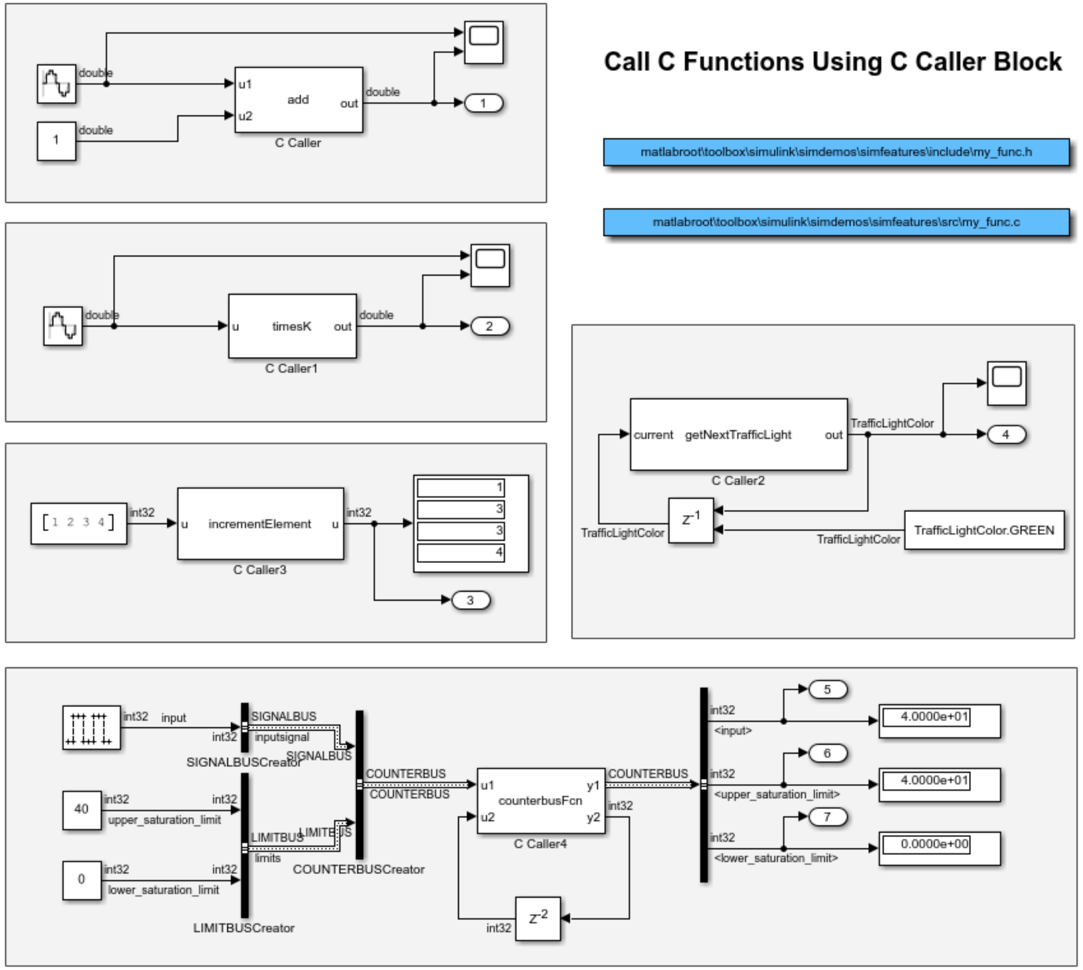


S-Function Builder

```
typedef struct {
    double   coeff;
    double   init;
    fault_T  fault;
} params_T;
```



Name	DataType
coeff	double
init	double
fault	Enum: fault_T



Call C Functions Using C Caller Block

matlabroot\toolbox\simulink\simdemos\simfeatures\include\my_func.h

matlabroot\toolbox\simulink\simdemos\simfeatures\src\my_func.c

Learn more:

C / C++ code integration

C Caller block

Integrate Software: Integrating Lane Detection C Code

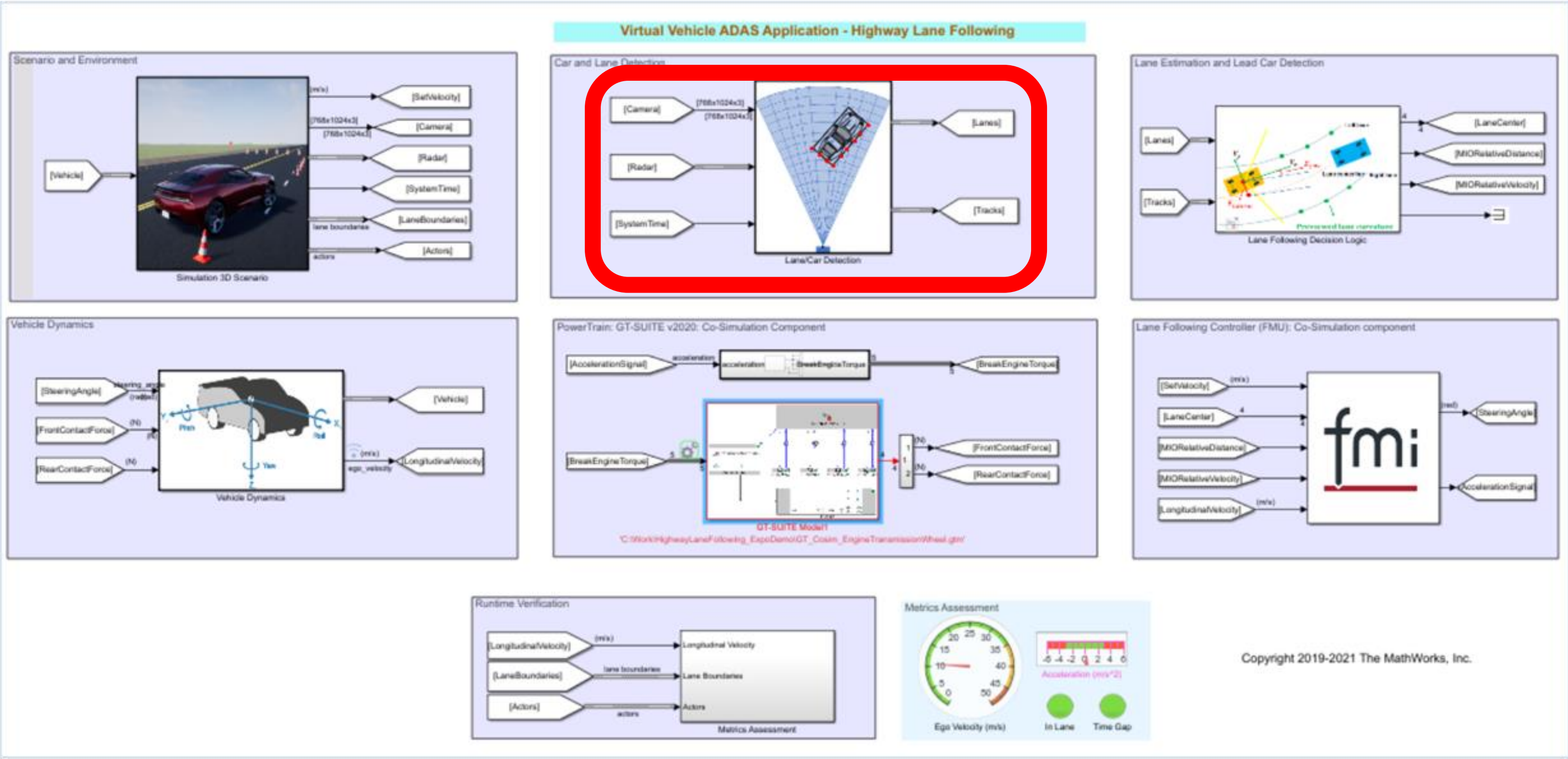
Create Vehicle

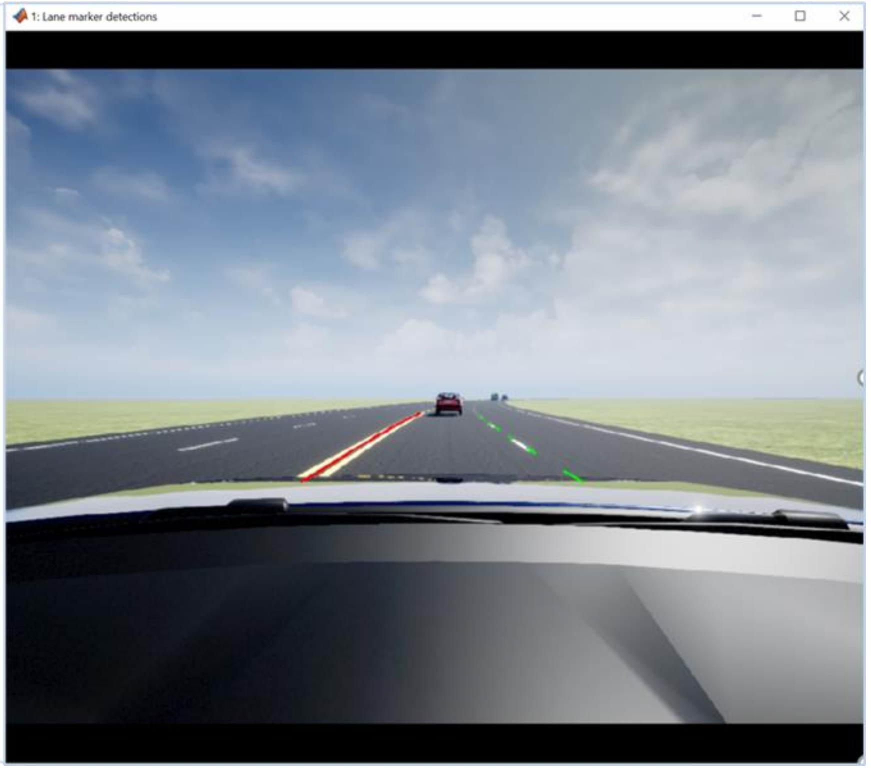
Integrate Software

Author Scenarios

Simulate & Analyze

Deploy Simulation





Integrating custom C code for lane marker detection

Learn more:

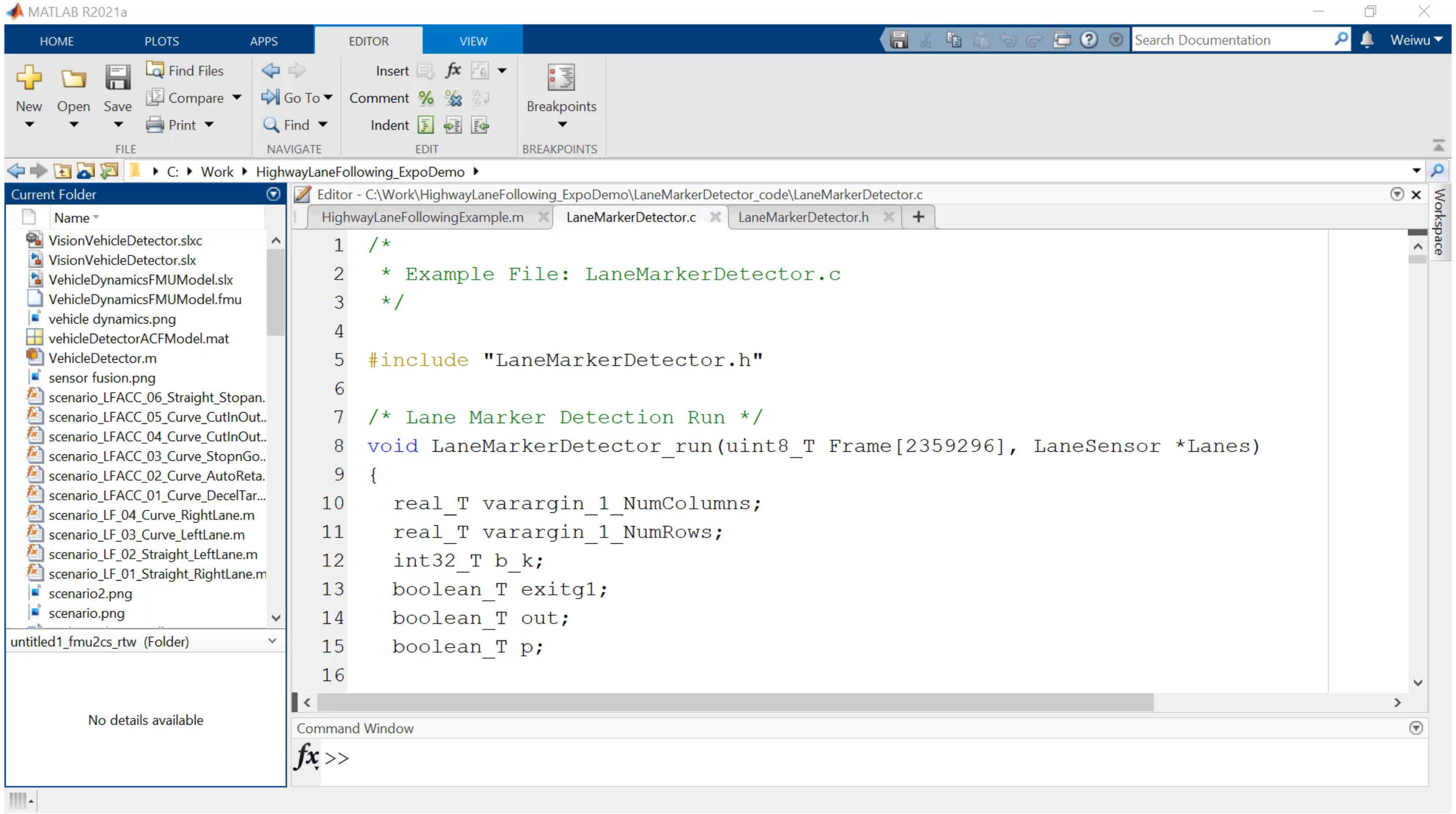
C Function Block

Integrate Software: Integrating Lane Detection C Code

Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation

Use C Function block to integrate code:

- Specify header + source files and where to find them
- Configure block to call external functions at:
  - Initialization
  - Each time step
  - Termination



VisionVehicleDetector.slxc
VisionVehicleDetector.slx
VehicleDynamicsFMUModel.slx
VehicleDynamicsFMUModel.fmu
vehicle dynamics.png
vehicleDetectorACFModel.mat
VehicleDetector.m
sensor fusion.png
scenario_LFACC_06_Straight_Stopan.
scenario_LFACC_05_Curve_CutInOut..
scenario_LFACC_04_Curve_CutInOut..
scenario_LFACC_03_Curve_StopnGo..
scenario_LFACC_02_Curve_AutoReta.
scenario_LFACC_01_Curve_DecelTar...
scenario_LF_04_Curve_RightLane.m
scenario_LF_03_Curve_LeftLane.m
scenario_LF_02_Straight_LeftLane.m
scenario_LF_01_Straight_RightLane.m
scenario2.png
scenario.png

untitled1_fmu2cs_rtw (Folder)

No details available

```c
/*
 * Example File: LaneMarkerDetector.c
 */

#include "LaneMarkerDetector.h"

/* Lane Marker Detection Run */
void LaneMarkerDetector_run(uint8_T Frame[2359296], LaneSensor *Lanes)
{
  real_T varargin_1_NumColumns;
  real_T varargin_1_NumRows;
  int32_T b_k;
  boolean_T exitg1;
  boolean_T out;
  boolean_T p;

```

```
fx >>
```

Learn more:

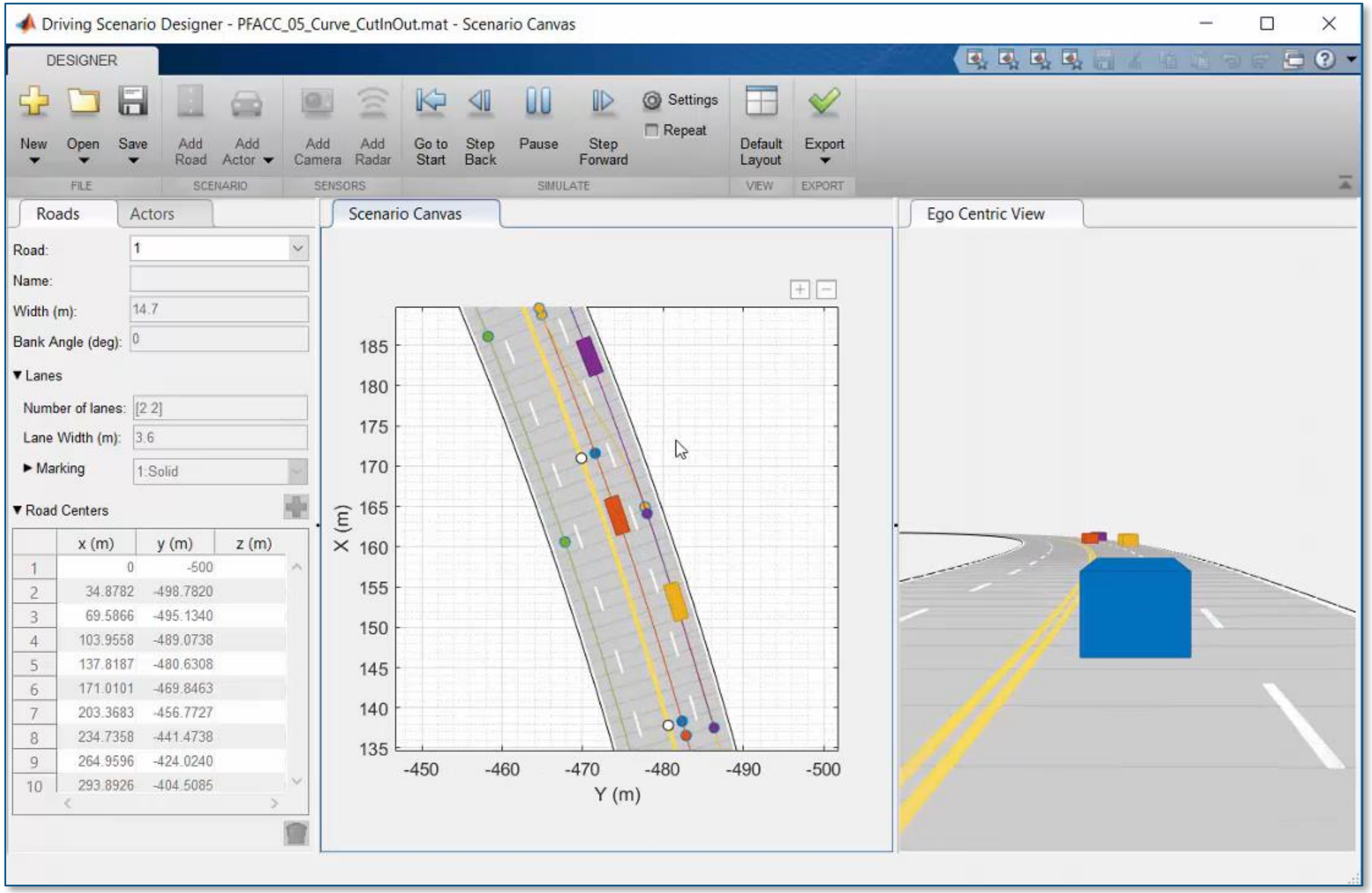
C Function Block

Author Scenarios: Graphical Scenario Authoring

Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation

Use Driving Scenario Designer to:

- Create roads and lane markings
- Add actors and trajectories
- Specify actor size and radar cross-section (RCS)
- Explore pre-built scenarios
- Import OpenDRIVE and HERE HD Live Map roads
- Export MATLAB code
- Export Simulink model



Road: 1

Name:

Width (m): 14.7

Bank Angle (deg): 0

▼ Lanes

Number of lanes: [2 2]

Lane Width (m): 3.6

► Marking 1:Solid

▼ Road Centers

	x (m)	y (m)	z (m)
1	0	-500	
2	34.8782	-498.7820	
3	69.5866	-495.1340	
4	103.9558	-489.0738	
5	137.8187	-480.6308	
6	171.0101	-469.8463	
7	203.3683	-456.7727	
8	234.7358	-441.4738	
9	264.9596	-424.0240	
10	293.8926	-404.5085	

Learn more:

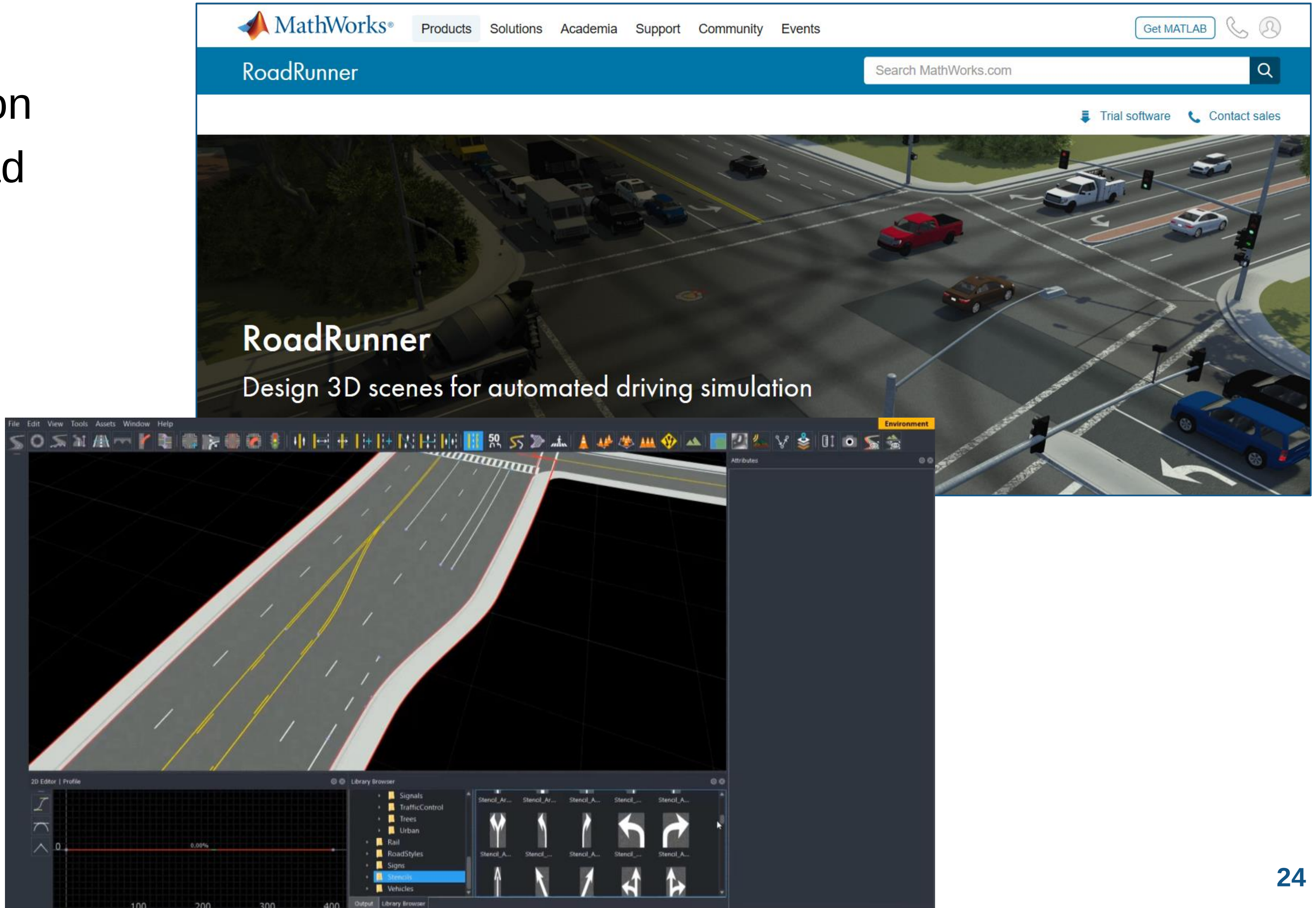
Automated Driving Toolbox

Author Scenarios: Automotive Scene Creation

Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation

Use RoadRunner to:

- Design 3D scenes for AD simulation
- Customize with region-specific road signs and markings
- Configure traffic signal timing
- Import from OpenDRIVE
- Export to OpenDRIVE, FBX, …
- Use scenes in Unreal, Unity, CARLA, …



RoadRunner

Design 3D scenes for automated driving simulation

Learn more:

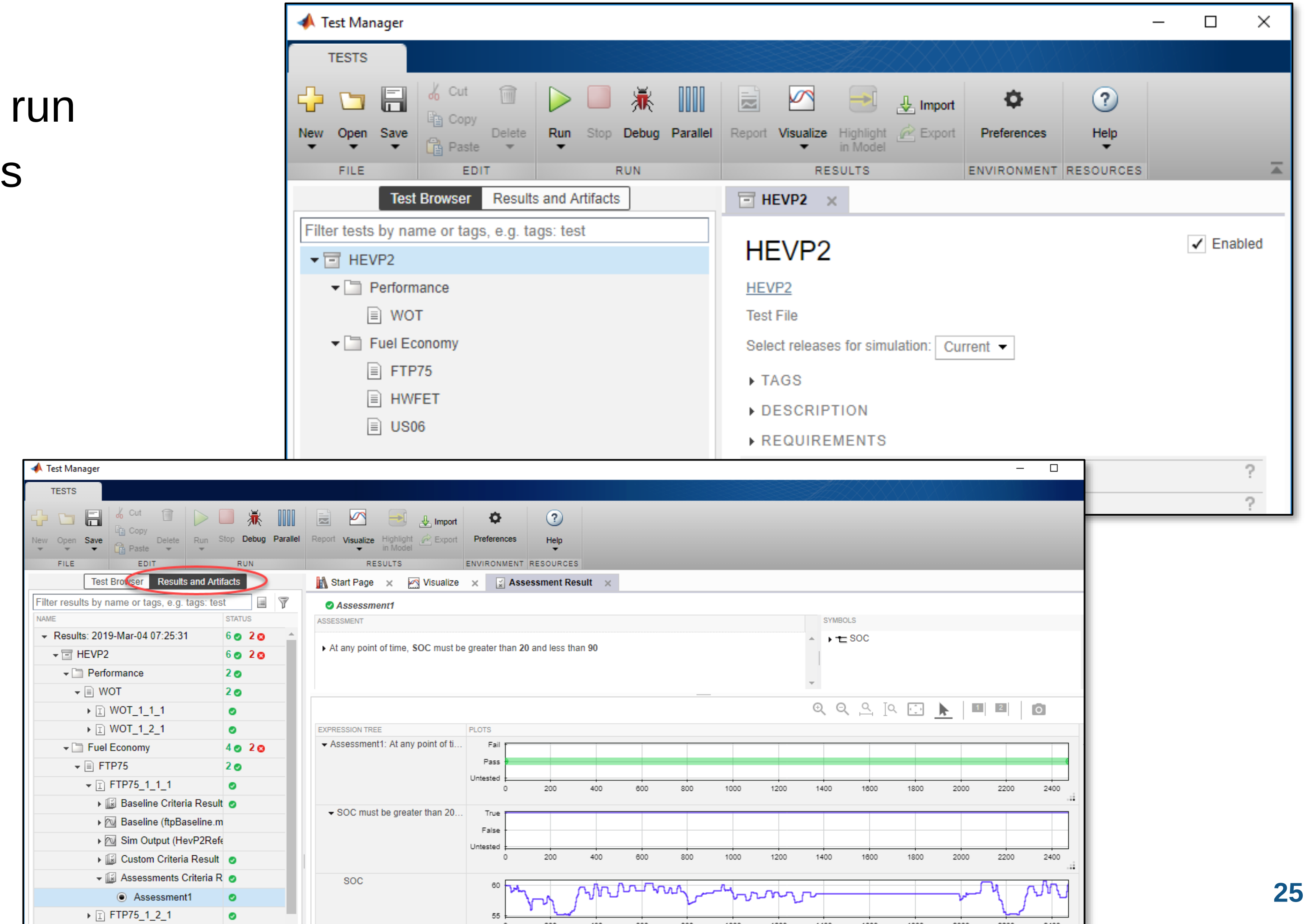
RoadRunner

Author Scenarios: Requirements Definition

Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation

Use V&V tools to:

- Define sequence of simulations to run
- Define requirements for these tests
- Define custom report template



Filter tests by name or tags, e.g. tags: test

- HEVP2
  - Performance
    - WOT
  - Fuel Economy
    - FTP75
    - HWFET
    - US06

HEVP2

☑ Enabled

HEVP2

Test File

Select releases for simulation: Current

Filter results by name or tags, e.g. tags: test

NAME	STATUS
Results: 2019-Mar-04 07:25:31	6 ✓ 2 ✗
HEVP2	6 ✓ 2 ✗
Performance	2 ✓
WOT	2 ✓
WOT_1_1_1	✓
WOT_1_2_1	✓
Fuel Economy	4 ✓ 2 ✗
FTP75	2 ✓
FTP75_1_1_1	✓
Baseline Criteria Result	✓
Baseline (ftpBaseline.m	
Sim Output (HevP2Refe	
Custom Criteria Result	✓
Assessments Criteria R	✓
Assessment1	✓
FTP75_1_2_1	✓

✓ *Assessment1*

At any point of time, **SOC** must be greater than **20** and less than **90**

SOC

Assessment1: At any point of ti...

SOC must be greater than 20...

SOC

HEVP2

Test Result Information

Result Type:	Test File Result
Parent:	None
Start Time:	04-Mar-2019 07:25:34
End Time:	04-Mar-2019 07:43:23
Outcome:	Total: 8, Passed: 6, Failed: 2

Test Suite Information

Name:	HEVP2

Back to Report Summary

Performance

Test Result Information

Result Type:	Test Suite Result
Parent:	HEVP2
Start Time:	04-Mar-2019 07:25:34
End Time:	04-Mar-2019 07:33:16
Outcome:	Total: 2, Passed: 2

Test Suite Information

Name:	Performance

Back to Report Summary

Learn more:

Verification & Validation

Author Scenarios: Custom Scene Creation

Create Vehicle

Integrate Software

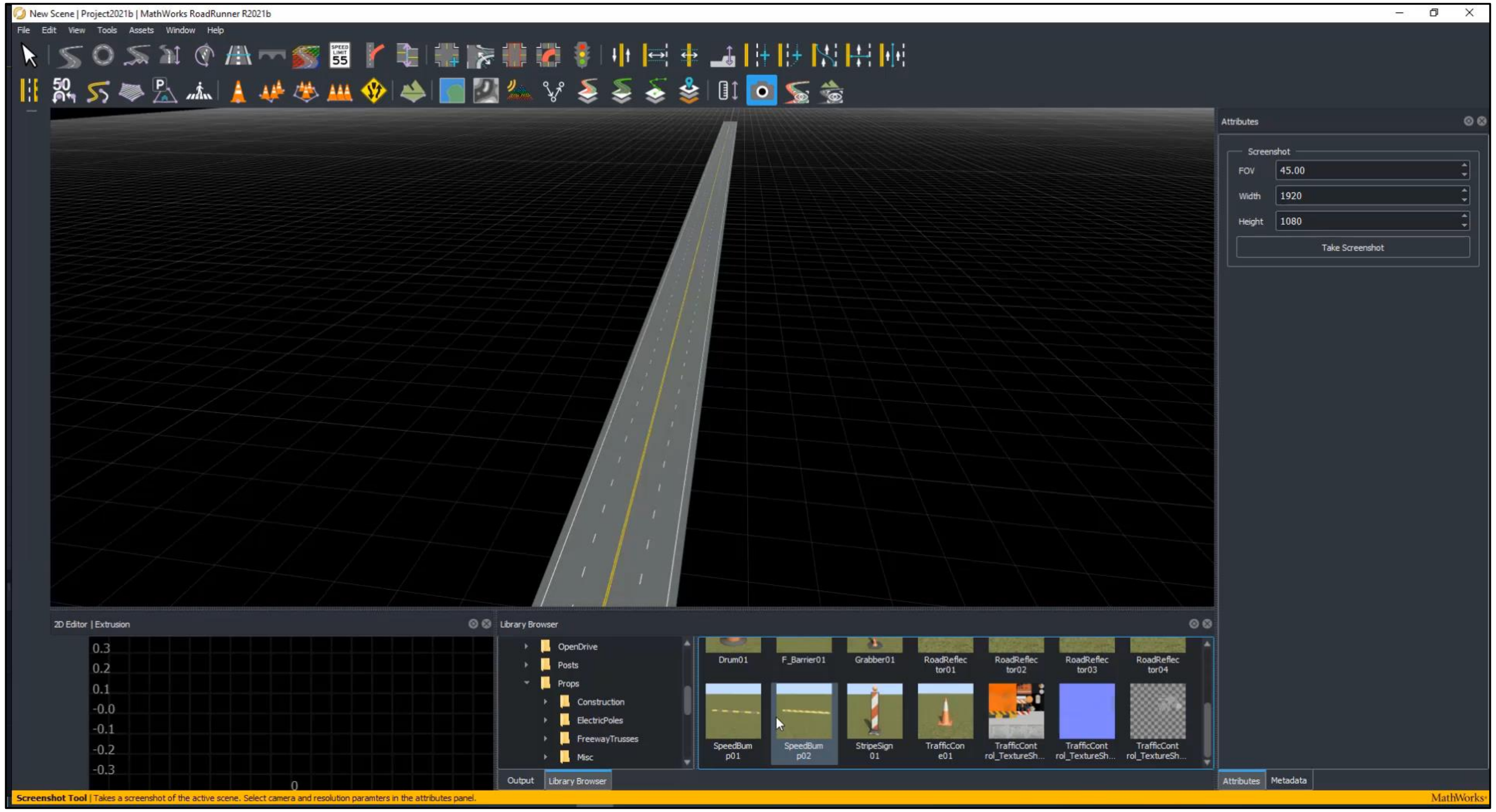
Author Scenarios

Simulate & Analyze

Deploy Simulation

Support Vehicle Dynamics and ADAS / AD testing:

- Create custom scenes in RoadRunner
- Import into Unreal for use with Simulink vehicle models
  - Multicast ray trace sensor to find contact patch
  - Speed bumps for suspension travel analysis
  - Hill for gradeability testing



New Scene | Project2021b | MathWorks RoadRunner R2021b

Attributes

Screenshot

FOV 45.00

Width 1920

Height 1080

Take Screenshot

Screenshot Tool | Takes a screenshot of the active scene. Select camera and resolution paramters in the attributes panel.

Learn more:

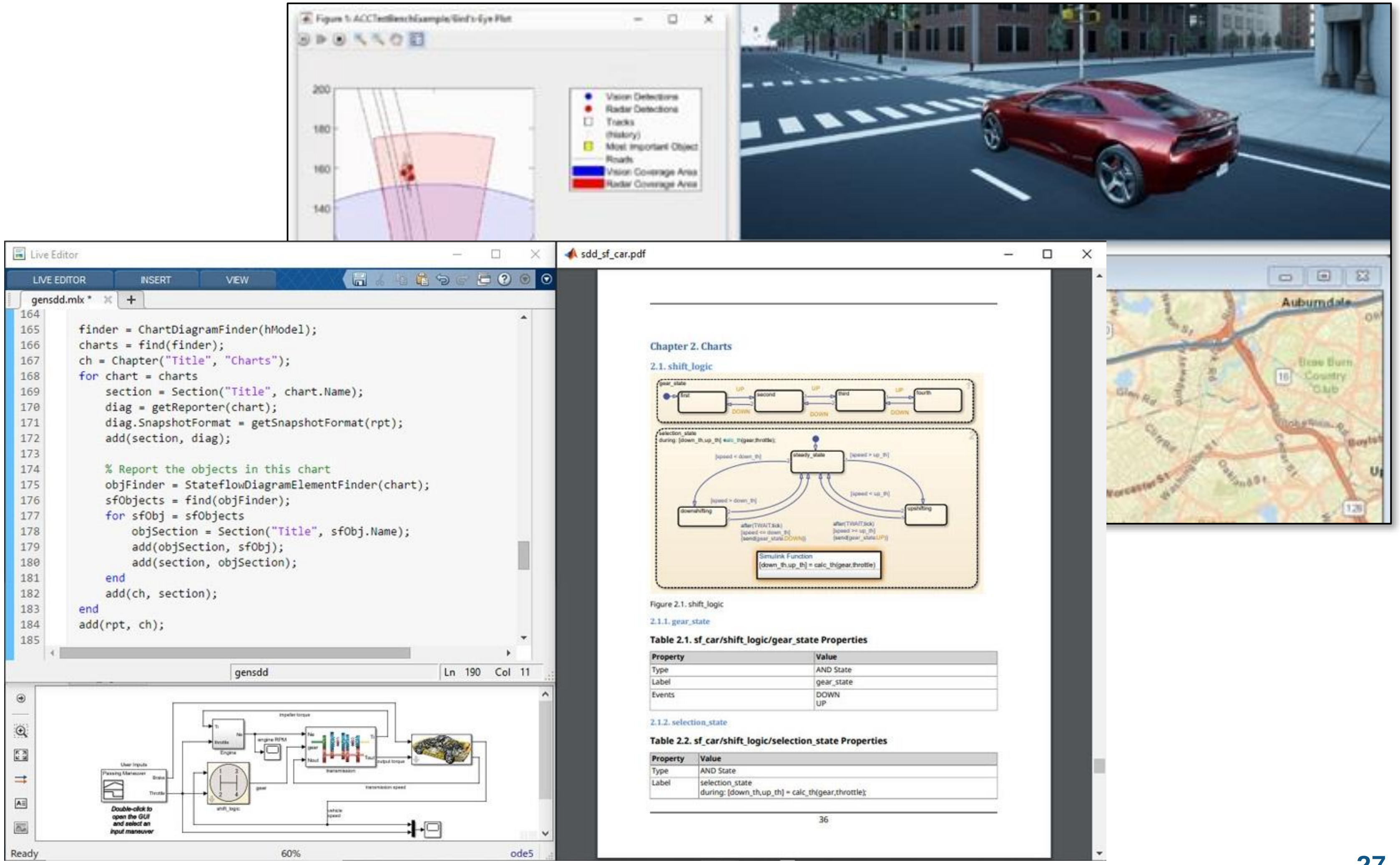
Using Unreal Engine with Simulink

Simulate & Analyze: Results Analysis

Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation

Use post-processing tools to:

- Review results with flexible MATLAB platform and visualization tools
- Interact with user-friendly Live Scripts
- Automate report generation



```
    finder = ChartDiagramFinder(hModel);
    charts = find(finder);
    ch = Chapter("Title", "Charts");
    for chart = charts
        section = Section("Title", chart.Name);
        diag = getReporter(chart);
        diag.SnapshotFormat = getSnapshotFormat(rpt);
        add(section, diag);

        % Report the objects in this chart
        objFinder = StateflowDiagramElementFinder(chart);
        sfObjects = find(objFinder);
        for sfObj = sfObjects
            objSection = Section("Title", sfObj.Name);
            add(objSection, sfObj);
            add(section, objSection);
        end
        add(ch, section);
    end
    add(rpt, ch);
```

Chapter 2. Charts

2.1. shift_logic

Figure 2.1. shift_logic

2.1.1. gear_state

Table 2.1. sf_car/shift_logic/gear_state Properties

Property	Value
Type	AND State
Label	gear_state
Events	DOWN UP

2.1.2. selection_state

Table 2.2. sf_car/shift_logic/selection_state Properties

Property	Value
Type	AND State
Label	selection_state during: [down_th,up_th] = calc_th(gear,throttle);

36

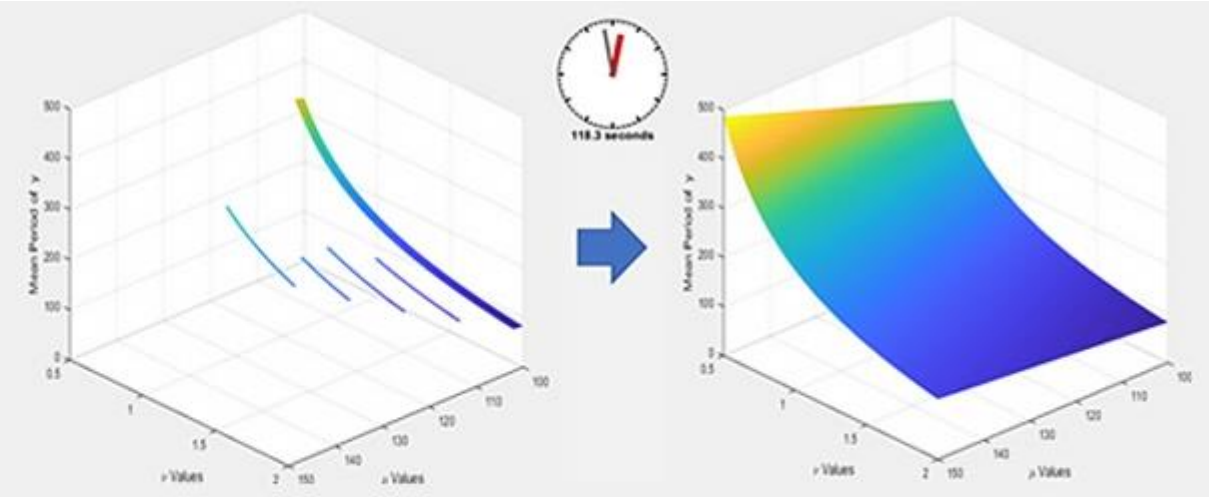
Learn more:

MATLAB Live Editor

Simulink Report Generator

Simulate & Analyze: Scalability

Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation



Use MATLAB and Simulink to:

- Distribute simulations to local multi-core, GPU, clusters, or the cloud
- Scale up computation power as needed without needing to rewrite code

Perform large-scale computations using multicore desktops, GPUs, clusters, grids, and clouds

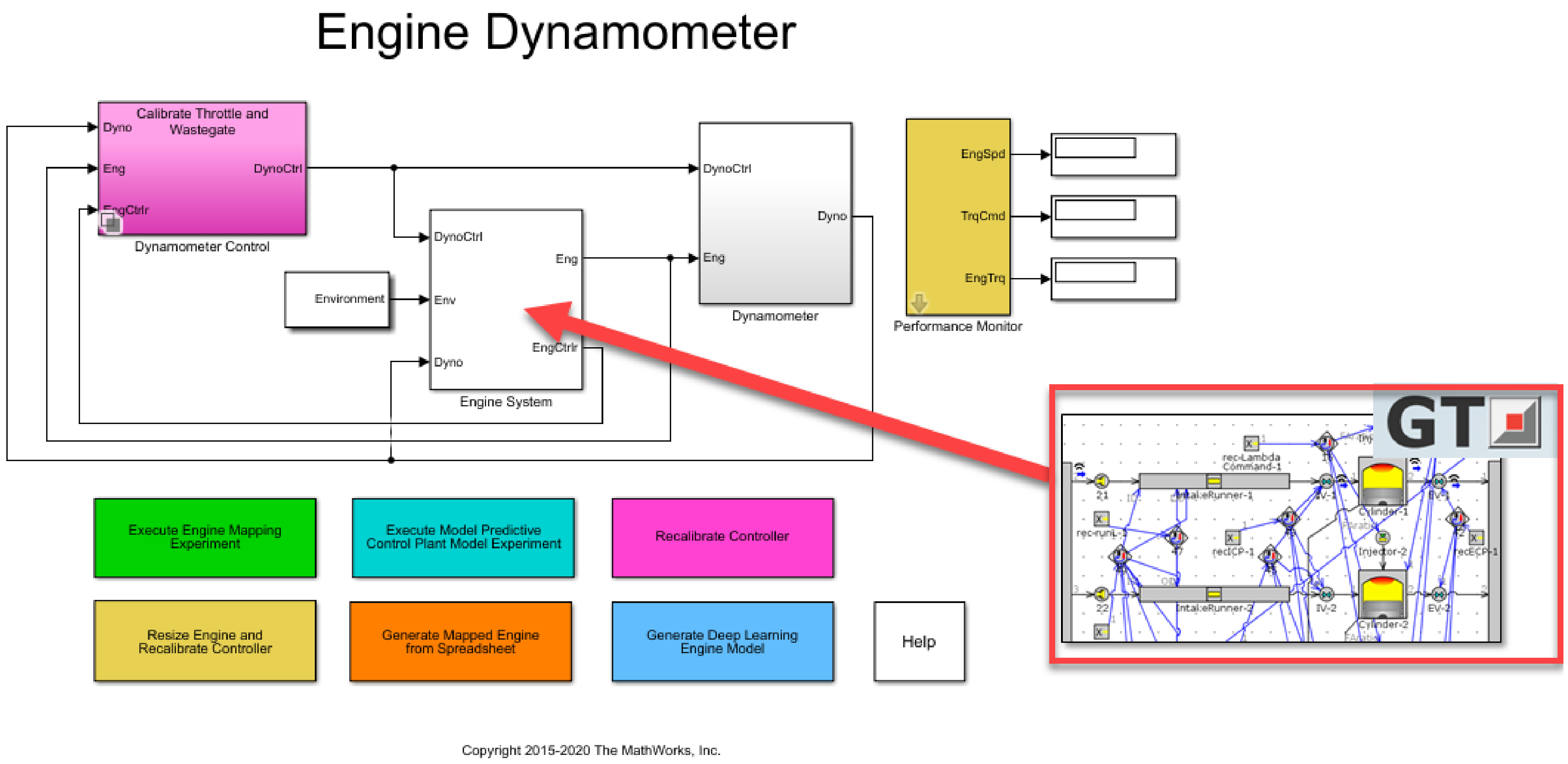
Learn more:

Parallel Computing Toolbox

MATLAB Parallel Server

Simulate & Analyze: Cloud-based ROM Generation

- Generate engine Reduced Order Model (ROM) for system-level analysis
- Automate with Simulink + GT Power Co-simulation
- Leverage parallel computing on Amazon Web Services (AWS) to speed up process



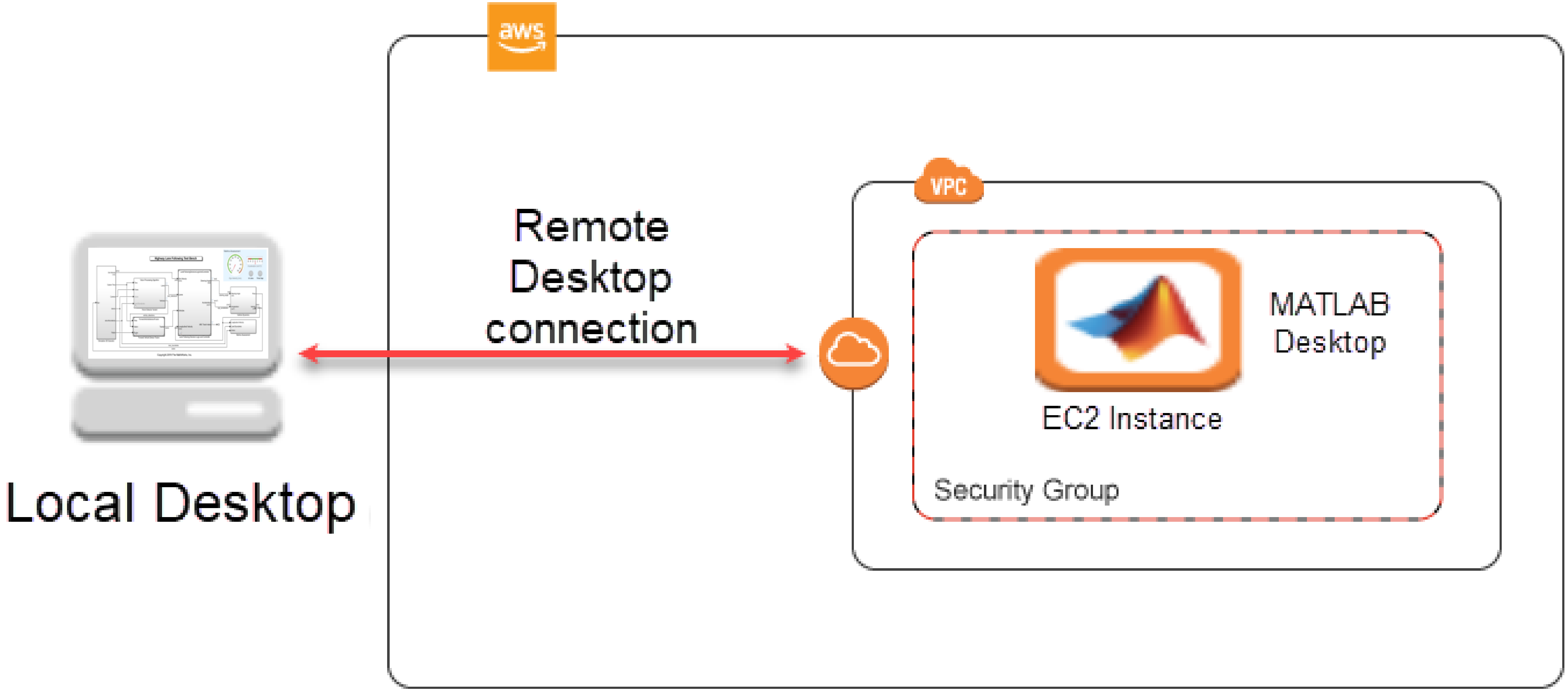
Copyright 2015-2020 The MathWorks, Inc.

Learn more:

[Reduced Order Modeling]

Simulate & Analyze: Cloud-based ROM Generation

Create Vehicle → Integrate Software → Author Scenarios → **Simulate & Analyze** → Deploy Simulation

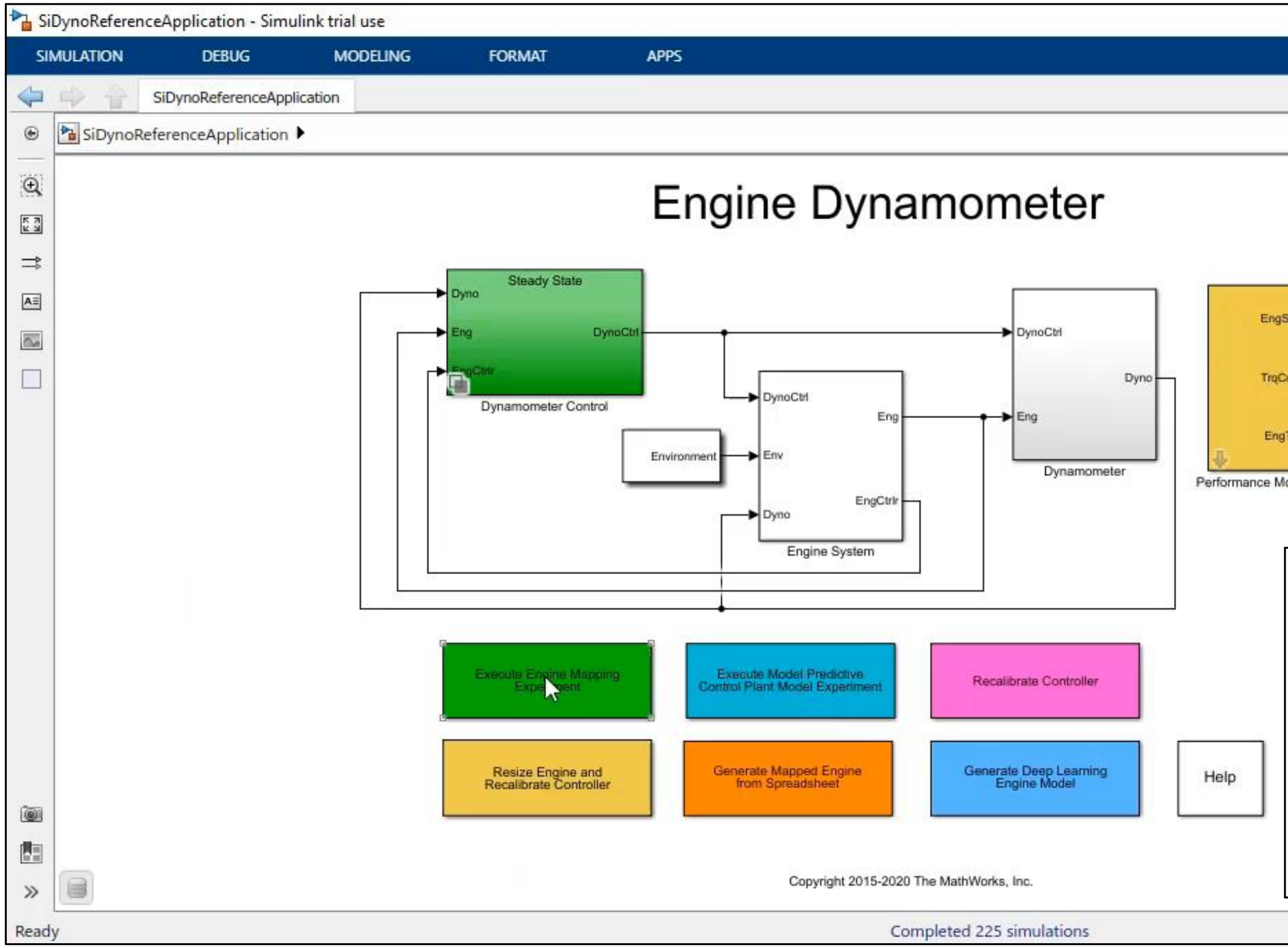


Leverage a Pre-built Cloud Configuration via Reference Architecture

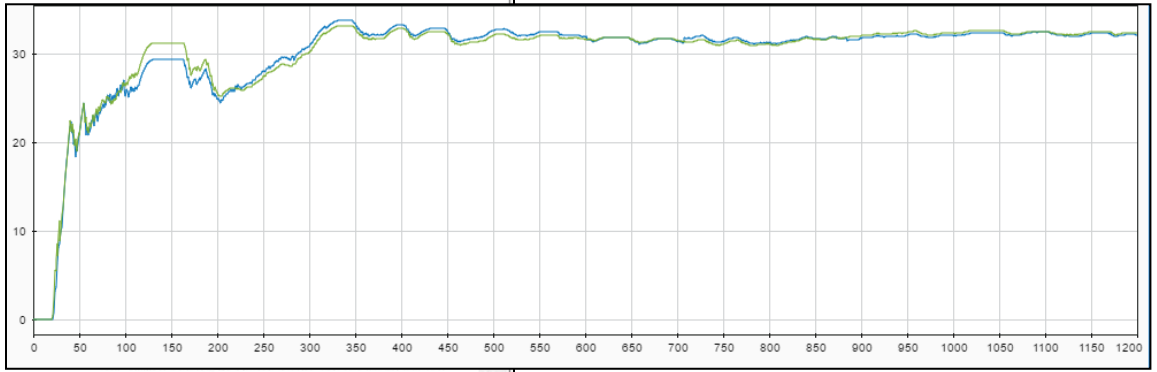
Learn more:
[MATLAB on Amazon Web Services](#) (AWS)

Simulate & Analyze: Cloud-based ROM Generation

Create Vehicle → Integrate Software → Author Scenarios → **Simulate & Analyze** → Deploy Simulation



SiDynoReferenceApplication - Simulink trial use
SIMULATION
DEBUG
MODELING
FORMAT
APPS
SiDynoReferenceApplication
SiDynoReferenceApplication
Engine Dynamometer
Steady State
Dyno
Eng
EngCtrl
DynoCtrl
Dynamometer Control
Environment
DynoCtrl
Env
Eng
Dyno
EngCtrl
Engine System
DynoCtrl
Dyno
Eng
Dynamometer
EngSpd
TrqCmd
EngTrq
Performance Monitor
Execute Engine Mapping Experiment
Execute Model Predictive Control Plant Model Experiment
Recalibrate Controller
Resize Engine and Recalibrate Controller
Generate Mapped Engine from Spreadsheet
Generate Deep Learning Engine Model
Help
Copyright 2015-2020 The MathWorks, Inc.
Ready
Completed 225 simulations
94%



ROM vs. CAE model on drive cycle test:

- 0.3% fuel economy difference
- 50x faster

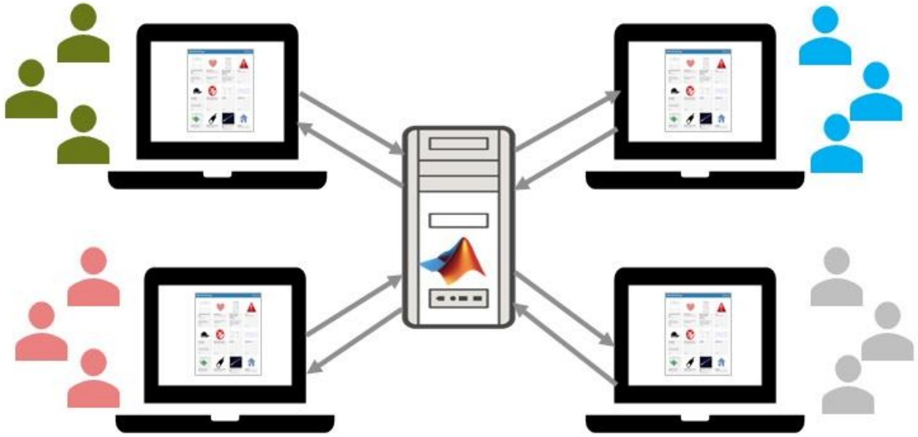
Automate generation of reduced order model in parallel on AWS

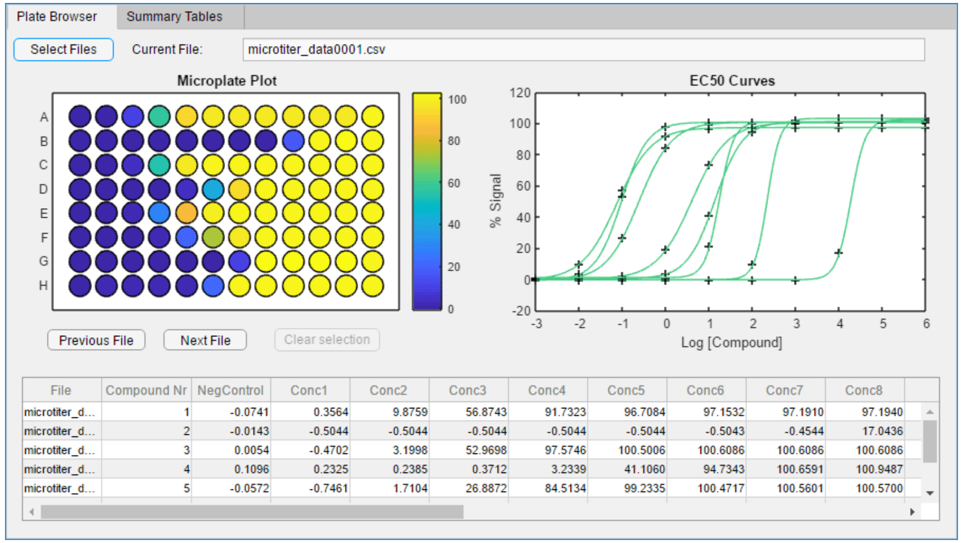
Deploy Simulation: Model Deployment

Create Vehicle → Integrate Software → Author Scenarios → Simulate & Analyze → Deploy Simulation

Use MATLAB and Simulink to take applications farther:

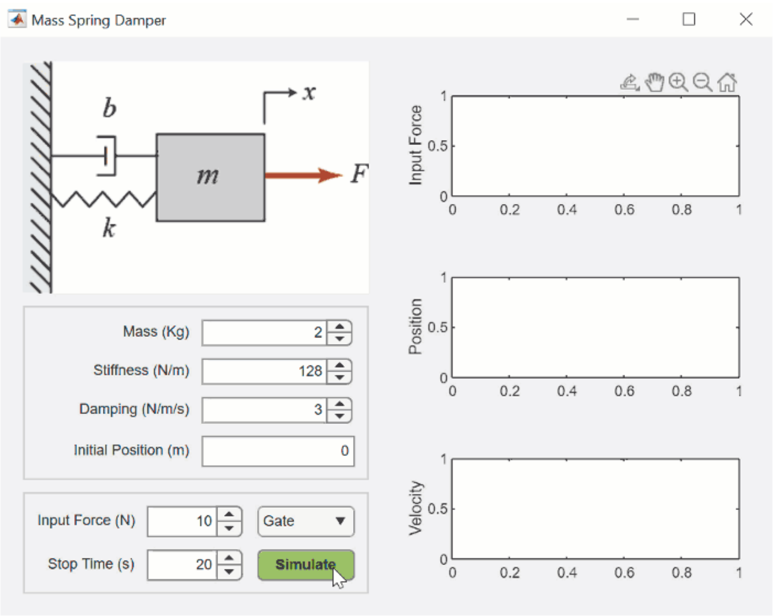
- Create custom UI’s
- Create installers for distribution
- Deploy models as executables, FMU’s or web apps
- Generate code for SIL, HIL testing
- Deploy your Simulink and Simscape models on cloud platforms like Databricks





Current File: microtiter_data0001.csv

File	Compound Nr	NegControl	Conc1	Conc2	Conc3	Conc4	Conc5	Conc6	Conc7	Conc8
microtiter_d...	1	-0.0741	0.3564	9.8759	56.8743	91.7323	96.7084	97.1532	97.1910	97.1940
microtiter_d...	2	-0.0143	-0.5044	-0.5044	-0.5044	-0.5044	-0.5044	-0.5043	-0.4544	17.0436
microtiter_d...	3	0.0054	-0.4702	3.1998	52.9698	97.5746	100.5006	100.6086	100.6086	100.6086
microtiter_d...	4	0.1096	0.2325	0.2385	0.3712	3.2339	41.1060	94.7343	100.6591	100.9487
microtiter_d...	5	-0.0572	-0.7461	1.7104	26.8872	84.5134	99.2335	100.4717	100.5601	100.5700



Mass (Kg) 2
Stiffness (N/m) 128
Damping (N/m/s) 3
Initial Position (m) 0
Input Force (N) 10 Gate
Stop Time (s) 20

Learn more:

[MATLAB Web App Server]

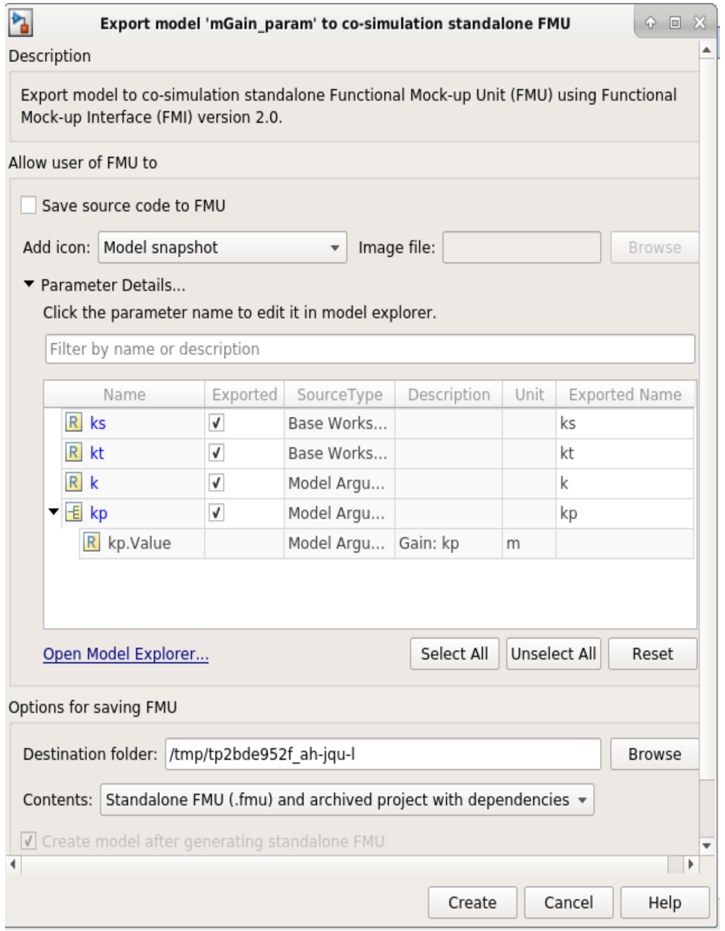
[MATLAB App Designer]

[Simulink Compiler]

[Embedded Systems]

Deploy Simulation: FMU Export

Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation



Export model 'mGain_param' to co-simulation standalone FMU

Description

Export model to co-simulation standalone Functional Mock-up Unit (FMU) using Functional Mock-up Interface (FMI) version 2.0.

Allow user of FMU to

☐ Save source code to FMU

Add icon: Model snapshot   Image file: Browse

▾ Parameter Details...

Click the parameter name to edit it in model explorer.

Filter by name or description

Name	Exported	SourceType	Description	Unit	Exported Name
ks	☑	Base Works...			ks
kt	☑	Base Works...			kt
k	☑	Model Argu...			k
kp	☑	Model Argu...			kp
kp.Value		Model Argu...	Gain: kp	m	

Open Model Explorer...   Select All   Unselect All   Reset

Options for saving FMU

Destination folder: /tmp/tp2bde952f_ah-jqu-l   Browse

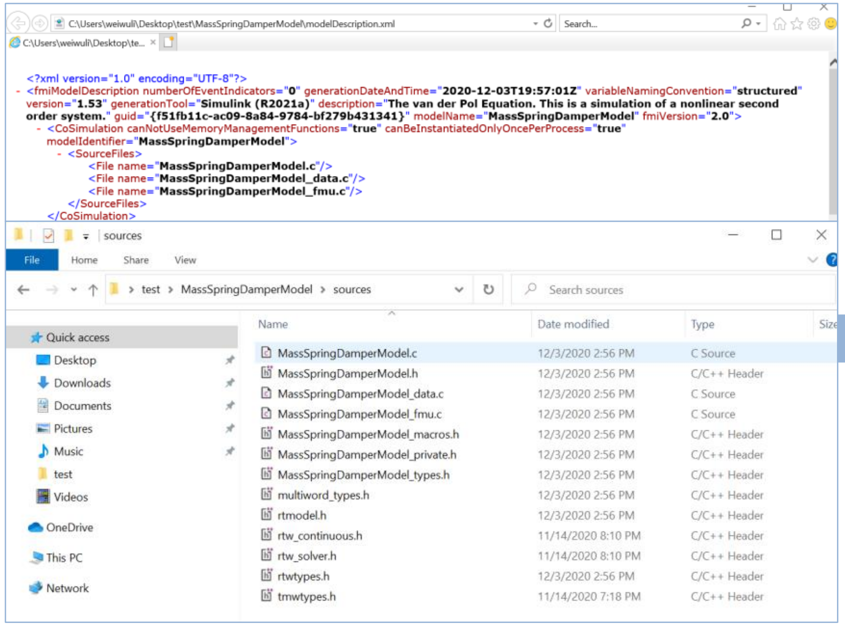
Contents: Standalone FMU (.fmu) and archived project with dependencies

☑ Create model after generating standalone FMU

Create   Cancel   Help

New FMU Export options:

- FMU for co-sim export with C code
- Export model arguments as FMU parameters
- Nested FMU export



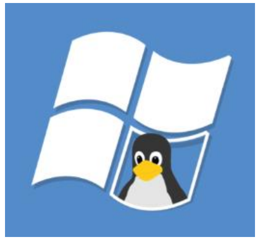
C:\Users\weiwuli\Desktop\test\MassSpringDamperModel\modelDescription.xml

```
  <?xml version="1.0" encoding="UTF-8"?>
- <fmiModelDescription numberOfEventIndicators="0" generationDateAndTime="2020-12-03T19:57:01Z" variableNamingConvention="structured"
  version="1.53" generationTool="Simulink (R2021a)" description="The van der Pol Equation. This is a simulation of a nonlinear second
  order system." guid="{f51fb11c-ac09-8a84-9784-bf279b431341}" modelName="MassSpringDamperModel" fmiVersion="2.0">
  - <CoSimulation canNotUseMemoryManagementFunctions="true" canBeInstantiatedOnlyOncePerProcess="true"
    modelIdentifier="MassSpringDamperModel">
    - <SourceFiles>
          <File name="MassSpringDamperModel.c"/>
          <File name="MassSpringDamperModel_data.c"/>
          <File name="MassSpringDamperModel_fmu.c"/>
      </SourceFiles>
    </CoSimulation>
```

sources

test › MassSpringDamperModel › sources

Name	Date modified	Type
MassSpringDamperModel.c	12/3/2020 2:56 PM	C Source
MassSpringDamperModel.h	12/3/2020 2:56 PM	C/C++ Header
MassSpringDamperModel_data.c	12/3/2020 2:56 PM	C Source
MassSpringDamperModel_fmu.c	12/3/2020 2:56 PM	C Source
MassSpringDamperModel_macros.h	12/3/2020 2:56 PM	C/C++ Header
MassSpringDamperModel_private.h	12/3/2020 2:56 PM	C/C++ Header
MassSpringDamperModel_types.h	12/3/2020 2:56 PM	C/C++ Header
multiword_types.h	12/3/2020 2:56 PM	C/C++ Header
rtmodel.h	12/3/2020 2:56 PM	C/C++ Header
rtw_continuous.h	11/14/2020 8:10 PM	C/C++ Header
rtw_solver.h	11/14/2020 8:10 PM	C/C++ Header
rtwtypes.h	12/3/2020 2:56 PM	C/C++ Header
tmwtypes.h	11/14/2020 7:18 PM	C/C++ Header



Deploy desktop simulation on different OS

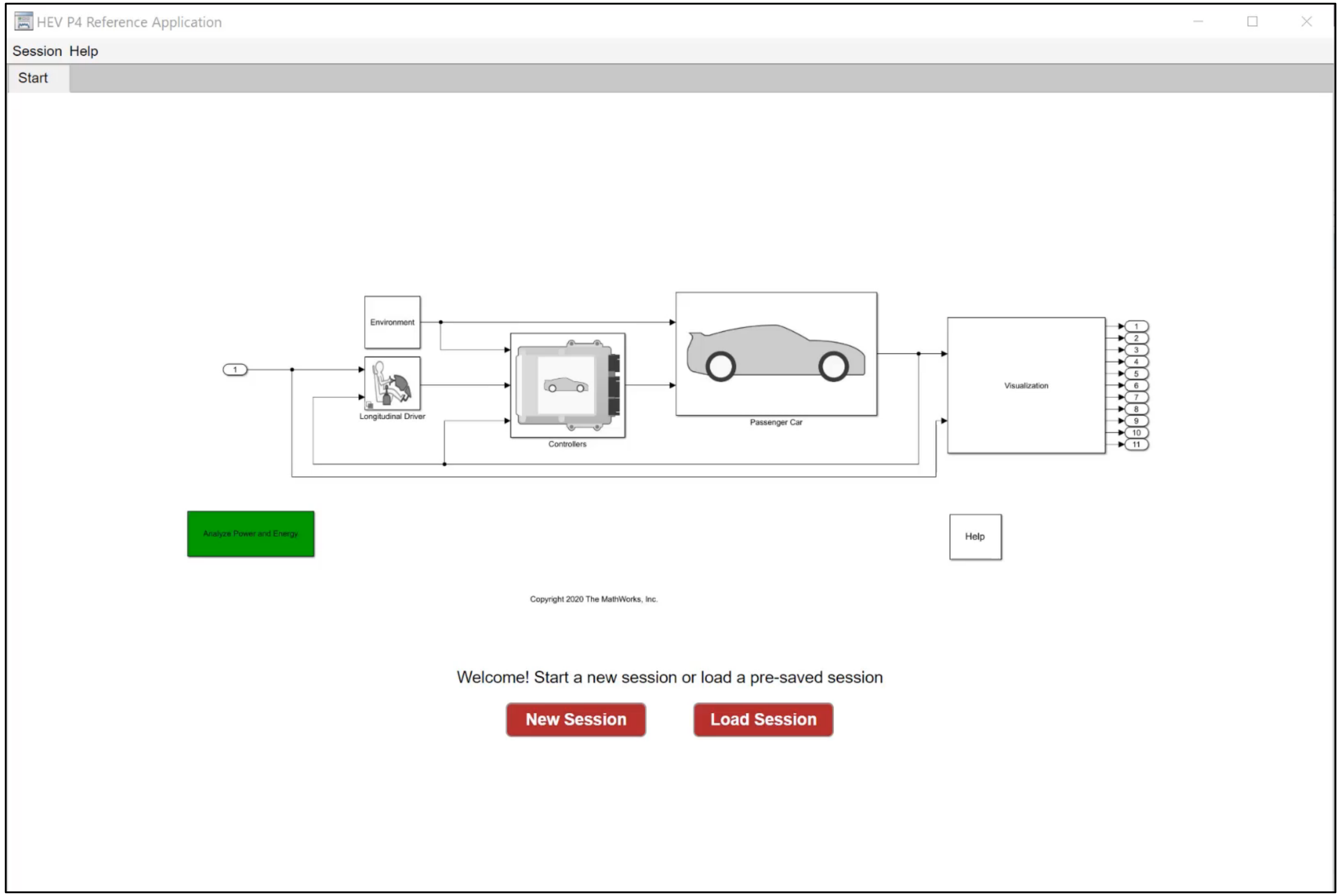
Real-time simulation

Learn more:

Co-simulation

Deploy Simulation: Web Apps

Create Vehicle → Integrate Software → Author Scenarios → Simulate & Analyze → Deploy Simulation



HEV P4 Reference Application

Welcome! Start a new session or load a pre-saved session

Deployed HEV model via web app:

- Start / load session
- Select drive cycle
- Specify key parameters
- Launch simulations on server
- Compare results
- Generate reports

Benefits:

- Purpose-built user interface
- Convenient way to share virtual vehicle with non-expert tool users

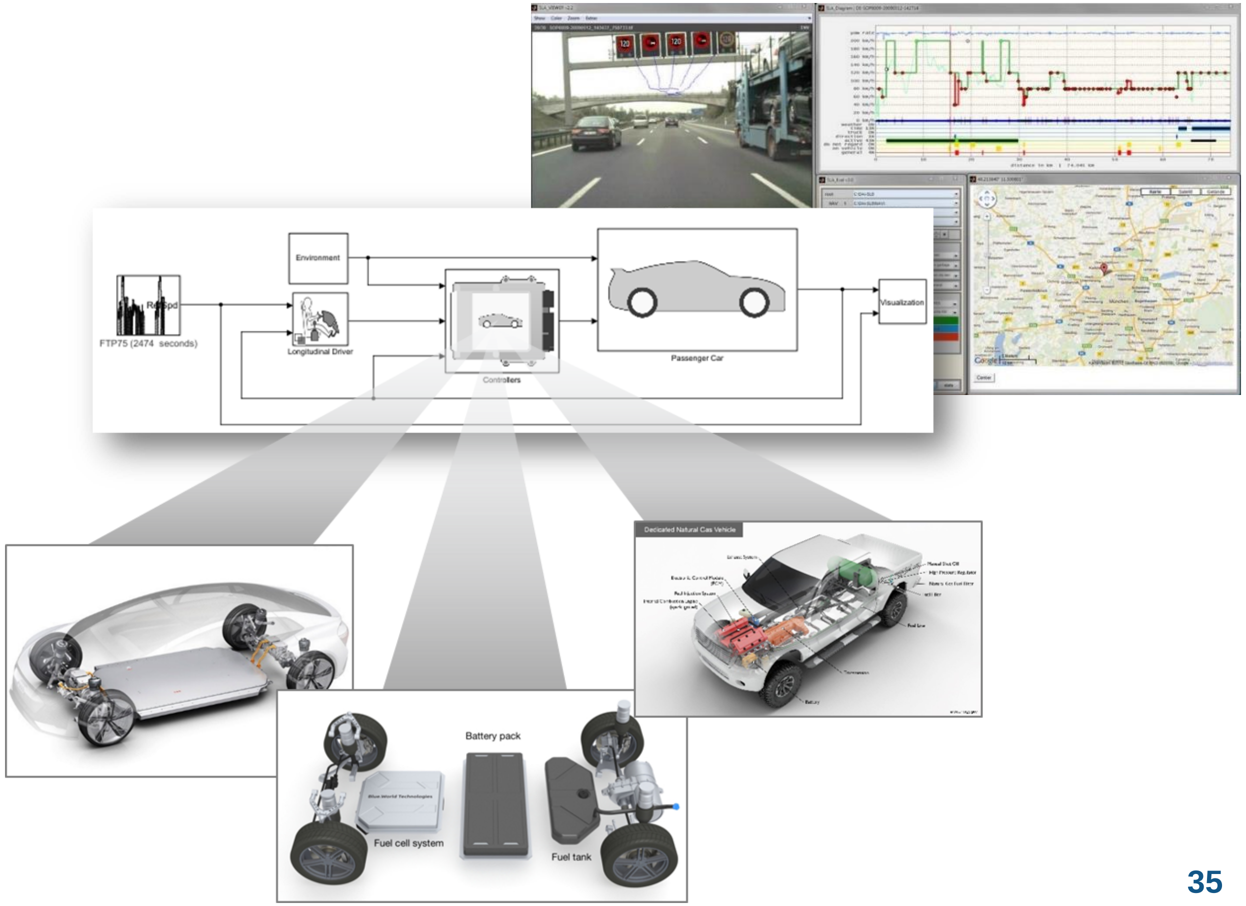
MathWorks Consulting Services Can Support You

Model Architecture
- Model assessment
- Simulation performance
- Interface standardization
- …

Construction
- Build process automation
- Database/Repo interface
- Model-Building know-how
- …

User Experience
- GUI driven workflow
- Tool compatibility support
- Artifact creation
- …

- Provide expert-level guidance
- Automate workflows
- Develop custom UI's



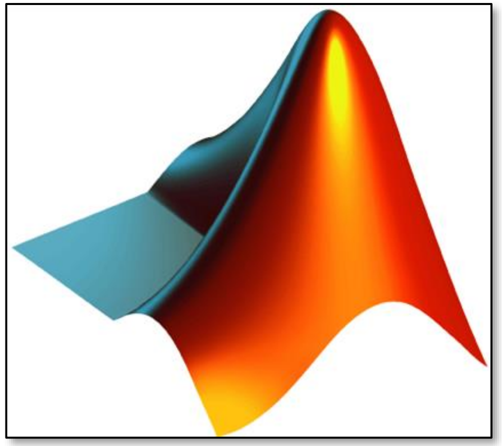
Learn more:

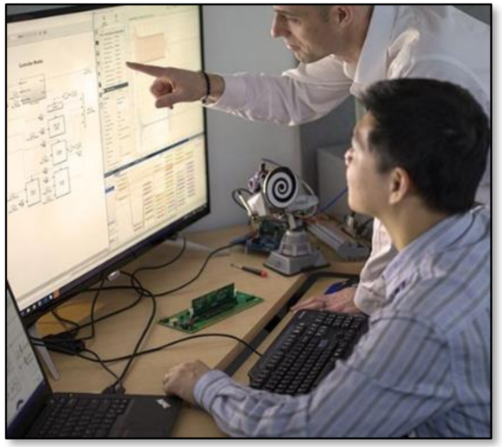
[MathWorks Consulting Services]

Key Takeaways

MathWorks provides a **powerful platform** for building your **Virtual Vehicle**

Our platform is very **flexible**, and we can help you **customize** it for your needs





Out-of-the-box capability

Custom virtual vehicle solution

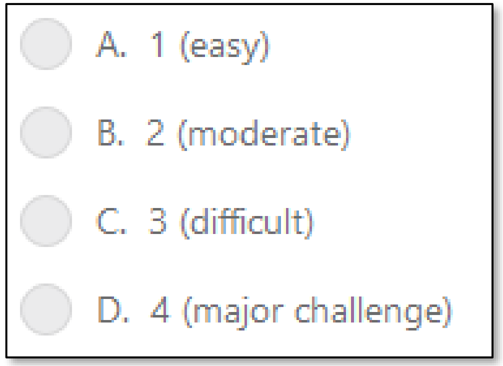
Presenter Contact Info and Poll Questions

Please contact us with questions

Brad Hieb
Application Engineer
bhieb@mathworks.com

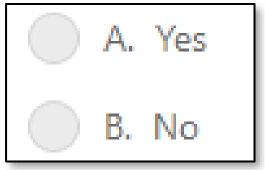
Mike Sasena, PhD
Automotive Product Manager
msasena@mathworks.com

- On a scale of 1 - 4, how challenging is it for your department to:
  - Create the vehicle model
  - Integrate software
  - Author scenarios
  - Simulate and analyze results
  - Deploy simulations



○ A. 1 (easy)
○ B. 2 (moderate)
○ C. 3 (difficult)
○ D. 4 (major challenge)

- Are you interested in a follow-up conversation with MathWorks?



○ A. Yes
○ B. No

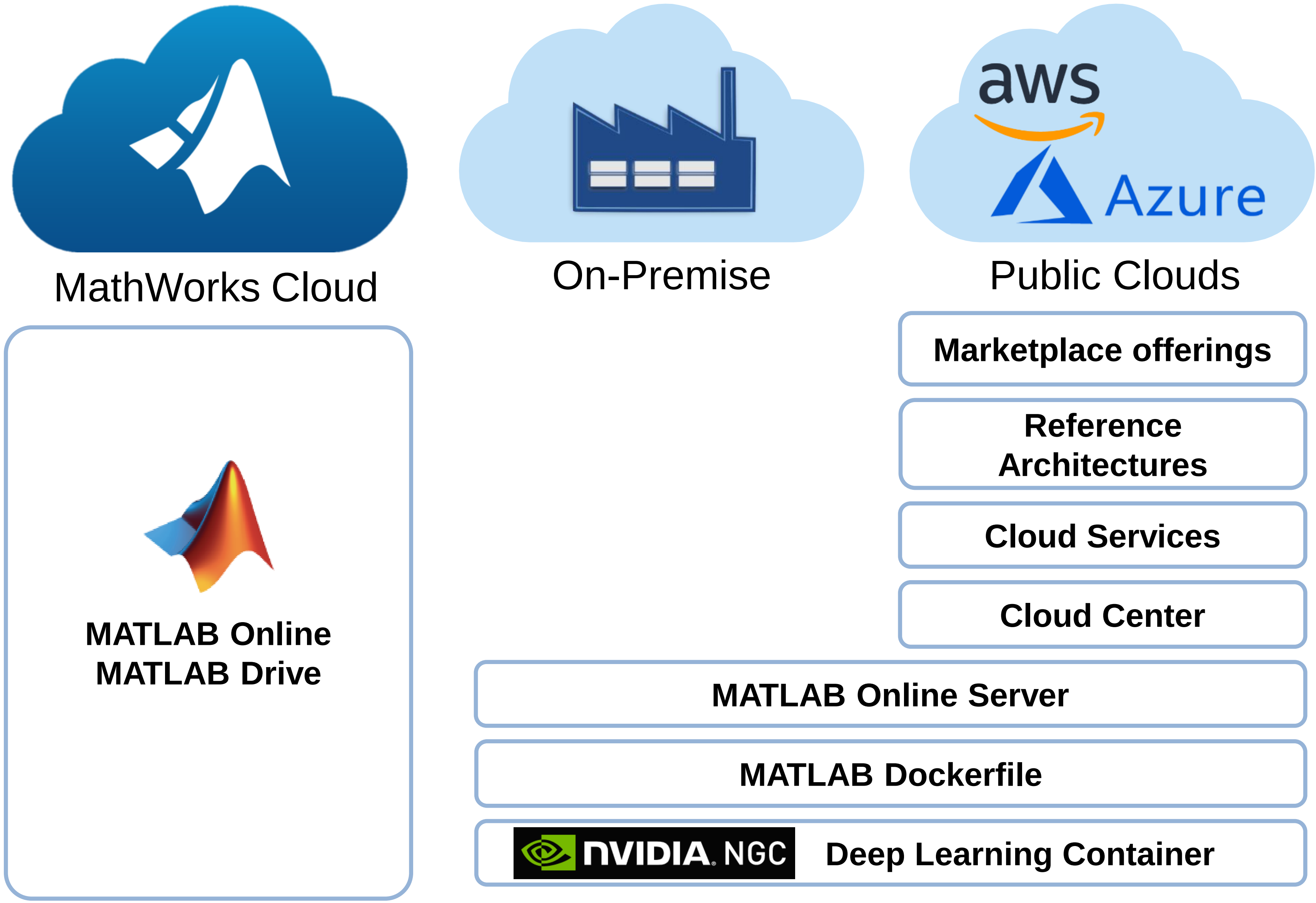
- Additional comments

Thank You

Brad Hieb
Application Engineer
bhieb@mathworks.com

Mike Sasena, PhD
Automotive Product Manager
msasena@mathworks.com

Whatever your cloud environment, MATLAB can work and scale



MathWorks Cloud

MATLAB Online
MATLAB Drive

On-Premise

Public Clouds

Marketplace offerings

Reference Architectures

Cloud Services

Cloud Center

MATLAB Online Server

MATLAB Dockerfile

Deep Learning Container

MATLAB/Simulink on the Cloud

amazon web services

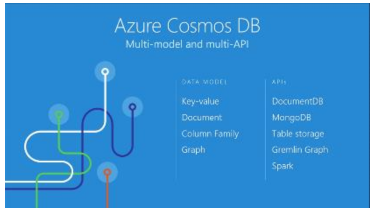
Google Cloud

Microsoft Azure

MATLAB Cloud Service Support

Amazon S3

Microsoft Azure Blob Storage


Azure Cosmos DB
Multi-model and multi-API

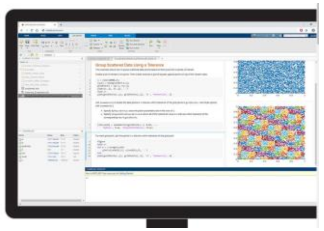
DATA MODEL
Key-value
Document
Column Family
Graph

APIs
DocumentDB
MongoDB
Table storage
Gremlin Graph
Spark

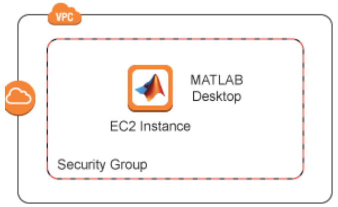
Amazon Athena

etc.

Accessibility

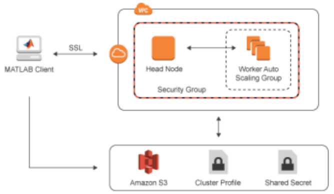


MATLAB Online Server
matlab.boeing.com



MATLAB Reference Architecture on Cloud

Scalability



MATLAB Parallel Server Reference Architecture

Deployment

MATLAB Production Server Cloud Console

Log In

Username

Password

MATLAB Production Server Reference Architecture

Additional Resources

- MathWorks Reference Architectures
  - Network License Manager for MATLAB on AWS
  - Network License Manager for MATLAB on Azure
  - Run MATLAB on AWS
  - Run MATLAB on Azure
  - MATLAB Parallel Server cluster on AWS
  - MATLAB Parallel Server cluster on Azure
  - MATLAB Production Server on AWS
  - MATLAB Production Server on Azure
- MATLAB Online Server (can run on AWS, Azure and Google cloud)