Simulink for Virtual Vehicle Development

30 June

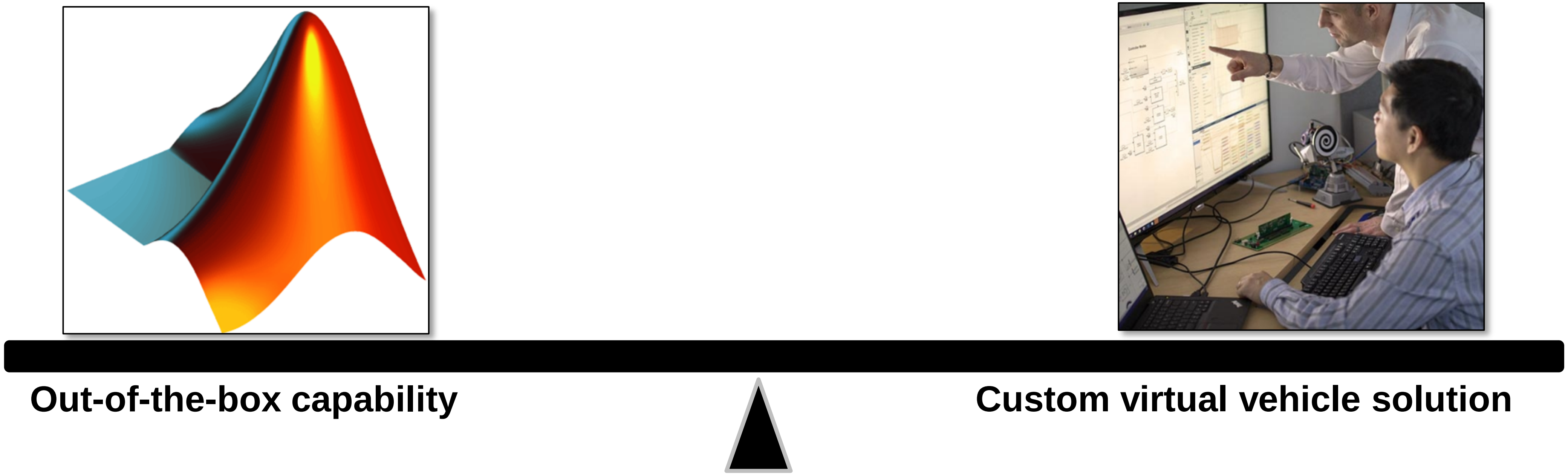
MathWorks
AUTOMOTIVE
CONFERENCE 2020

MathWorks®

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

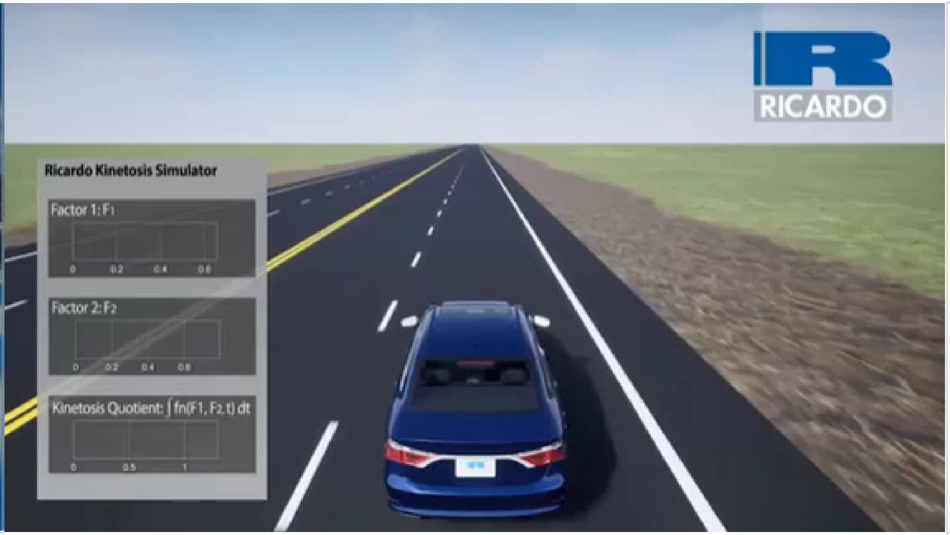


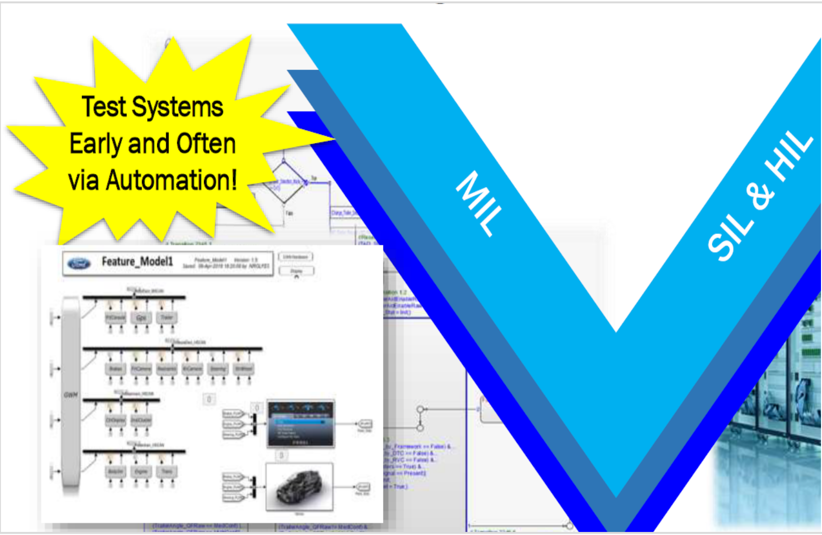
Out-of-the-box capability

Custom virtual vehicle solution

Virtual vehicle: functional simulation of full vehicle behaviors

Reduced years of effort and expensive prototypes


RICARDO
Ricardo Kinetosis Simulator
Factor 1: F1
Factor 2: F2
Kinetosis Quotient: ∫ fn(F1, F2, t) dt



Tesla: vehicle design tradeoff

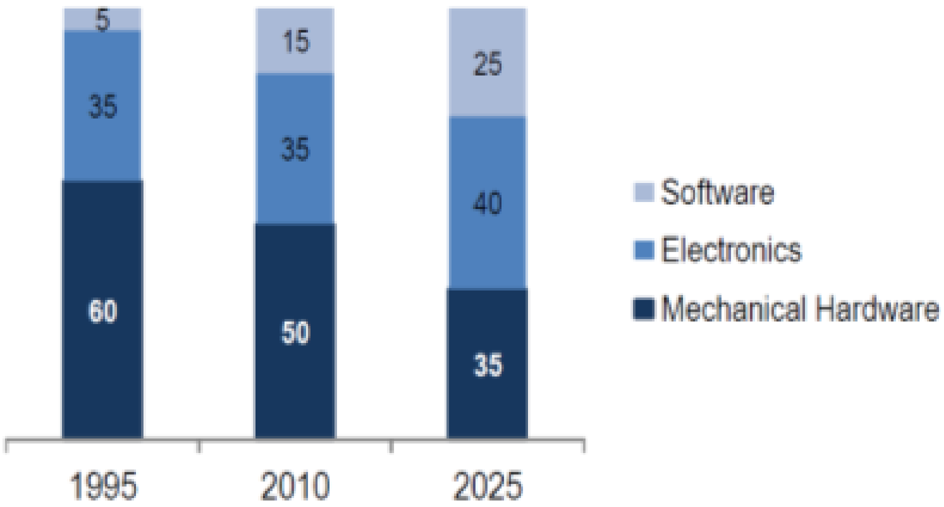
Ricardo: simulating passenger comfort

Ford: software validation

Reduce physical testing needed before design validation

Embedded software is essential for many virtual vehicle applications

Share of Value Add per Vehicle (%)



Source: BMW

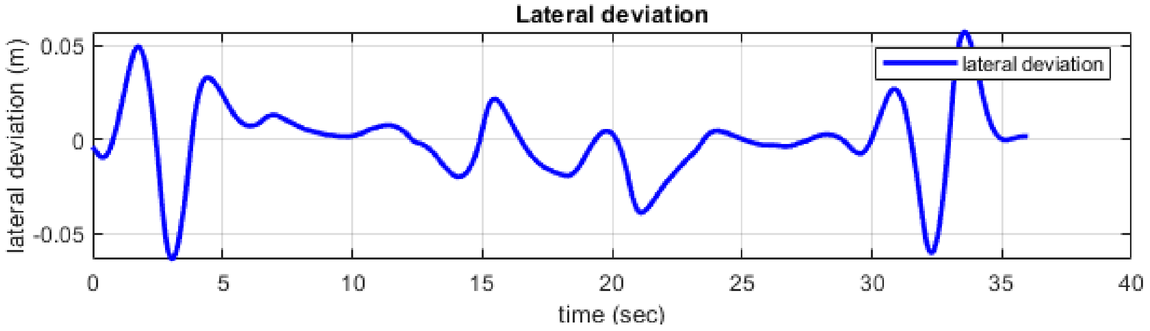
Virtual vehicle applications such as attribute development, software validation, calibration **require simulation of embedded software**.

- Application software behavior fully represented
- Interfaces consistent with software component definitions

Example: Validating lane following software functional safety requirement (FSR)

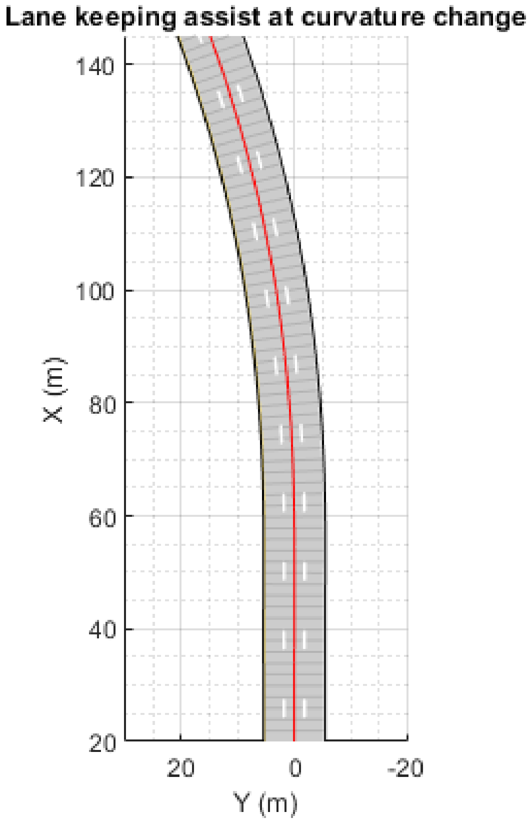
?

FSR: The lane following system lateral error shall be less than 1 meter



Questions to consider:

- System performance under normal conditions?
- Impact of environment conditions?
- Impact of a component failure?
- Required processor throughput?

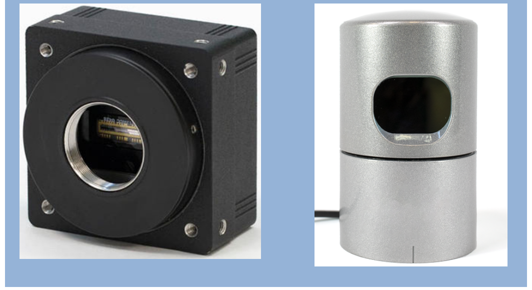


System level interactions need to be considered

?

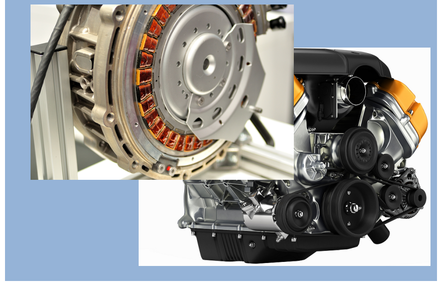
FSR: The lane following system lateral error shall be less than 1 meter

Sensors



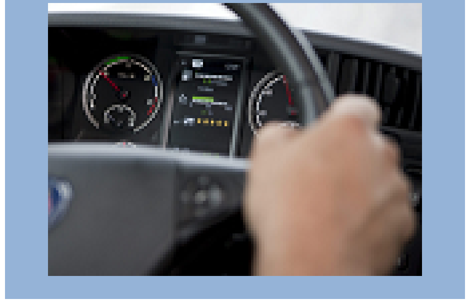
Controllers

Powertrain



Environment

Driver



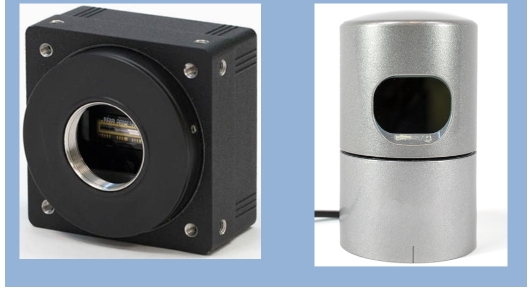
Vehicle

System level testing typically occurs with hardware integration

?

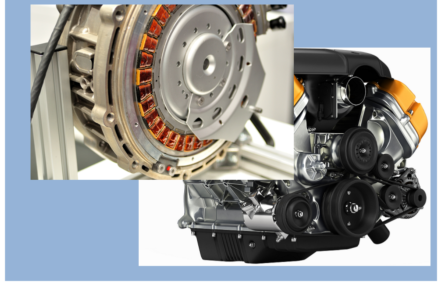
FSR: The lane following system lateral error shall be less than 1 meter

Sensors



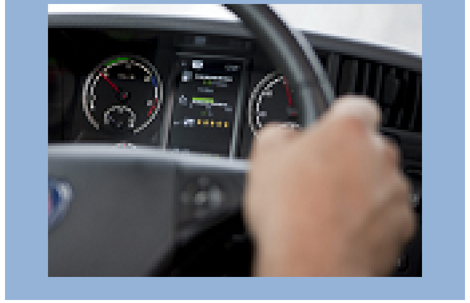
Controllers

Powertrain



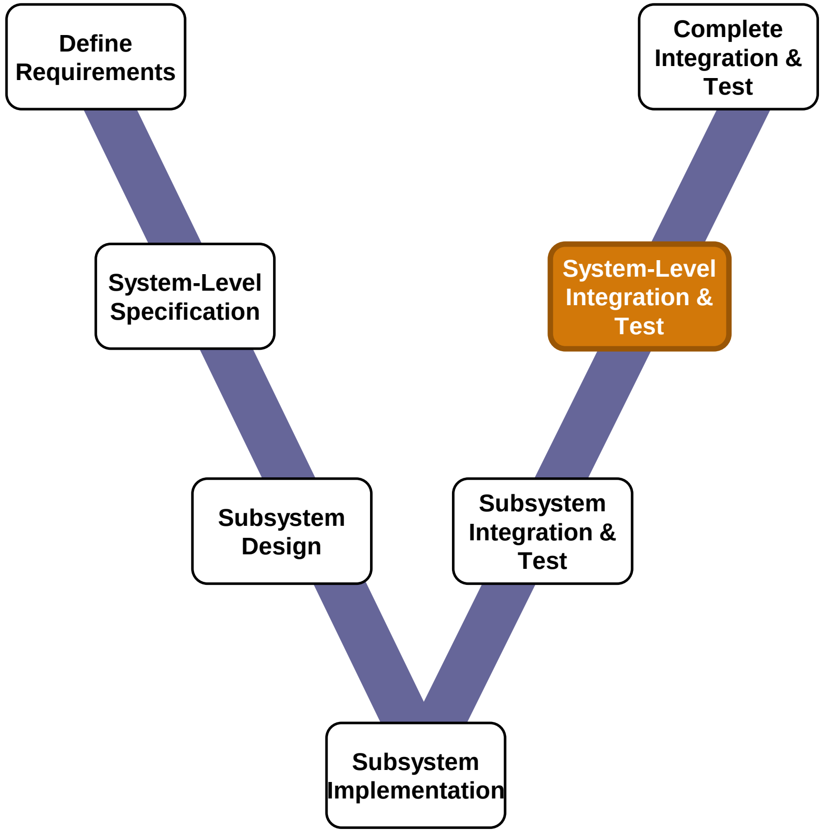
Environment

Driver



Vehicle

Discovering problems during system-level integration is ***expensive***



Define Requirements

System-Level Specification

Subsystem Design

Subsystem Implementation

Subsystem Integration & Test

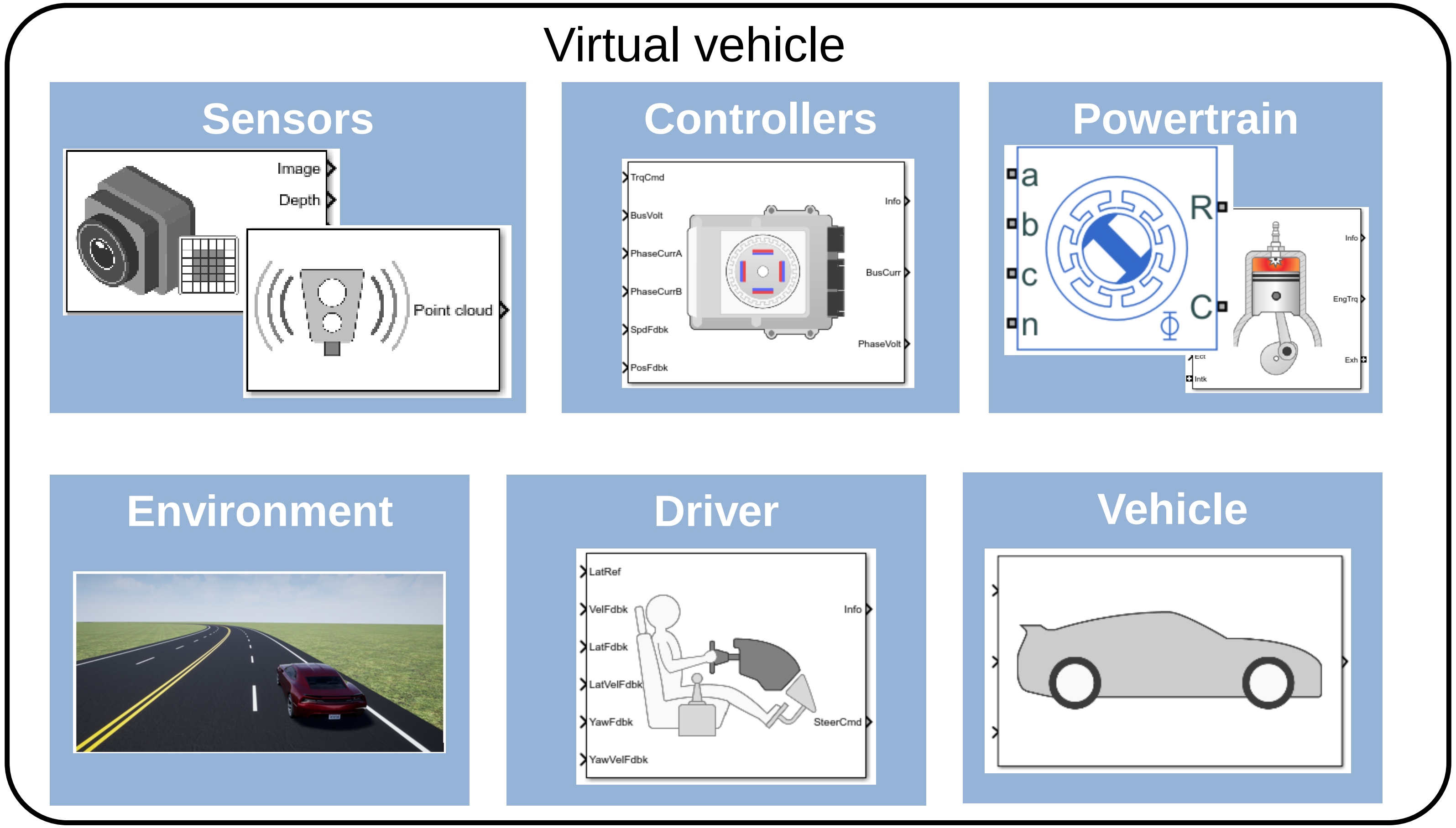
System-Level Integration & Test

Complete Integration & Test

Validate software against function safety requirements early

FSR: The lane following system lateral error shall be less than 1 meter

*Use simulation to do system-level integration testing **early***



Virtual vehicle

Sensors

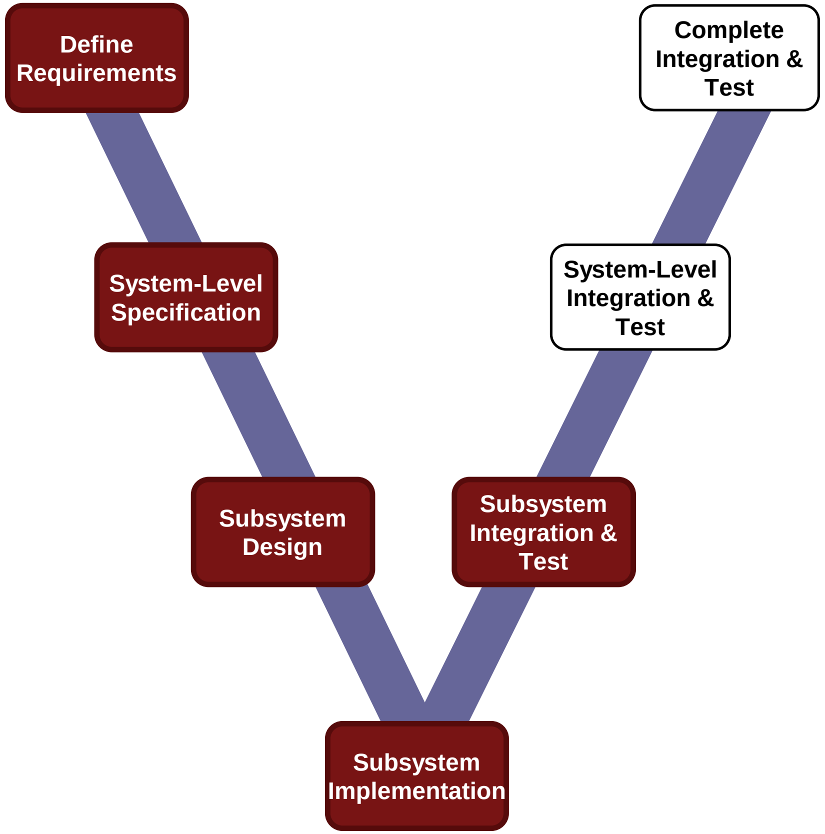
Controllers

Powertrain

Environment

Driver

Vehicle



Define Requirements

System-Level Specification

Subsystem Design

Subsystem Implementation

Subsystem Integration & Test

System-Level Integration & Test

Complete Integration & Test

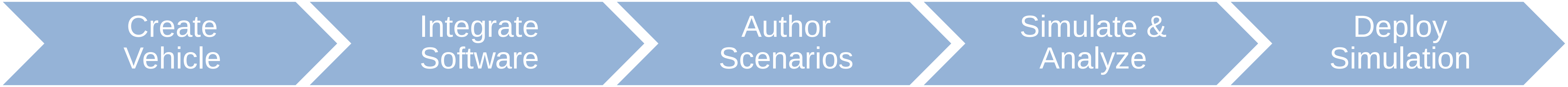
Agenda

- Common challenges
- MathWorks solutions
- Case study

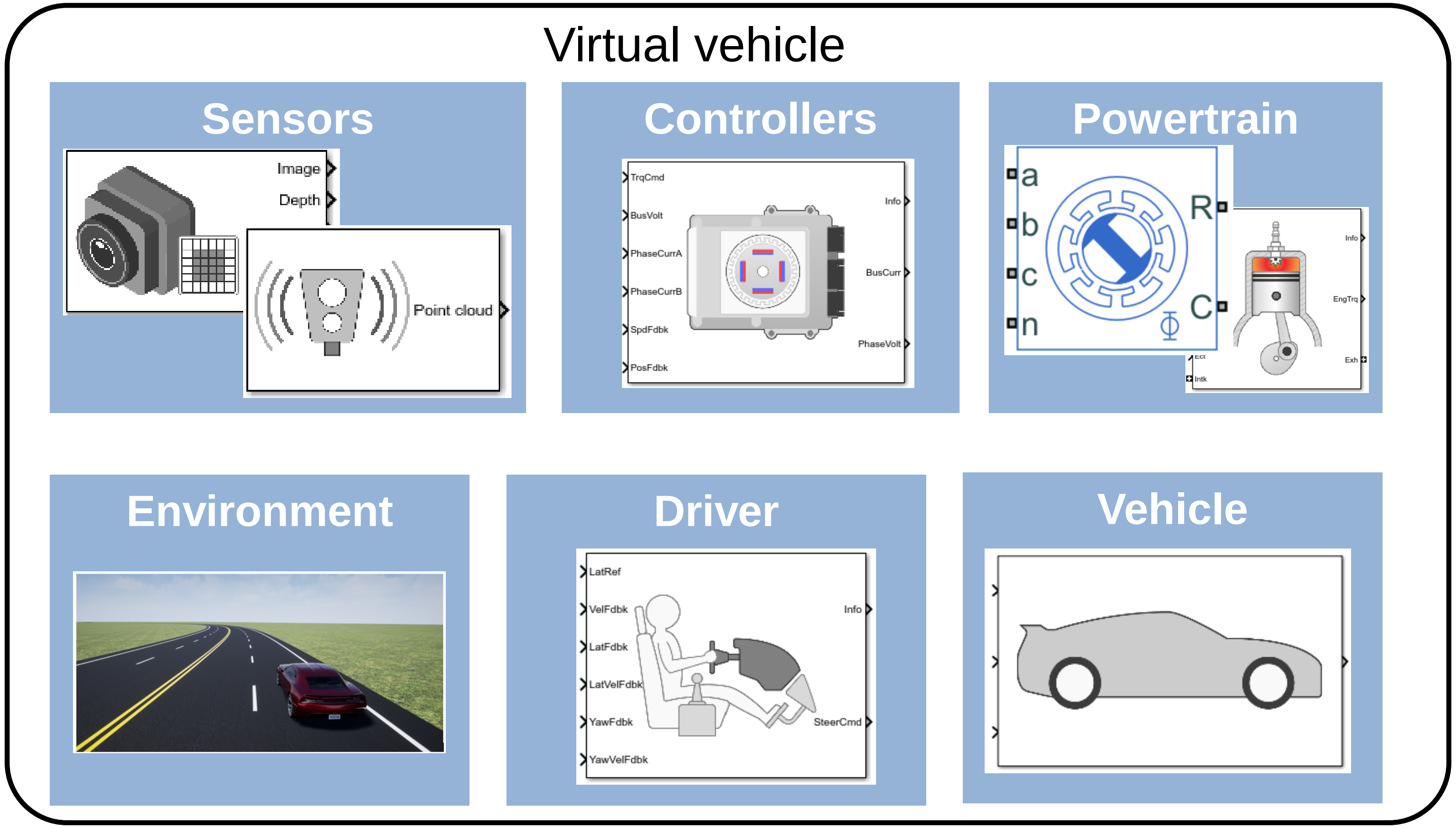
Agenda

- **Common challenges**
- MathWorks solutions
- Case study

Challenges to early system-level testing



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



Virtual vehicle

Sensors

Controllers

Powertrain

Environment

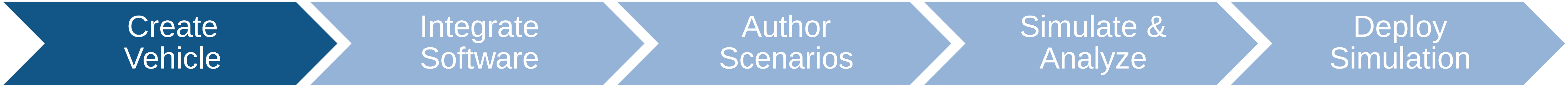
Driver

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

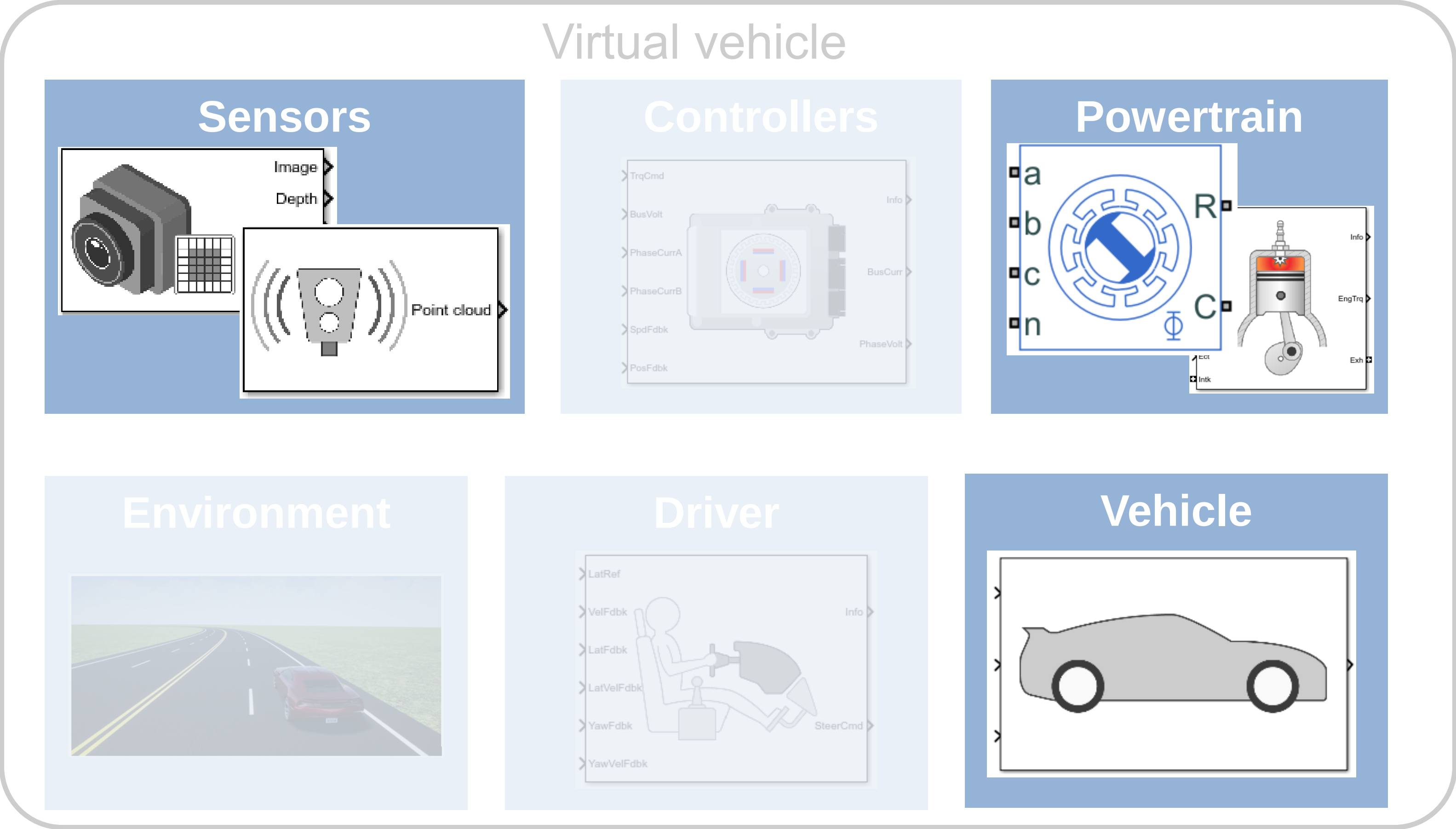

Create Vehicle

Integrate Software

Author Scenarios

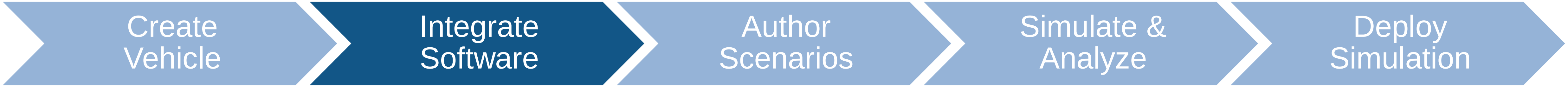
Simulate & Analyze

Deploy Simulation



- 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



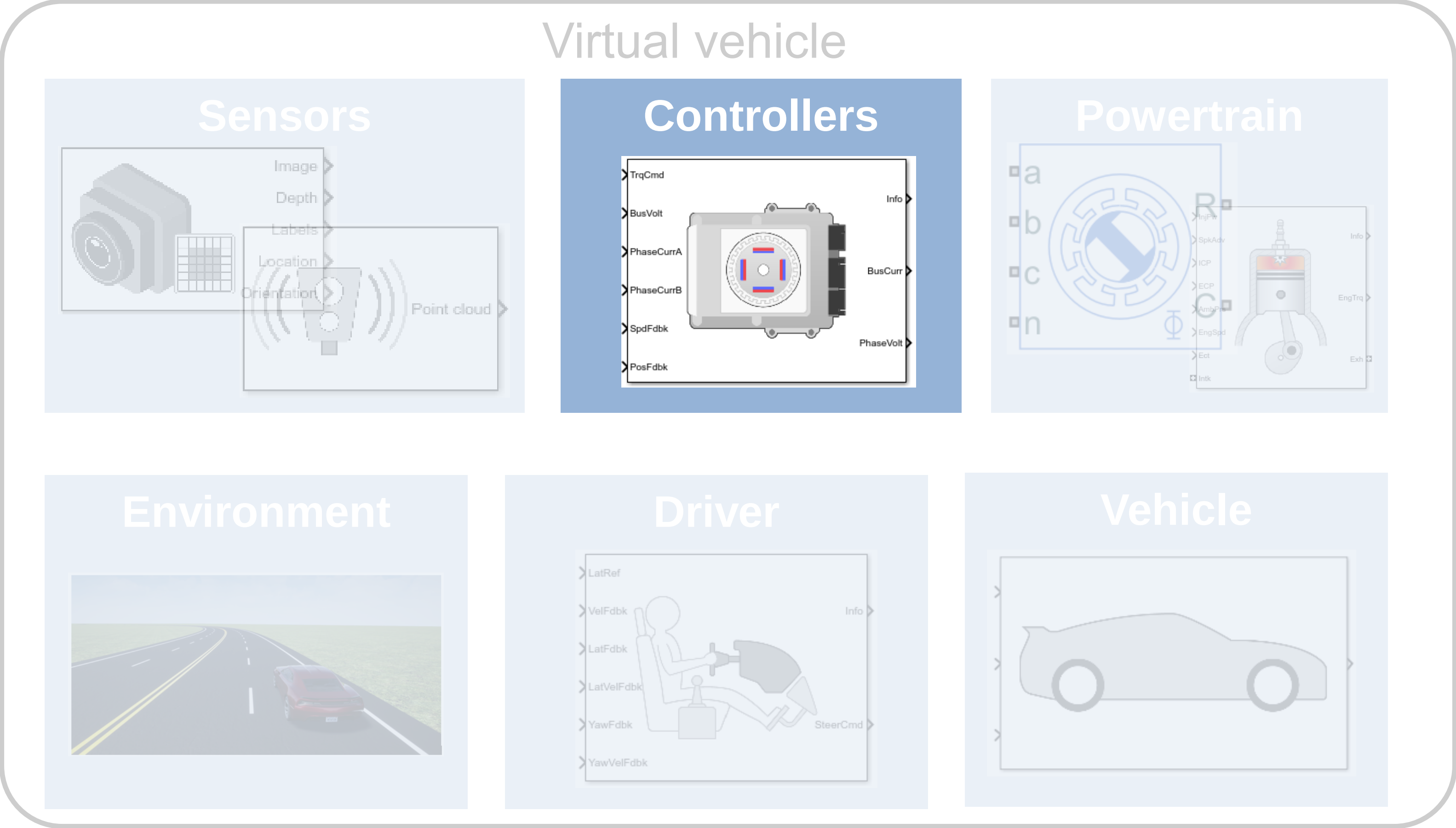
Create Vehicle

Integrate Software

Author Scenarios

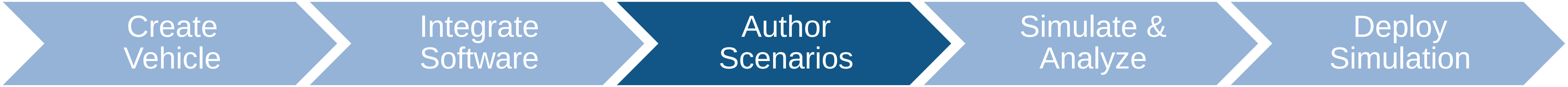
Simulate & Analyze

Deploy Simulation



- 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

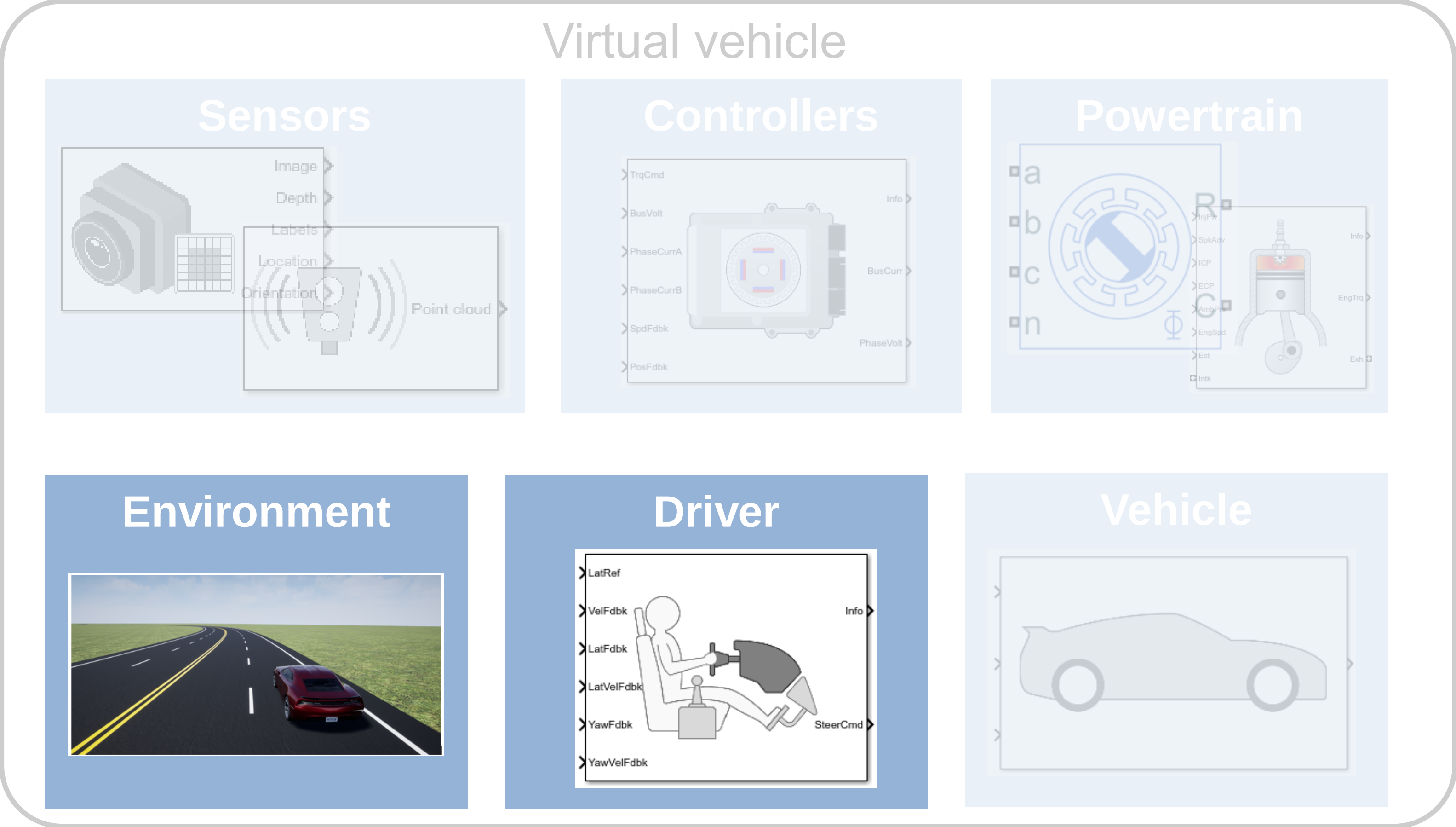

Create Vehicle

Integrate Software

Author Scenarios

Simulate & Analyze

Deploy Simulation


Virtual vehicle

Sensors

Controllers

Powertrain

Environment

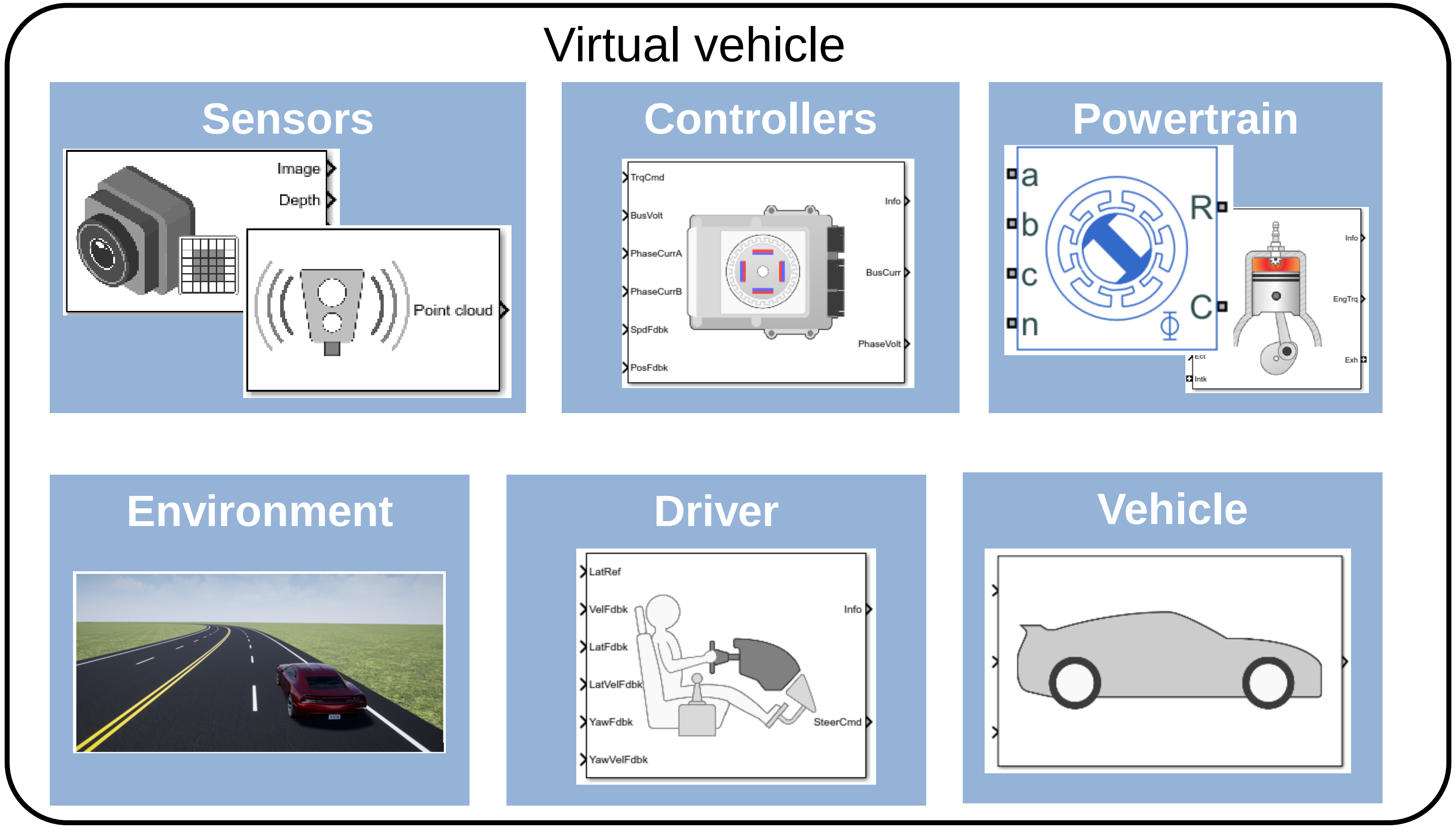
Driver

Vehicle

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

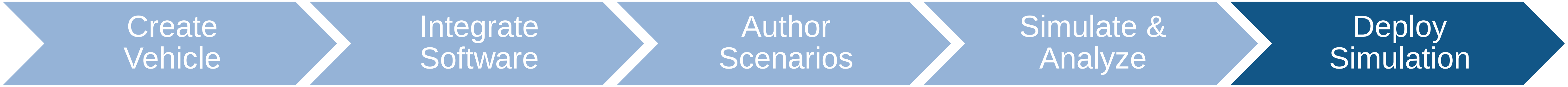
Challenges to early system-level testing

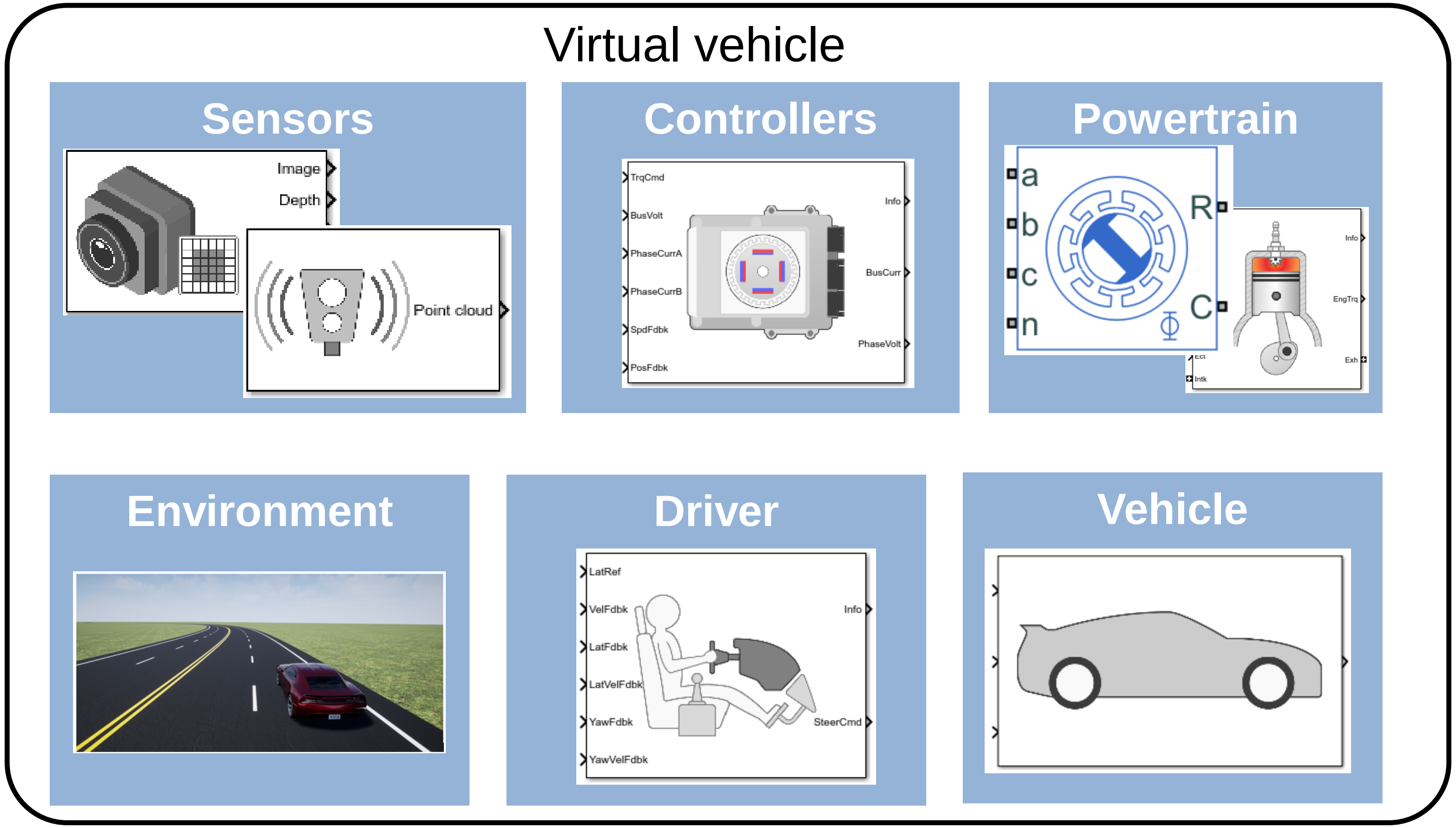
Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation



- 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


Virtual vehicle

Sensors

Controllers

Powertrain

Environment

Driver

Vehicle

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

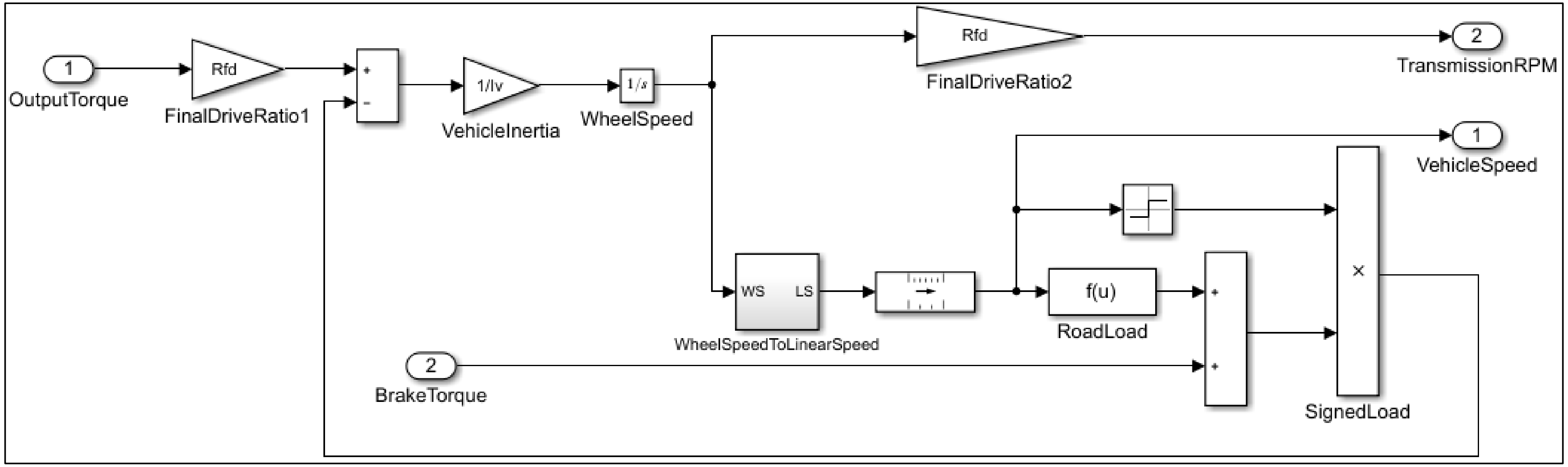
Agenda

- Common challenges
- **MathWorks solutions**
- Case study

MathWorks Virtual Vehicle: reference applications

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

- Start with in-house vehicle models
  - We can help you customize it and apply best practices for Model-Based Design



Learn more:

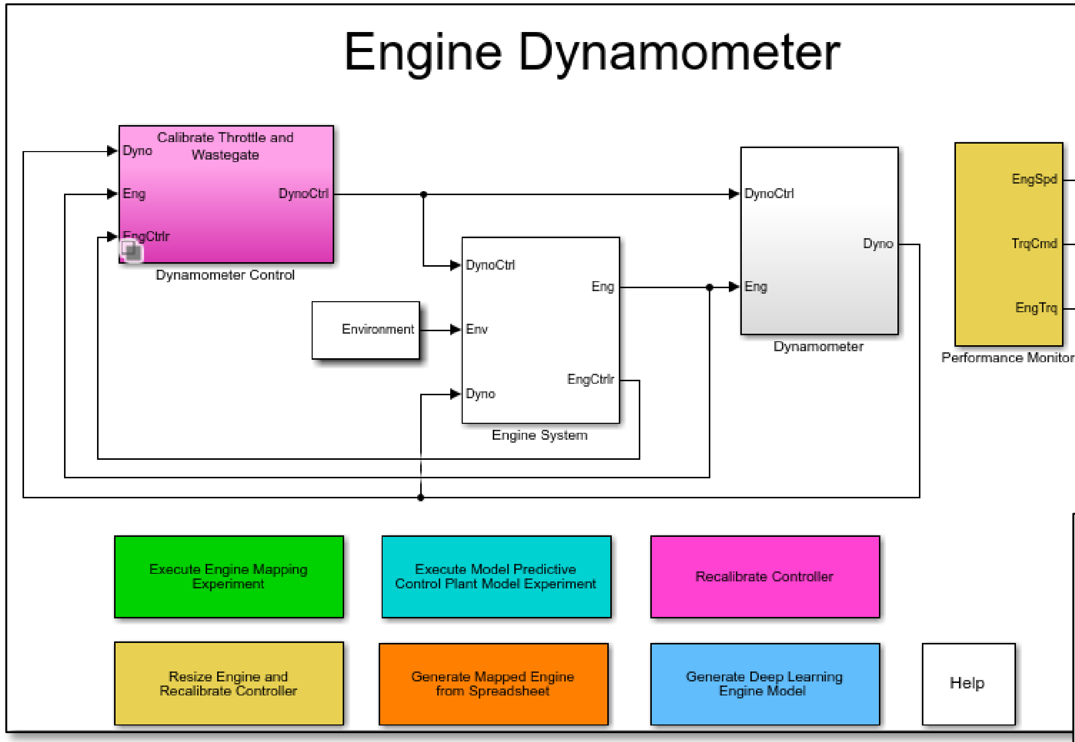
[Powertrain Blockset]

[Vehicle Dynamics Blockset]

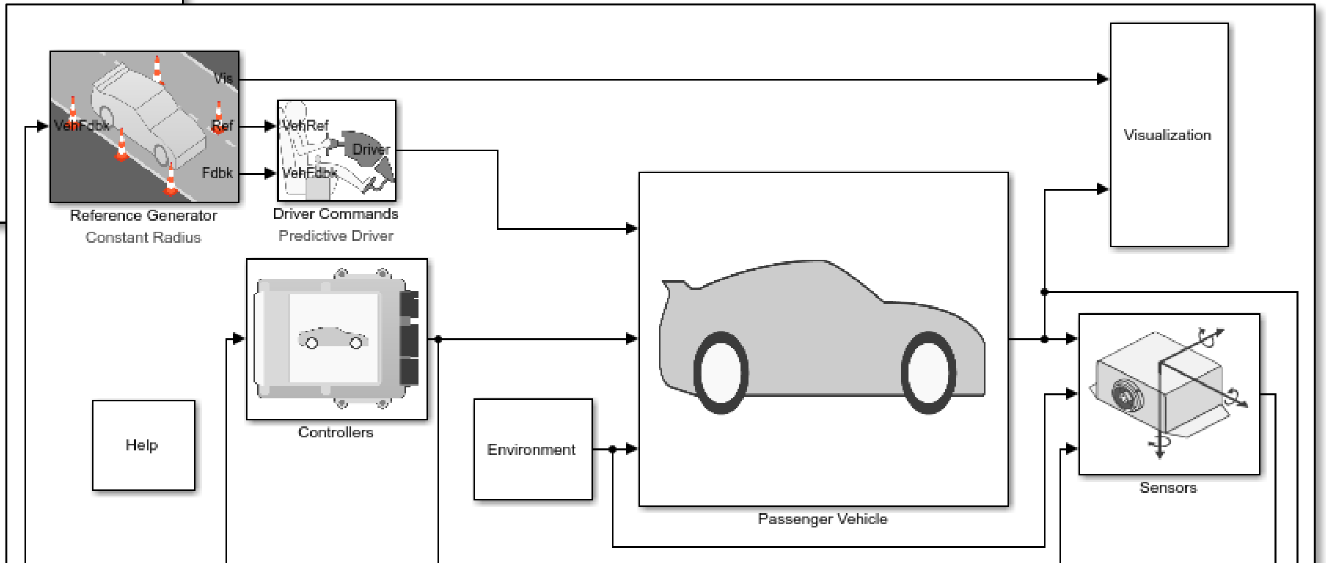
[Automated Driving Toolbox]

MathWorks Virtual 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
Dyno
Eng
Dynamometer
EngSpd
TrqCmd
EngTrq
Performance Monitor
Environment
DynoCtrl
Eng
Env
EngCtrlr
Dyno
Engine System
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

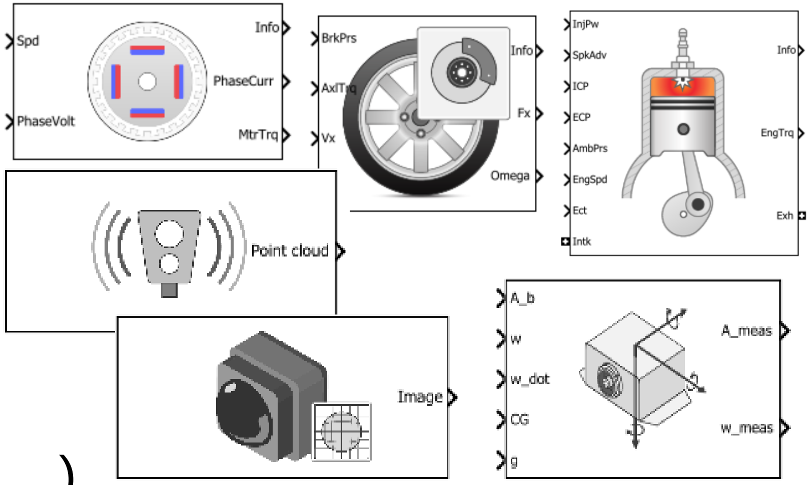
Automated Driving Toolbox

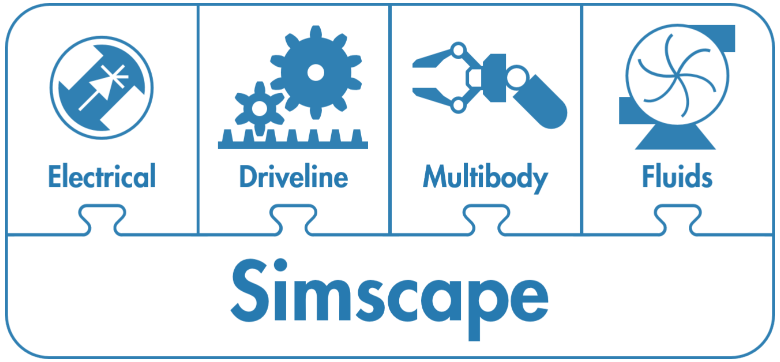
MathWorks Virtual Vehicle: model customization

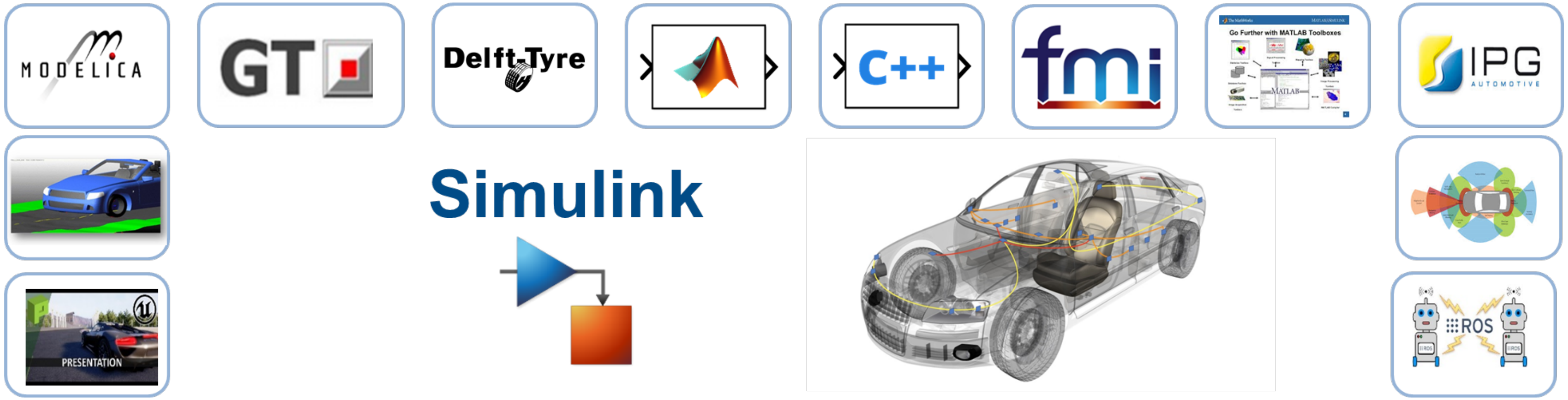
Create Vehicle → Integrate Software → Author Scenarios → Simulate & Analyze → Deploy Simulation

Add detail where needed using:

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







Learn more:

[Simscape]

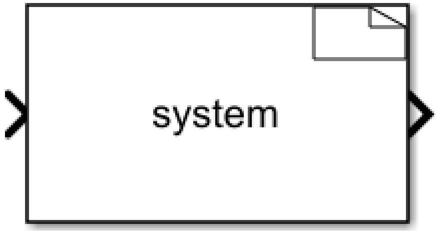
[Multi-core cosim]

[Integrate with existing sims]

MathWorks Virtual Vehicle: C code integration

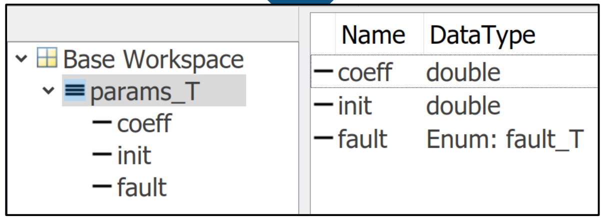
Integrate controller algorithms:

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

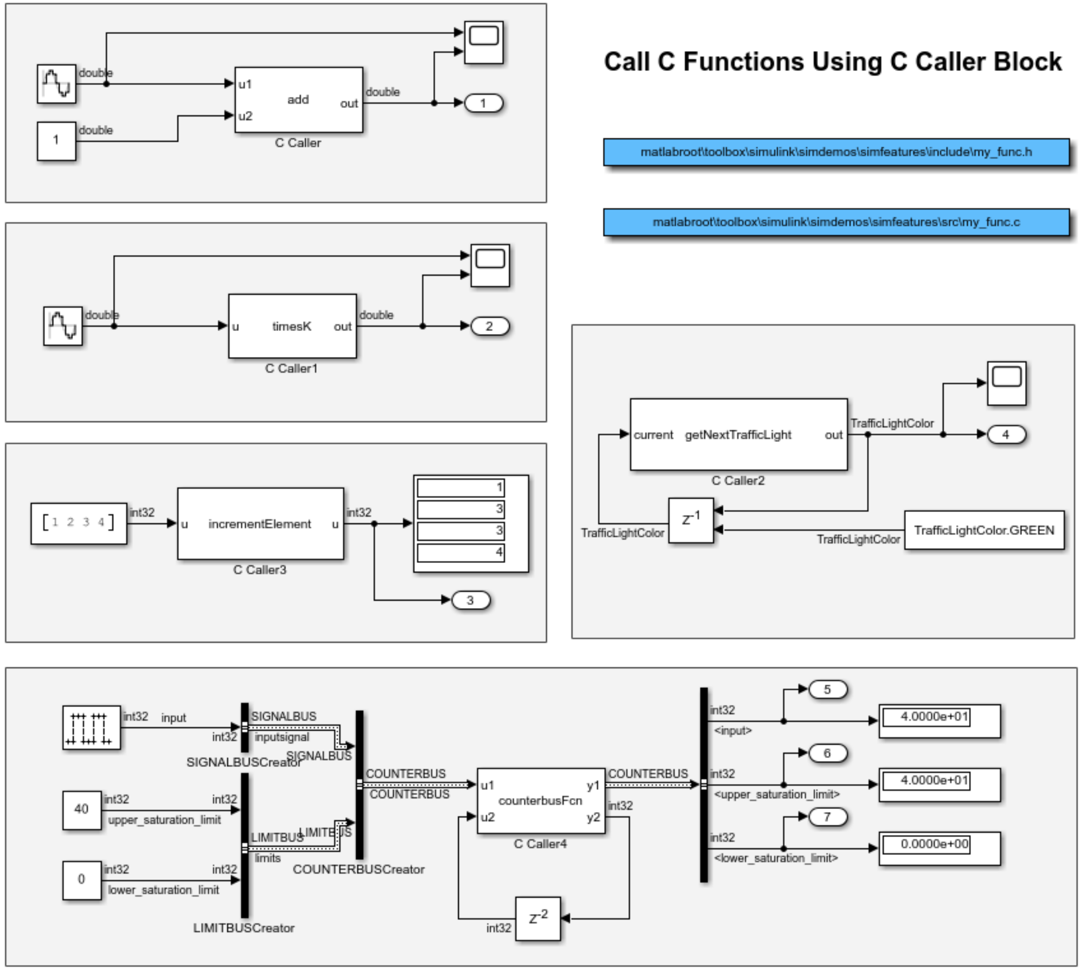


S-Function Builder

```
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

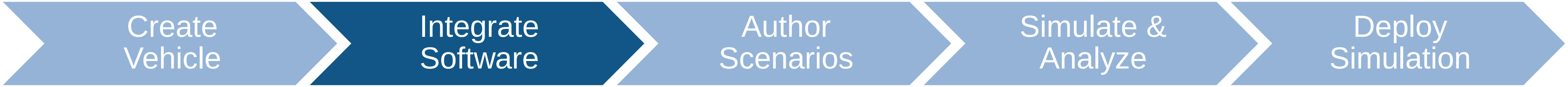
matlabroot\toolbox\simulink\simdemos\simfeatures\src\my_func.c

Learn more:

C / C++ code integration

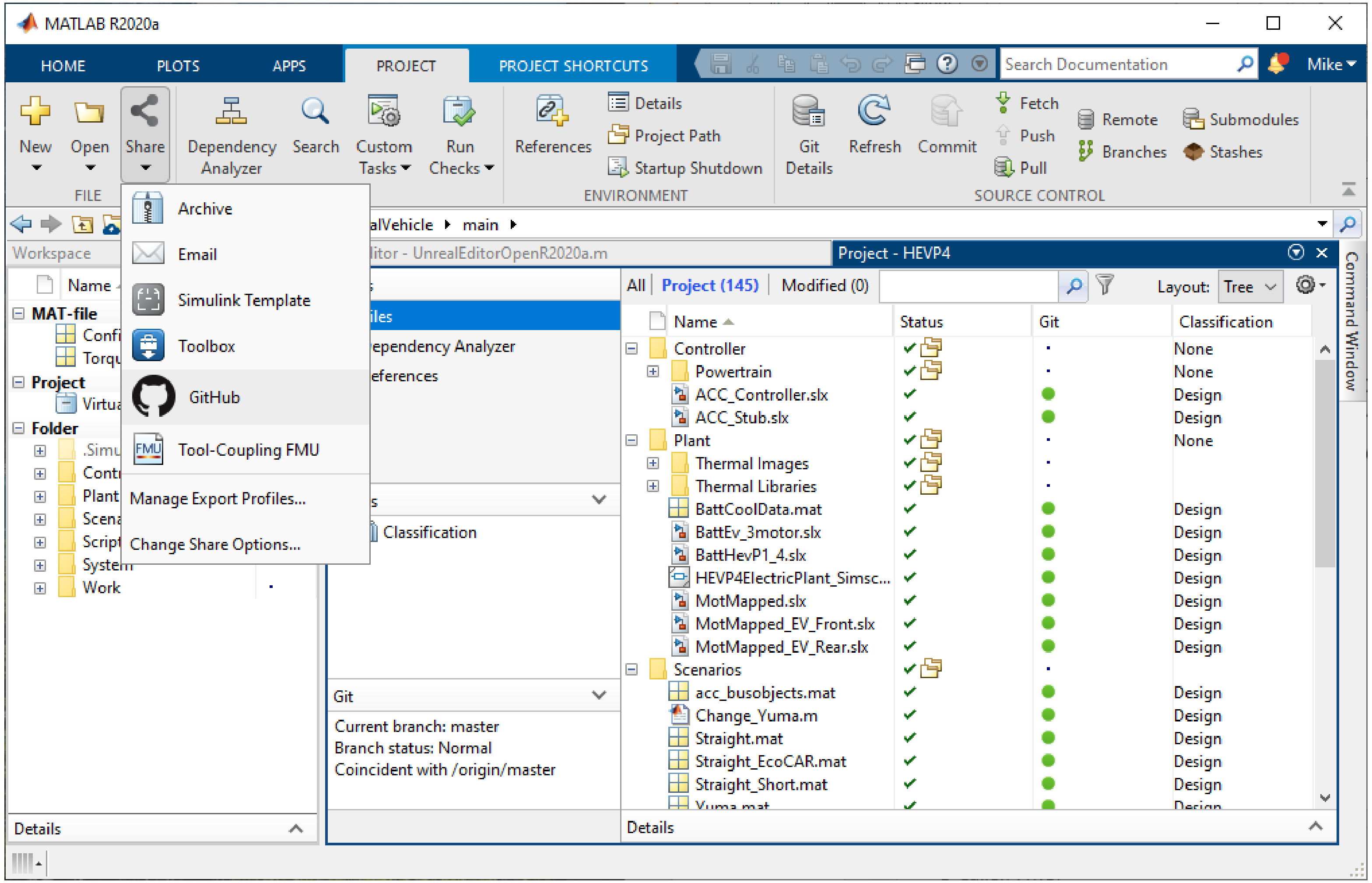
C Caller block

MathWorks Virtual Vehicle: complex project management



Use MathWorks platform to:

- Collaborate across teams
- Reference related project files
- Manage version control



Name	Status	Git	Classification
Controller	✓	·	None
Powertrain	✓	·	None
ACC_Controller.slx	✓	●	Design
ACC_Stub.slx	✓	●	Design
Plant	✓	·	None
Thermal Images	✓	·	
Thermal Libraries	✓	·	
BattCoolData.mat	✓	●	Design
BattEv_3motor.slx	✓	●	Design
BattHevP1_4.slx	✓	●	Design
HEVP4ElectricPlant_Simsc...	✓	●	Design
MotMapped.slx	✓	●	Design
MotMapped_EV_Front.slx	✓	●	Design
MotMapped_EV_Rear.slx	✓	●	Design
Scenarios	✓	·	
acc_busobjects.mat	✓	●	Design
Change_Yuma.m	✓	●	Design
Straight.mat	✓	●	Design
Straight_EcoCAR.mat	✓	●	Design
Straight_Short.mat	✓	●	Design
Yuma.mat	✓	●	Design

Current branch: master
Branch status: Normal
Coincident with /origin/master

Learn more:

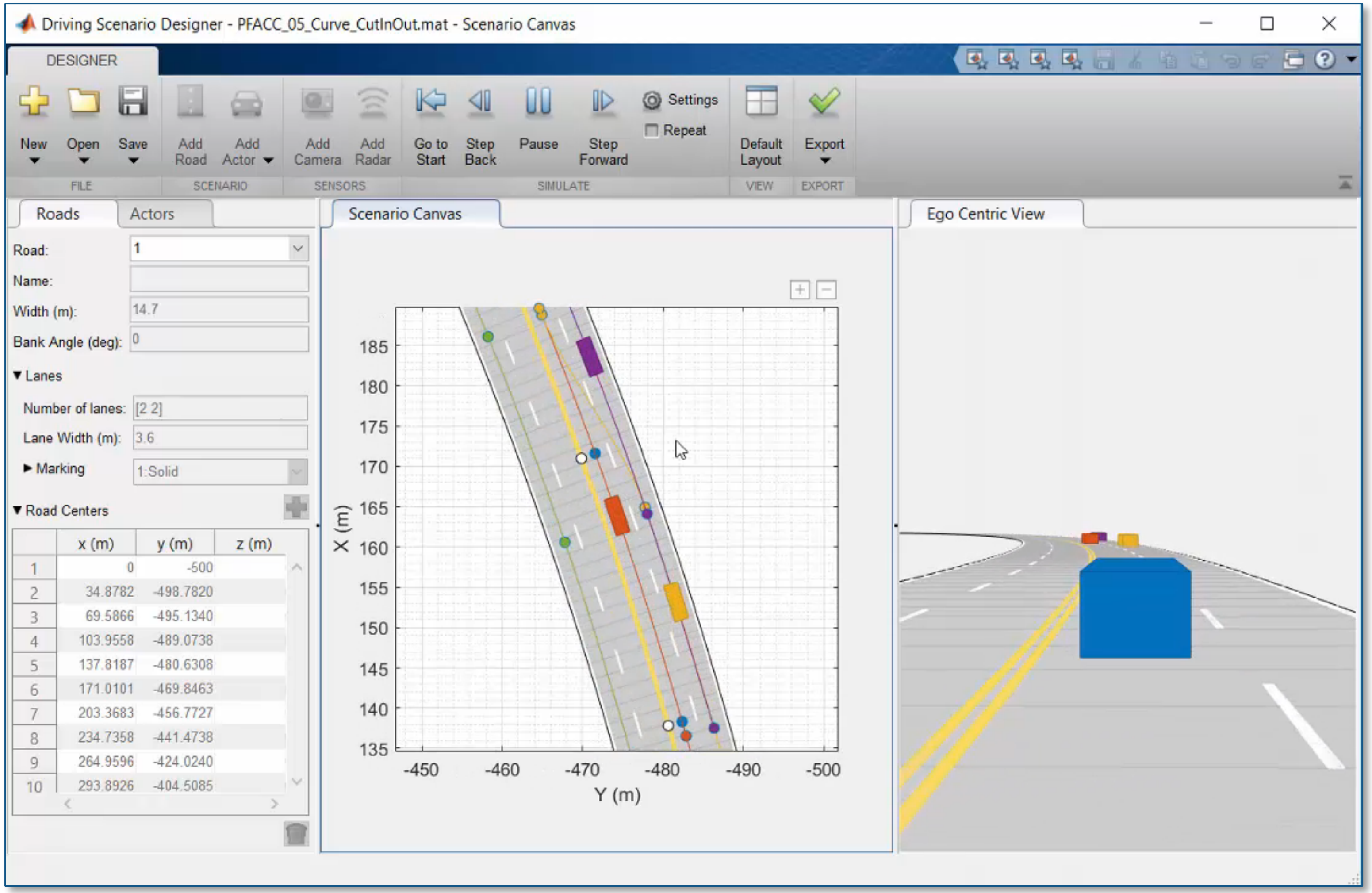
MATLAB Projects

MathWorks Virtual Vehicle: 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



Driving Scenario Designer - PFACC_05_Curve_CutInOut.mat - Scenario Canvas

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

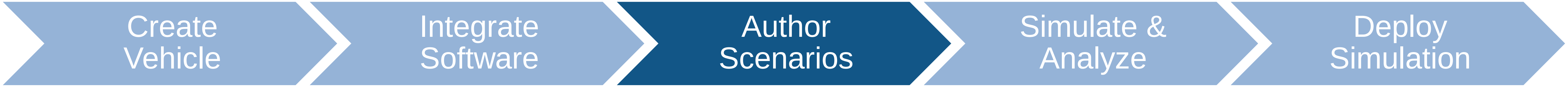
▼ Road Centers

	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:

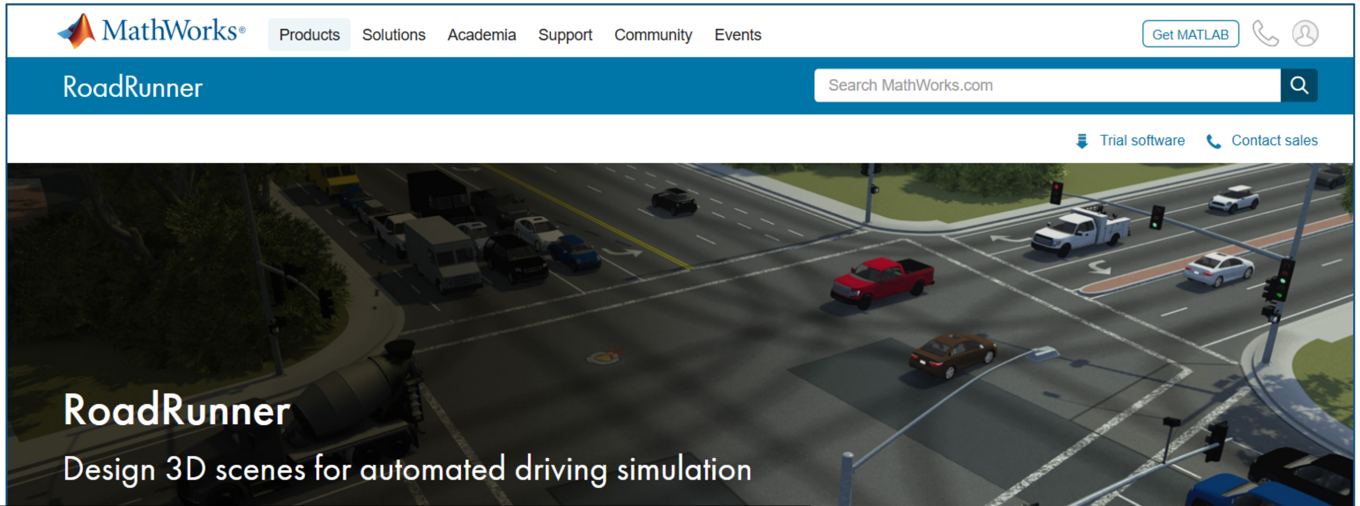
[Automated Driving Toolbox]

MathWorks Virtual Vehicle: automotive scene creation


Create Vehicle
Integrate Software
Author Scenarios
Simulate & Analyze
Deploy Simulation

Use RoadRunner to:

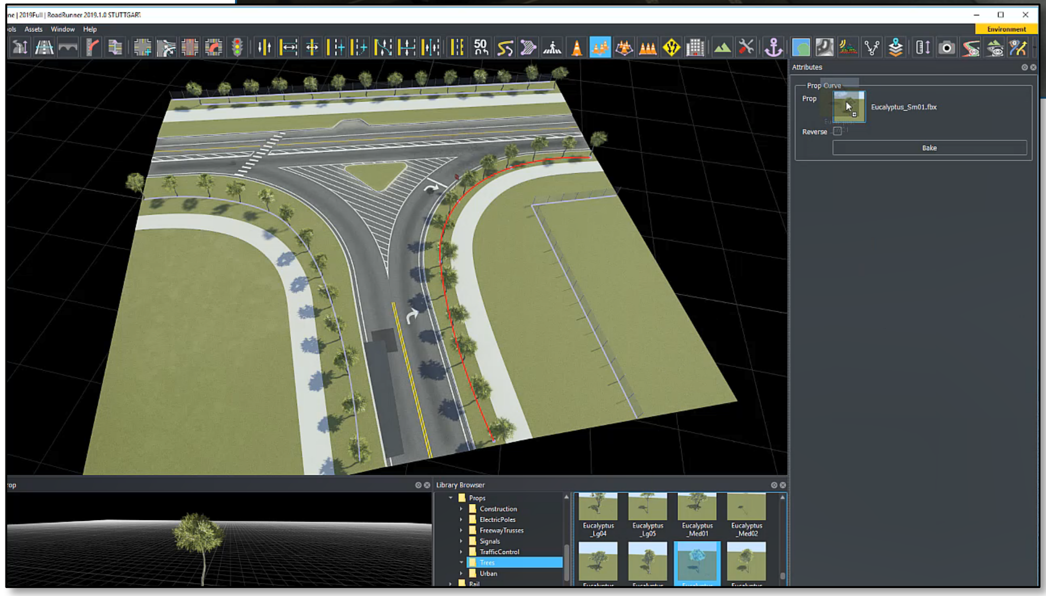
- Design 3D scenes for AD simulation
- Customize with region-specific road signs and markings
- Configure traffic signal timing
- Import from OpenDRIVE
- Export to OpenDRIVE, FBX, …
- Use scenes in Unreal, Unity, CARLA, …



RoadRunner

RoadRunner

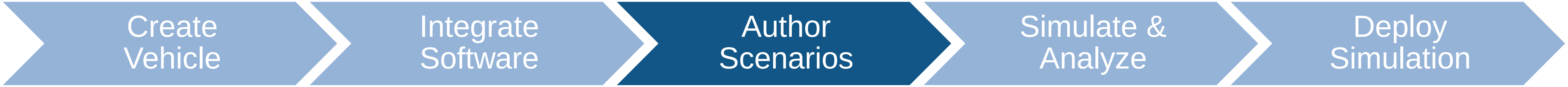
Design 3D scenes for automated driving simulation



Learn more:

[RoadRunner]

MathWorks Virtual Vehicle: requirements definition



Create Vehicle

Integrate Software

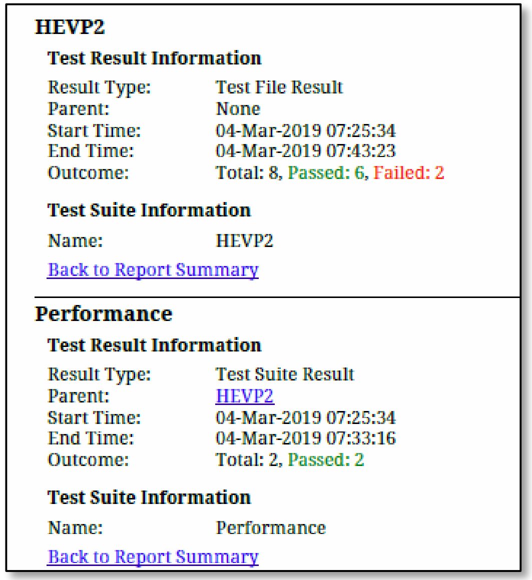
Author Scenarios

Simulate & Analyze

Deploy Simulation

Use V&V tools to:

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



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

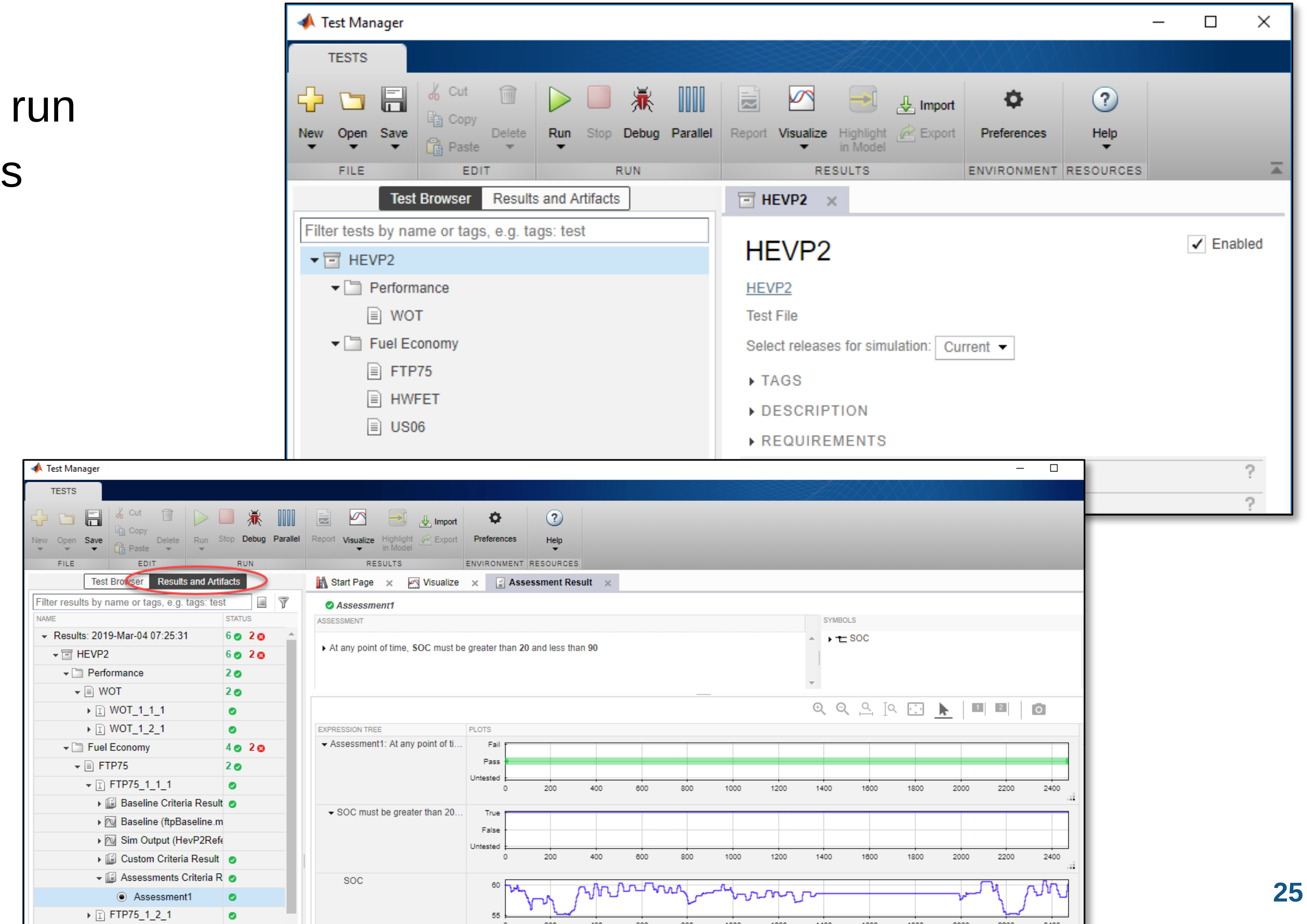
Test Result Information

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



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

- TAGS
- DESCRIPTION
- REQUIREMENTS

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

ASSESSMENT

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

SYMBOLS

- SOC

Learn more:

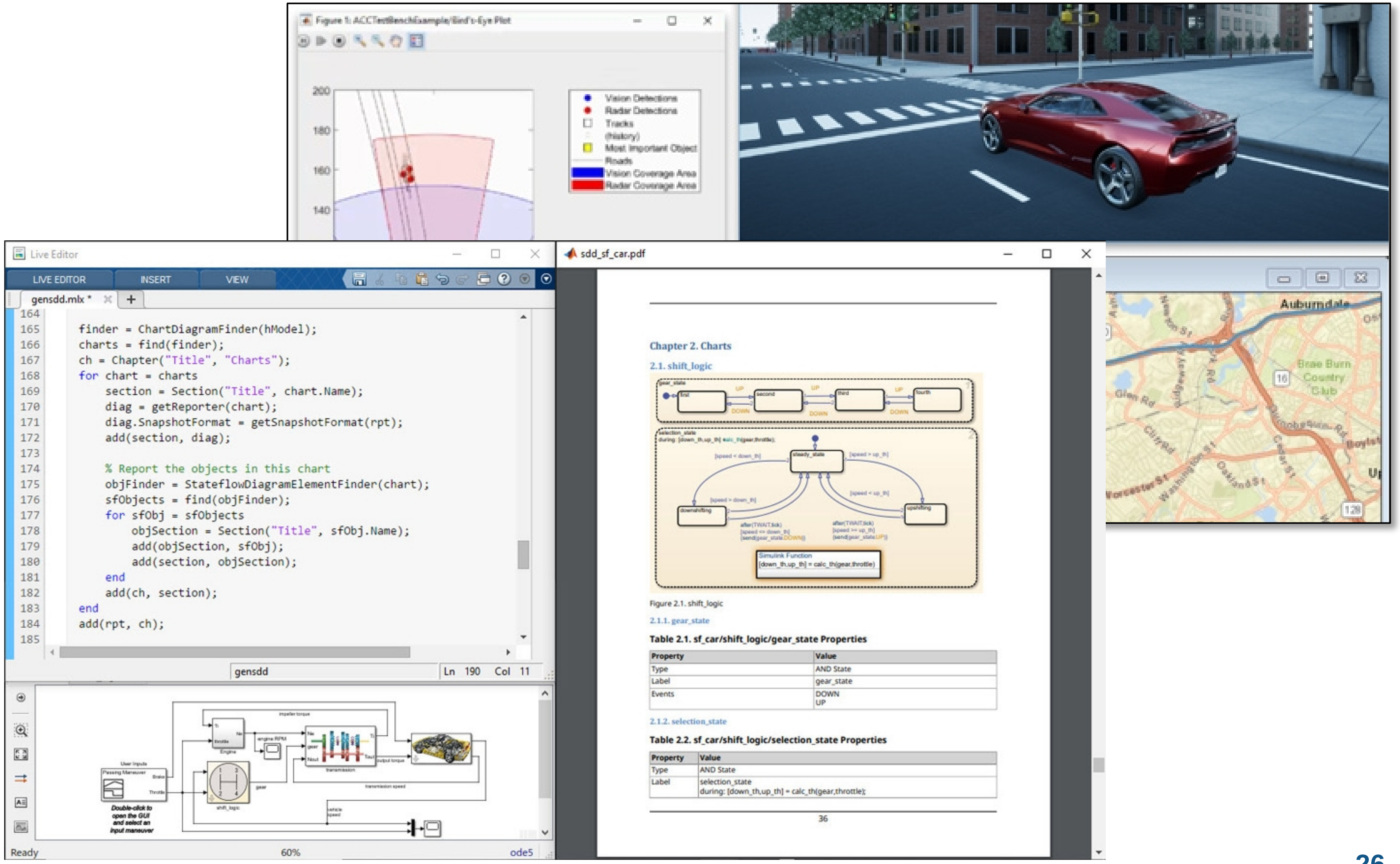
Verification & Validation

MathWorks Virtual Vehicle: 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);
```

gensdd Ln 190 Col 11

Ready 60% ode5

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

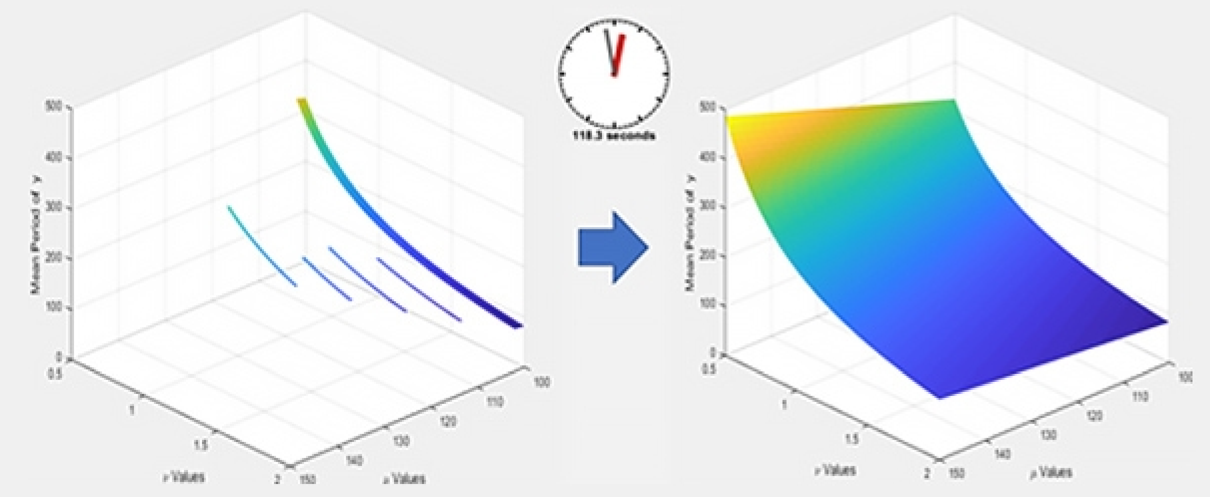
Learn more:

MATLAB Live Editor

Simulink Report Generator

MathWorks Virtual Vehicle: 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]

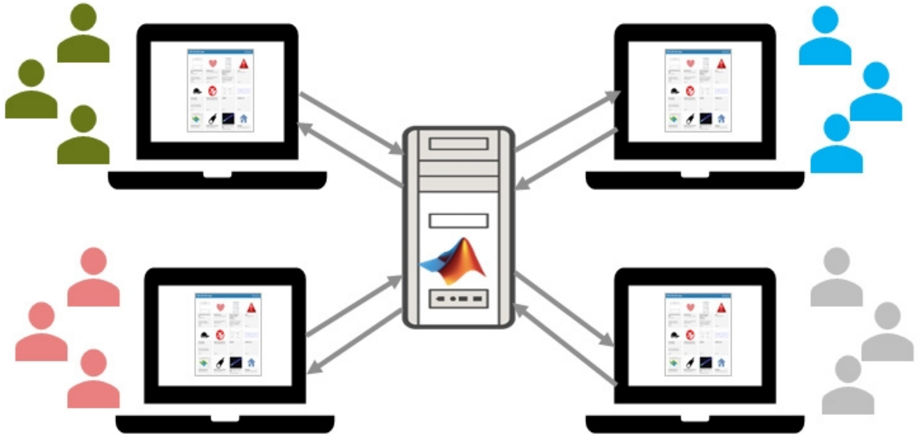
[MATLAB Parallel Server]

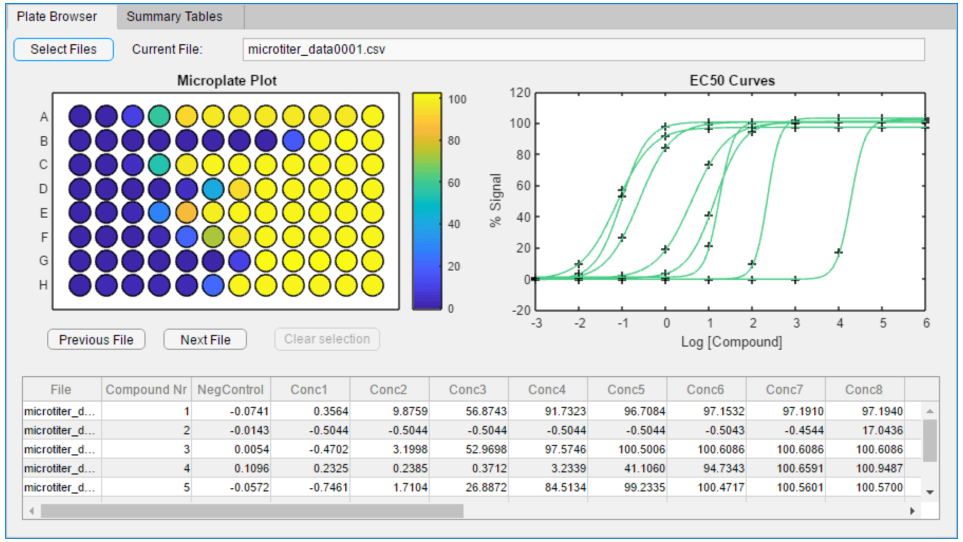
MathWorks Virtual Vehicle: model deployment

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

Use MATLAB and Simulink to take applications farther:

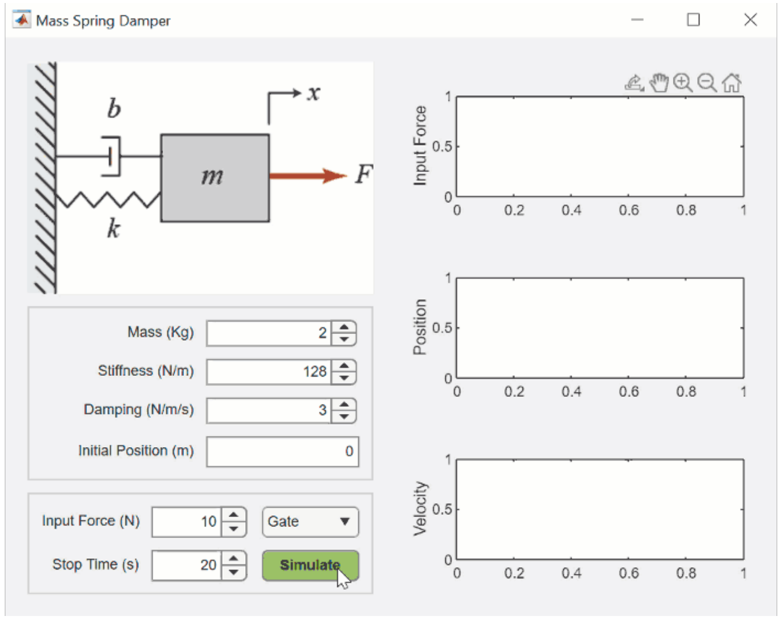
- Create custom UI's
- Create installers for distribution
- Deploy models as executables, FMU's or web apps
- Generate code for SIL, HIL testing





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]

MathWorks Consulting Services can support you

Model Architecture

Model assessment
Simulation performance
Interface standardization
…

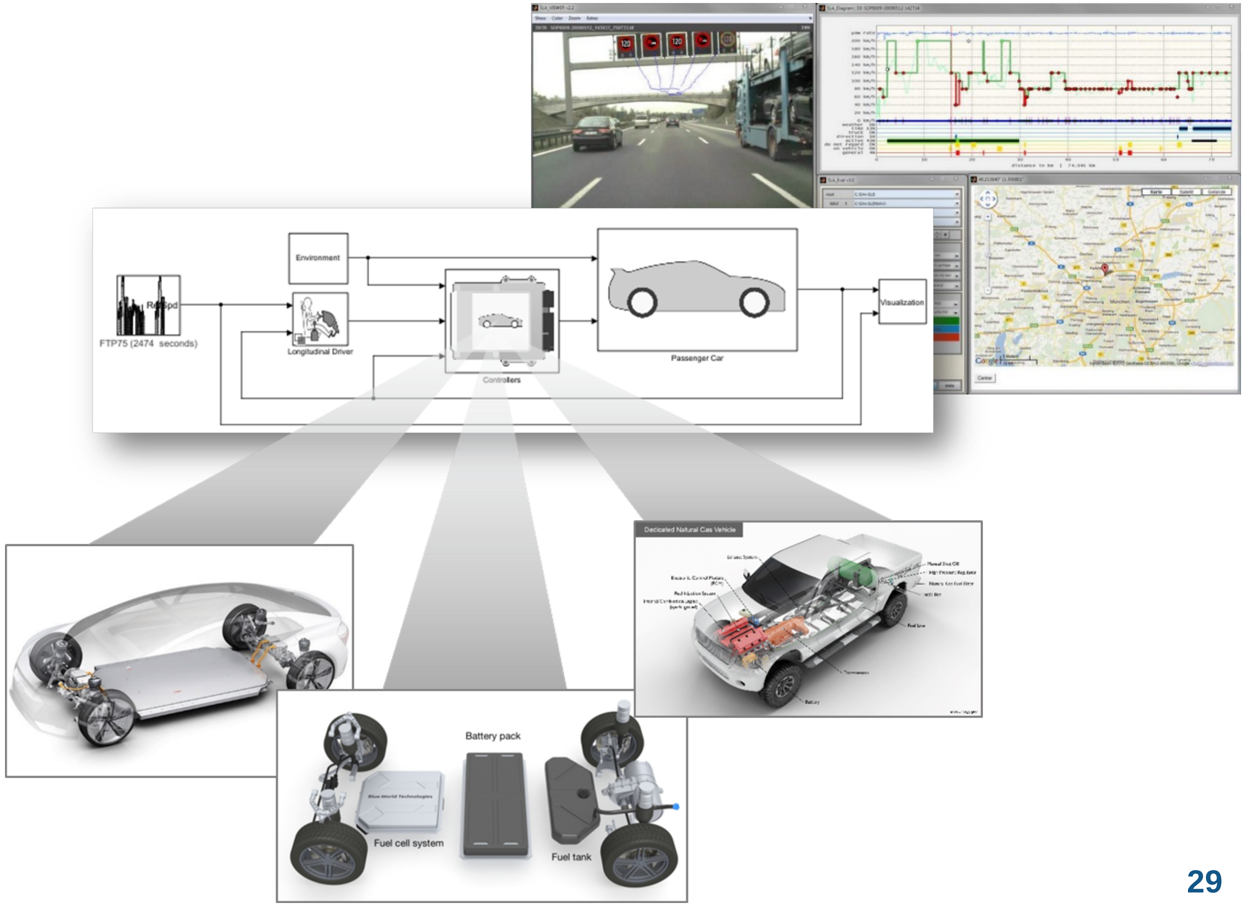
Construction

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



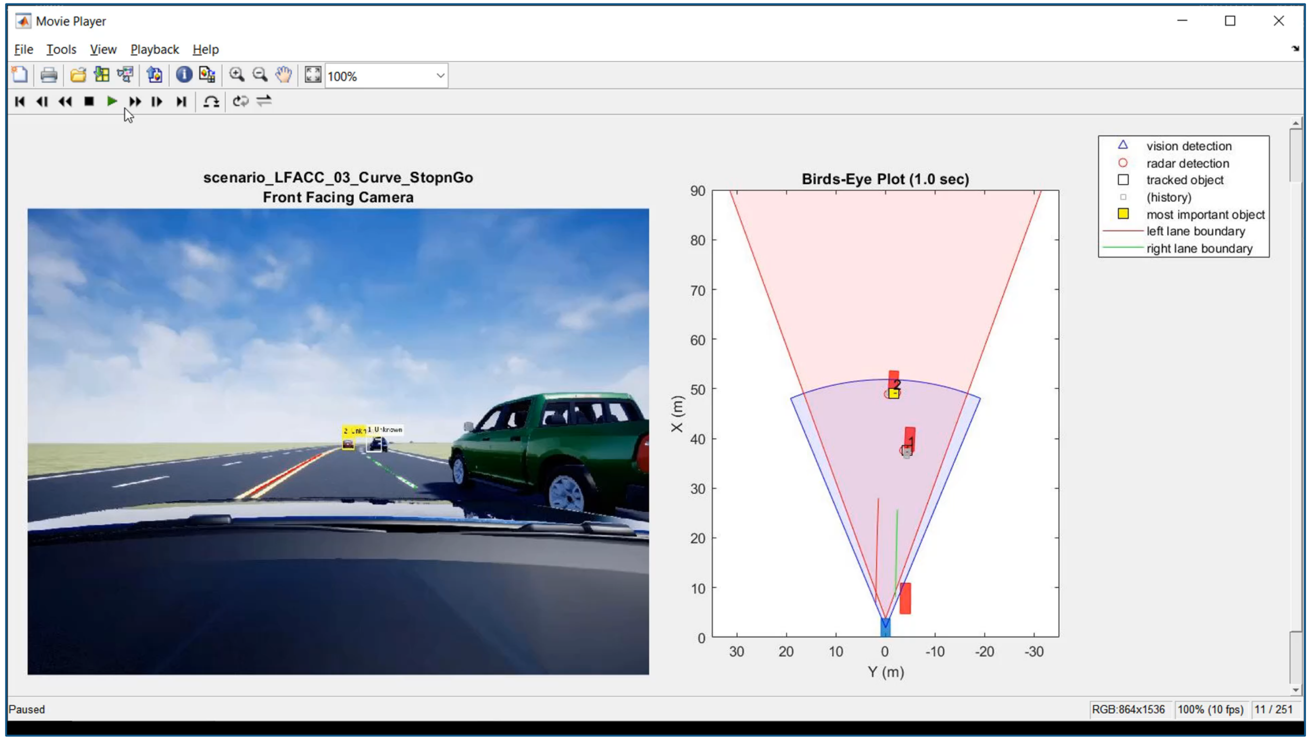
Agenda

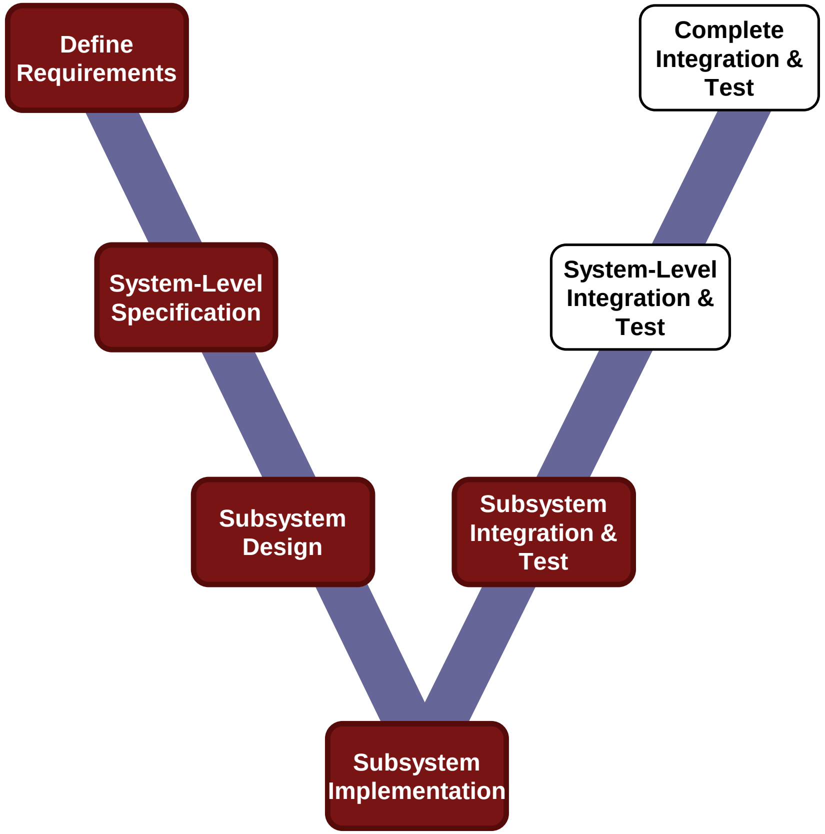
- Common challenges
- MathWorks solutions
- **Case study**

Validate software against function safety requirements early

☑ **FSR: The lane following system lateral error shall be less than 1 meter**

Use simulation to do system-level integration testing ***early***




Define Requirements
System-Level Specification
Subsystem Design
Subsystem Implementation
Subsystem Integration & Test
System-Level Integration & Test
Complete Integration & Test

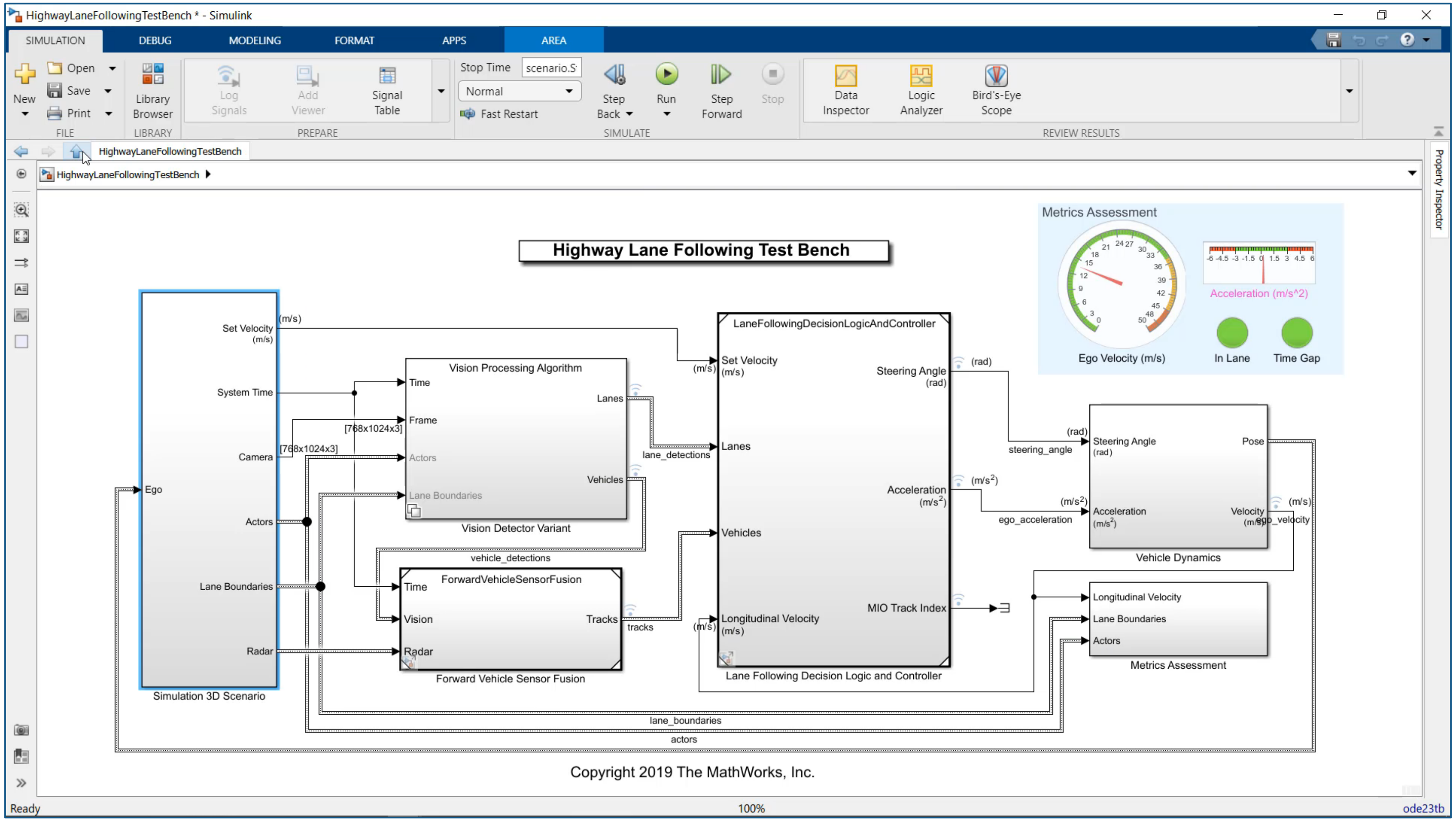
Learn more:

[Highway Lane Following]

[Automate Testing for Highway Lane Following]

Case study: highway lane following algorithm

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

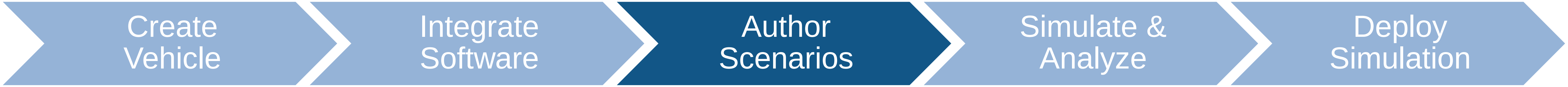


- Create Unreal Engine scene
- Specify target trajectories
- Model camera and radar sensors
- Model ego vehicle dynamics
- Specify system metrics

Learn more:

Highway Lane Following

Case study: highway lane following algorithm



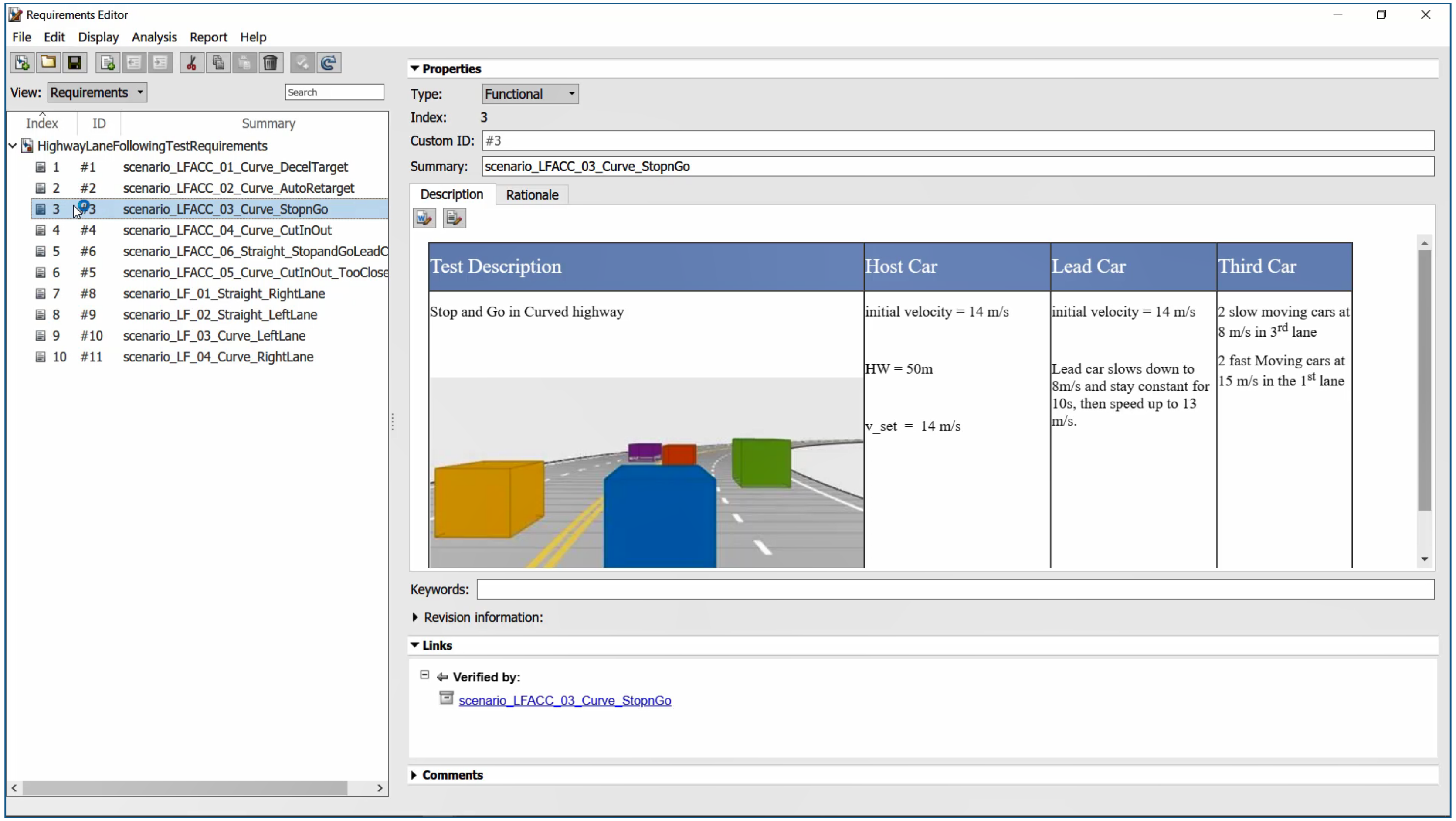
Create Vehicle

Integrate Software

Author Scenarios

Simulate & Analyze

Deploy Simulation



View: Requirements

Search

Index	ID	Summary
		HighwayLaneFollowingTestRequirements
1	#1	scenario_LFACC_01_Curve_DecelTarget
2	#2	scenario_LFACC_02_Curve_AutoRetarget
3	#3	scenario_LFACC_03_Curve_StopnGo
4	#4	scenario_LFACC_04_Curve_CutInOut
5	#6	scenario_LFACC_06_Straight_StopandGoLeadC
6	#5	scenario_LFACC_05_Curve_CutInOut_TooClose
7	#8	scenario_LF_01_Straight_RightLane
8	#9	scenario_LF_02_Straight_LeftLane
9	#10	scenario_LF_03_Curve_LeftLane
10	#11	scenario_LF_04_Curve_RightLane

Properties

Type: Functional

Index: 3

Custom ID: #3

Summary: scenario_LFACC_03_Curve_StopnGo

Description | Rationale

Test Description	Host Car	Lead Car	Third Car

Stop and Go in Curved highway	initial velocity = 14 m/s
	HW = 50m
	v_set = 14 m/s
		initial velocity = 14 m/s
		Lead car slows down to 8m/s and stay constant for 10s, then speed up to 13 m/s.
			2 slow moving cars at 8 m/s in 3rd lane
			2 fast Moving cars at 15 m/s in the 1st lane

Keywords:

Revision information:

Links

Verified by:

scenario_LFACC_03_Curve_StopnGo

Comments

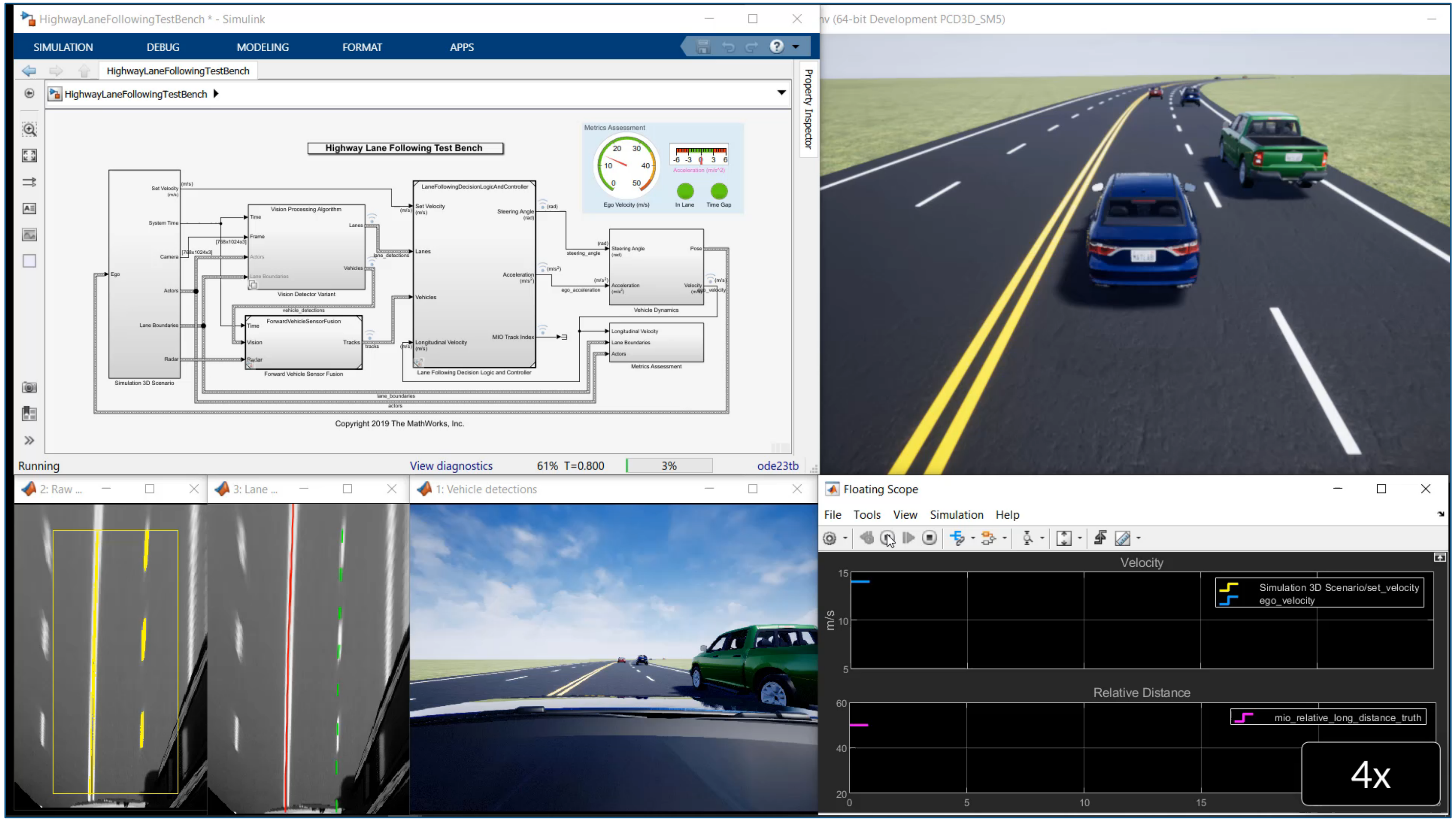
- Author and associate requirements and scenarios

Learn more:

[Automate Testing for Highway Lane Following]()

Case study: highway lane following algorithm

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



- Visualize system behavior with Unreal Engine
- Visualize lane detections
- Visualize vehicle detections
- Visualize control signals
- Log simulation data

Learn more:

Highway Lane Following

Case study: highway lane following algorithm

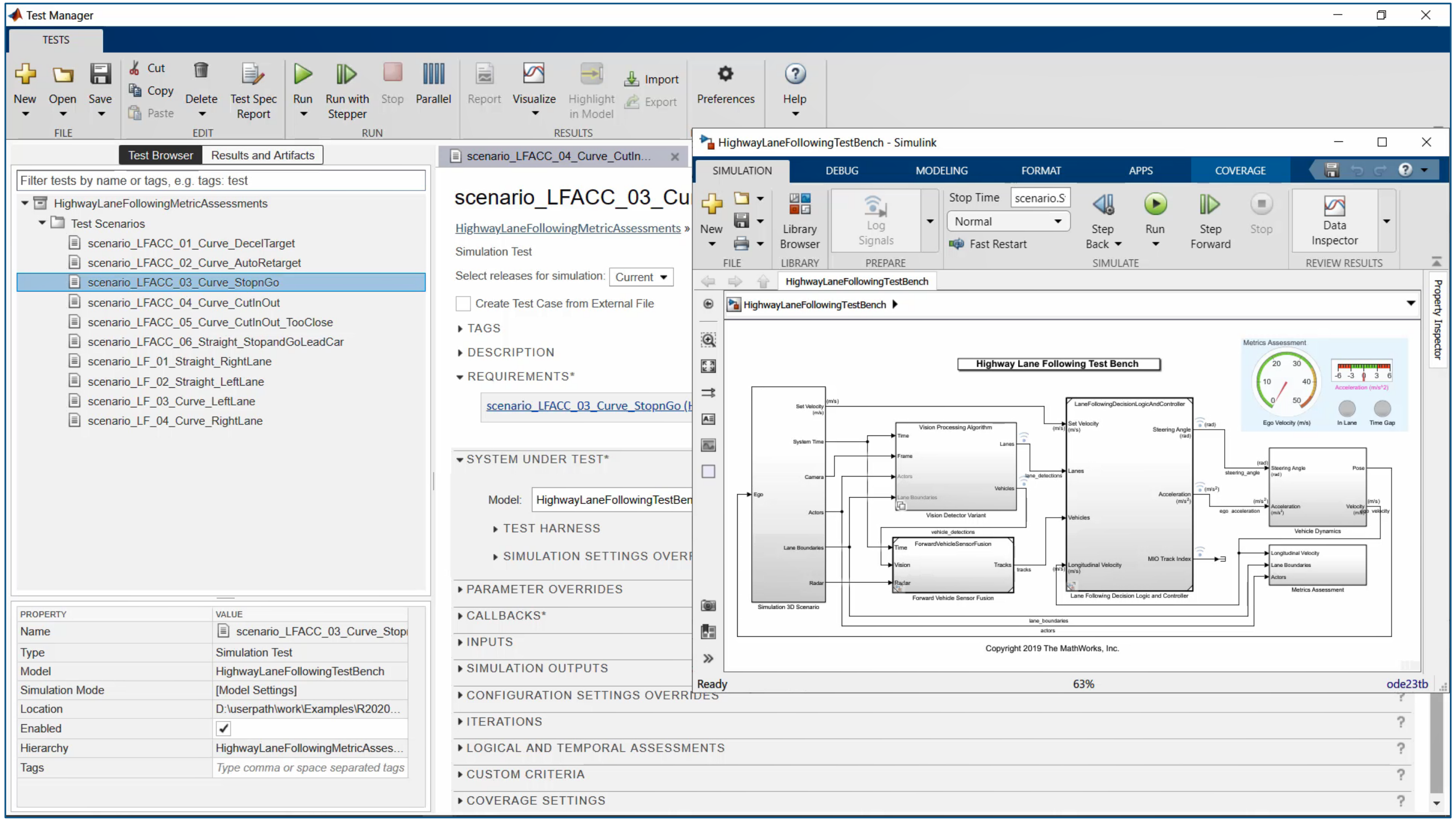
Create Vehicle

Integrate Software

Author Scenarios

Simulate & Analyze

Deploy Simulation



- HighwayLaneFollowingMetricAssessments
  - Test Scenarios
    - scenario_LFACC_01_Curve_DecelTarget
    - scenario_LFACC_02_Curve_AutoRetarget
    - scenario_LFACC_03_Curve_StopnGo
    - scenario_LFACC_04_Curve_CutInOut
    - scenario_LFACC_05_Curve_CutInOut_TooClose
    - scenario_LFACC_06_Straight_StopandGoLeadCar
    - scenario_LF_01_Straight_RightLane
    - scenario_LF_02_Straight_LeftLane
    - scenario_LF_03_Curve_LeftLane
    - scenario_LF_04_Curve_RightLane

PROPERTY	VALUE
Name	scenario_LFACC_03_Curve_Stop
Type	Simulation Test
Model	HighwayLaneFollowingTestBench
Simulation Mode	[Model Settings]
Location	D:\userpath\work\Examples\R2020...
Enabled	✓
Hierarchy	HighwayLaneFollowingMetricAsses...
Tags	Type comma or space separated tags

scenario_LFACC_03_Cu

HighwayLaneFollowingMetricAssessments »

Simulation Test

Select releases for simulation: Current

□ Create Test Case from External File

- TAGS
- DESCRIPTION
- REQUIREMENTS*
  - scenario_LFACC_03_Curve_StopnGo (H
- SYSTEM UNDER TEST*
  - Model: HighwayLaneFollowingTestBen
  - TEST HARNESS
  - SIMULATION SETTINGS OVERR
- PARAMETER OVERRIDES
- CALLBACKS*
- INPUTS
- SIMULATION OUTPUTS
- CONFIGURATION SETTINGS OVERRIDES
- ITERATIONS
- LOGICAL AND TEMPORAL ASSESSMENTS
- CUSTOM CRITERIA
- COVERAGE SETTINGS

Stop Time: scenario.S

Normal

- Automate test execution and reporting
- Execute simulations in parallel

Learn more:

Automate Testing for Highway Lane Following

Case study: highway lane following algorithm

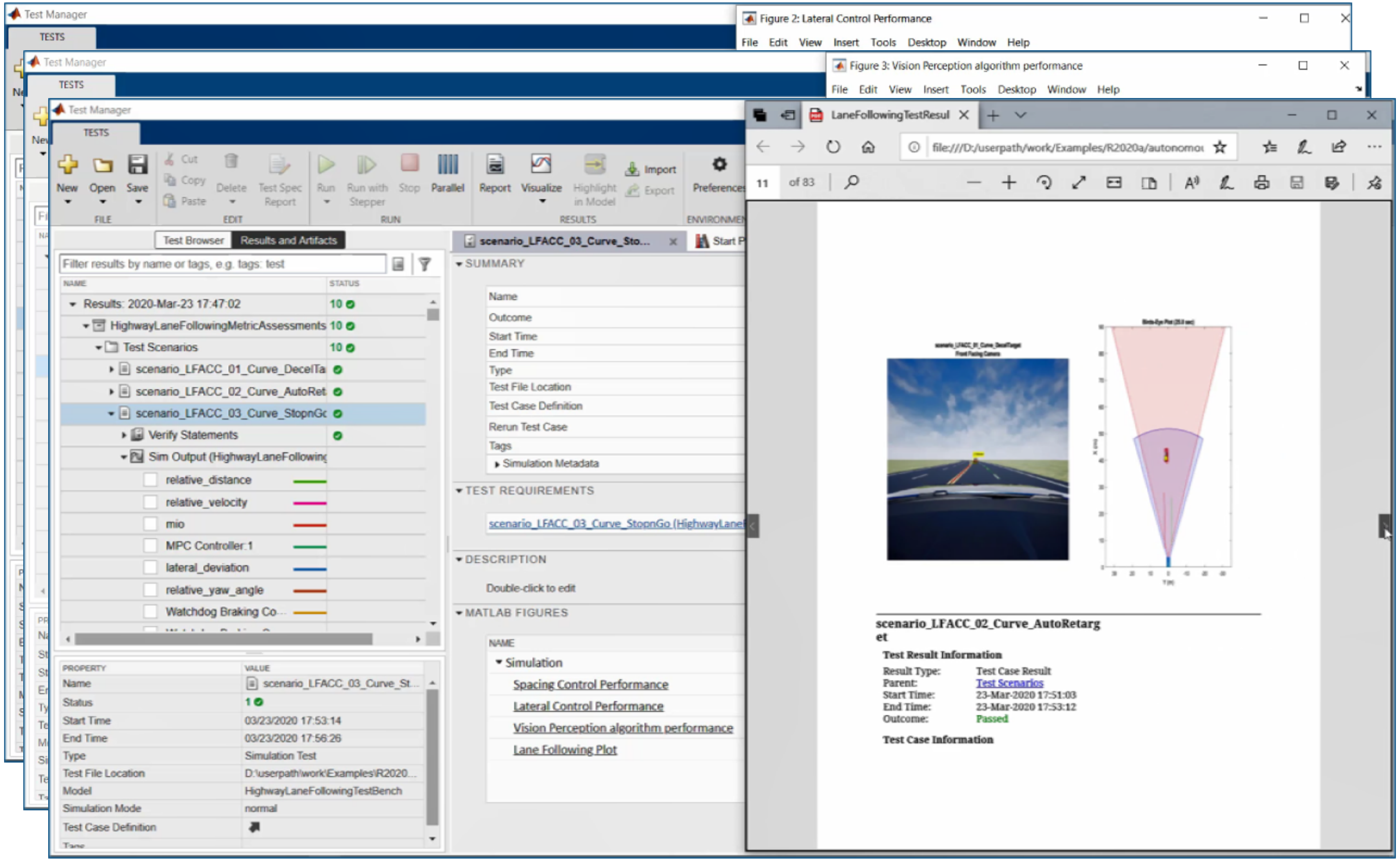
Create Vehicle

Integrate Software

Author Scenarios

Simulate & Analyze

Deploy Simulation



Test Manager
TESTS
Figure 2: Lateral Control Performance
File Edit View Insert Tools Desktop Window Help
Figure 3: Vision Perception algorithm performance
File Edit View Insert Tools Desktop Window Help
LaneFollowingTestResul
file:///D:/userpath/work/Examples/R2020a/autonomou
11 of 83
New Open Save
Cut Copy Paste Delete Test Spec Report
Run Run with Stepper Stop Parallel
Report Visualize Highlight in Model Import Export
Preferences
FILE EDIT RUN RESULTS ENVIRONMENT
Test Browser
Results and Artifacts
Filter results by name or tags, e.g. tags: test
scenario_LFACC_03_Curve_Sto...
Start P

NAME	STATUS
Results: 2020-Mar-23 17:47:02	10 ✓
HighwayLaneFollowingMetricAssessments	10 ✓
Test Scenarios	10 ✓
scenario_LFACC_01_Curve_DecelTa	✓
scenario_LFACC_02_Curve_AutoRet	✓
scenario_LFACC_03_Curve_StopnGc	✓
Verify Statements	✓
Sim Output (HighwayLaneFollowing	
□ relative_distance	
□ relative_velocity	
□ mio	
□ MPC Controller:1	
□ lateral_deviation	
□ relative_yaw_angle	
□ Watchdog Braking Co...	

PROPERTY	VALUE
Name	scenario_LFACC_03_Curve_St...
Status	1 ✓
Start Time	03/23/2020 17:53:14
End Time	03/23/2020 17:56:26
Type	Simulation Test
Test File Location	D:\userpath\work\Examples\R2020...
Model	HighwayLaneFollowingTestBench
Simulation Mode	normal
Test Case Definition	

SUMMARY

Name

Outcome

Start Time

End Time

Type

Test File Location

Test Case Definition

Rerun Test Case

Tags

Simulation Metadata

TEST REQUIREMENTS

scenario_LFACC_03_Curve_StopnGo (HighwayLane

DESCRIPTION

Double-click to edit

MATLAB FIGURES

NAME

- Simulation
  - Spacing Control Performance
  - Lateral Control Performance
  - Vision Perception algorithm performance
  - Lane Following Plot

scenario_LFACC_02_Curve_AutoRetarget

Test Result Information

Result Type:	Test Case Result
Parent:	Test Scenarios
Start Time:	23-Mar-2020 17:51:03
End Time:	23-Mar-2020 17:53:12
Outcome:	Passed

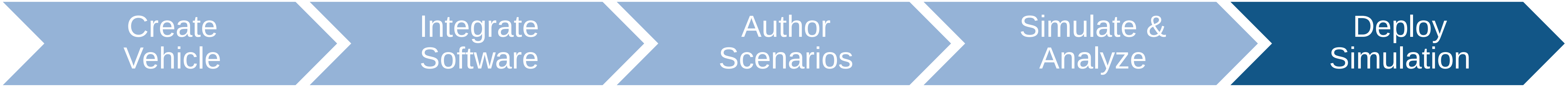
Test Case Information

- Assess system metrics
- Assess lane detection metrics

Learn more:

[Automate Testing for Highway Lane Following]()

Case study: highway lane following algorithm

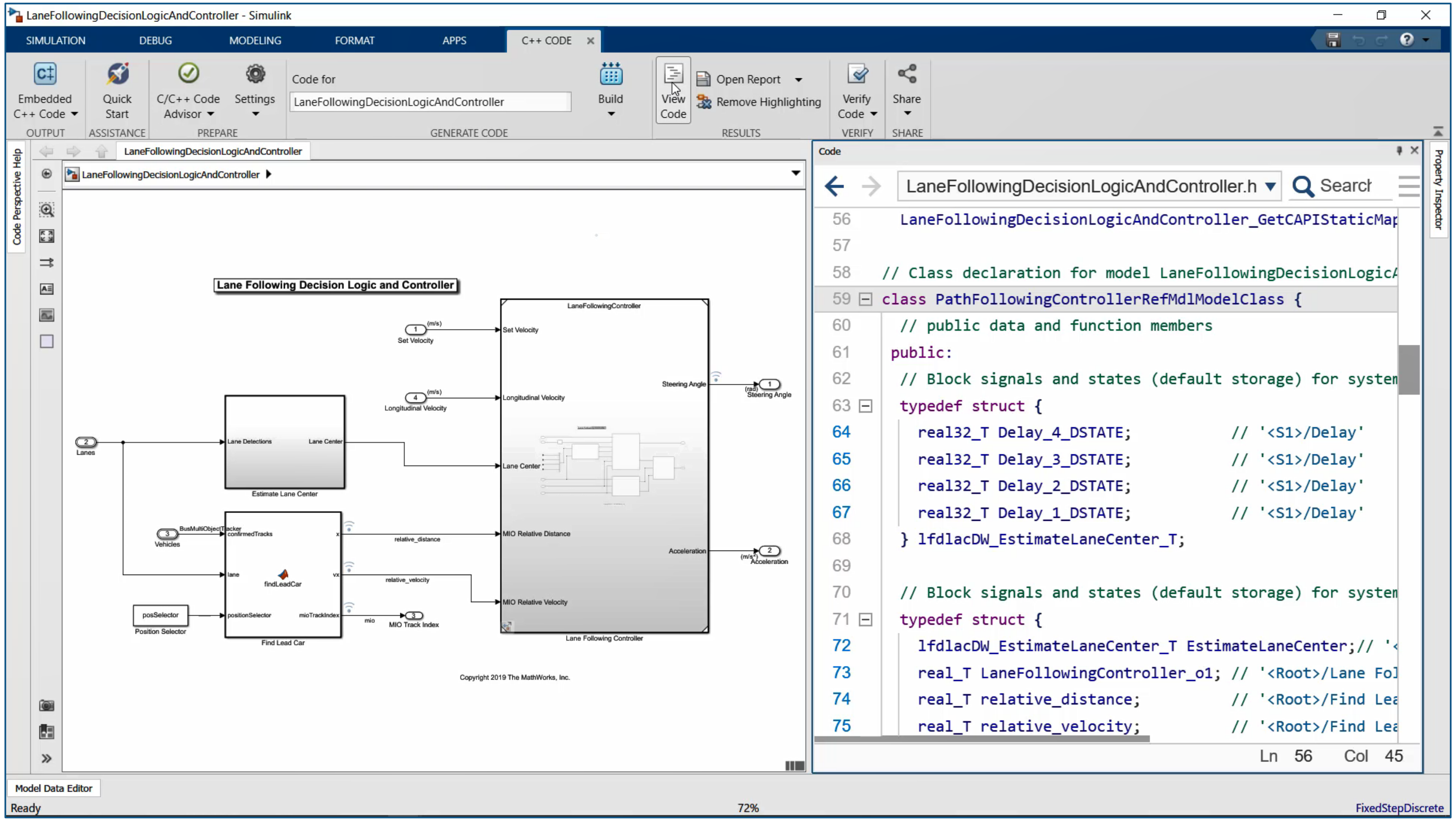

Create Vehicle

Integrate Software

Author Scenarios

Simulate & Analyze

Deploy Simulation



```
  LaneFollowingDecisionLogicAndController_GetCAPIStaticMap

// Class declaration for model LaneFollowingDecisionLogicA
class PathFollowingControllerRefMdlModelClass {
  // public data and function members
 public:
  // Block signals and states (default storage) for systen
  typedef struct {
    real32_T Delay_4_DSTATE;           // '<S1>/Delay'
    real32_T Delay_3_DSTATE;           // '<S1>/Delay'
    real32_T Delay_2_DSTATE;           // '<S1>/Delay'
    real32_T Delay_1_DSTATE;           // '<S1>/Delay'
  } lfdlacDW_EstimateLaneCenter_T;

  // Block signals and states (default storage) for systen
  typedef struct {
    lfdlacDW_EstimateLaneCenter_T EstimateLaneCenter;// '<
    real_T LaneFollowingController_o1; // '<Root>/Lane Fol
    real_T relative_distance;          // '<Root>/Find Lea
    real_T relative_velocity;          // '<Root>/Find Lea
```

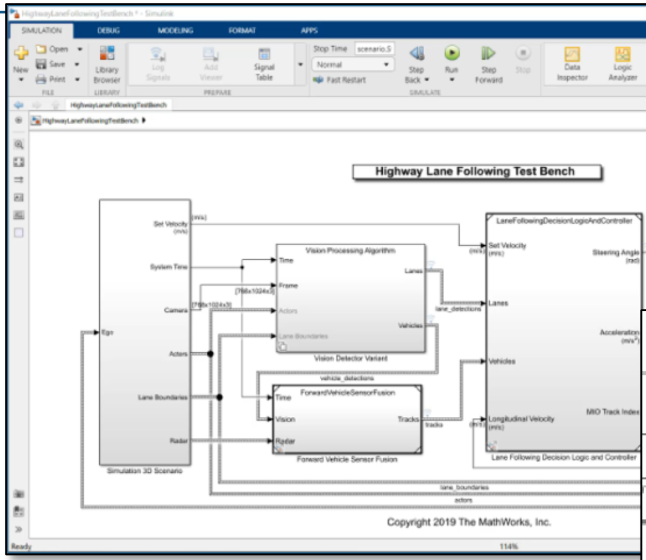
- Generate algorithm code
- Test with Software-in-the-Loop (SIL) simulation
- Workflow could be extended to test hand coded algorithms

Learn more:

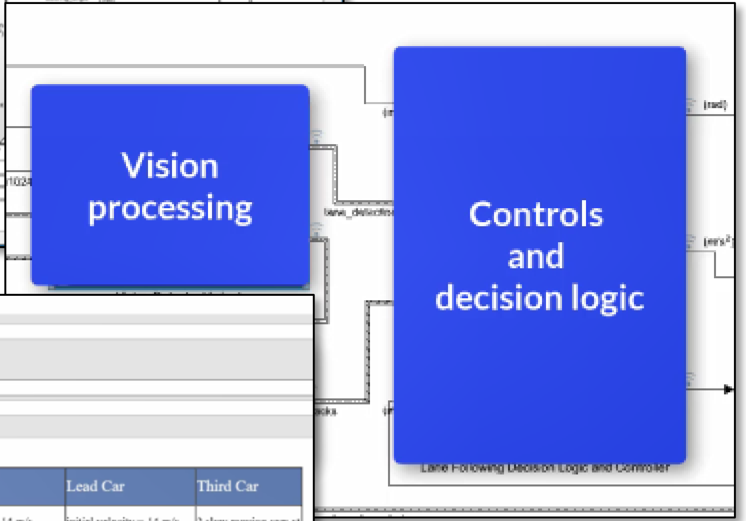
Automate Testing for Highway Lane Following

Summary

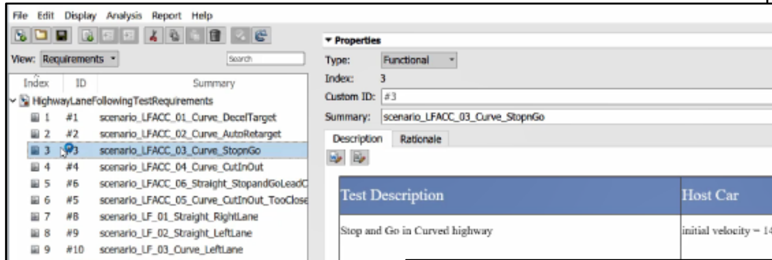
1. Started with reference application, then customized
2. Integrated software
3. Defined scenarios to test
4. Simulated model and analyzed results
5. Deployed model



Highway Lane Following Test Bench
Metrics Assessment
Vision Processing Algorithm
Vision Detector Variant
Forward Vehicle Sensor Fusion
LaneFollowingDecisionLogicAndController
Lane Following Decision Logic and Controller
Simulation 3D Scenario
Copyright 2019 The MathWorks, Inc.



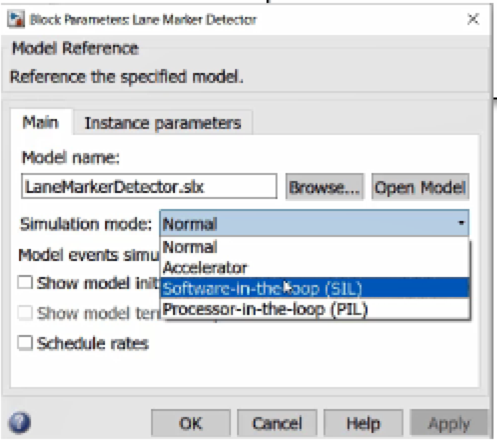
Vision processing
Controls and decision logic
Lane Following Decision Logic and Controller



Index	ID	Summary
		HighwayLaneFollowingTestRequirements
1	#1	scenario_LFACC_01_Curve_DecelTarget
2	#2	scenario_LFACC_02_Curve_AutoRetarget
3	#3	scenario_LFACC_03_Curve_StopnGo
4	#4	scenario_LFACC_04_Curve_CutInOut
5	#6	scenario_LFACC_06_Straight_StopandGoLeadC
6	#5	scenario_LFACC_05_Curve_CutInOut_TooClose
7	#8	scenario_LF_01_Straight_RightLane
8	#9	scenario_LF_02_Straight_LeftLane
9	#10	scenario_LF_03_Curve_LeftLane
10	#11	scenario_LF_04_Curve_RightLane

Type: Functional
Index: 3
Custom ID: #3
Summary: scenario_LFACC_03_Curve_StopnGo

Test Description	Host Car	Lead Car	Third Car
Stop and Go in Curved highway	initial velocity = 14 m/s	initial velocity = 14 m/s	2 slow moving cars at 8 m/s in 3rd lane



Block Parameters: Lane Marker Detector

Model Reference

Reference the specified model.

Model name: LaneMarkerDetector.slx

Simulation mode: Normal

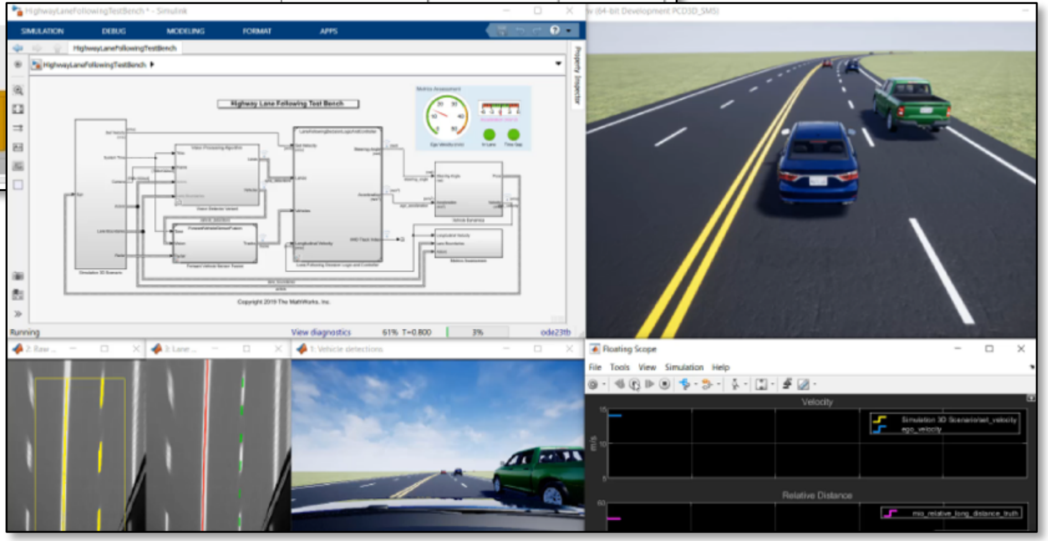
- Normal
- Accelerator
- Software-in-the-loop (SIL)
- Processor-in-the-loop (PIL)

Model events simu

□ Show model init

□ Show model ter

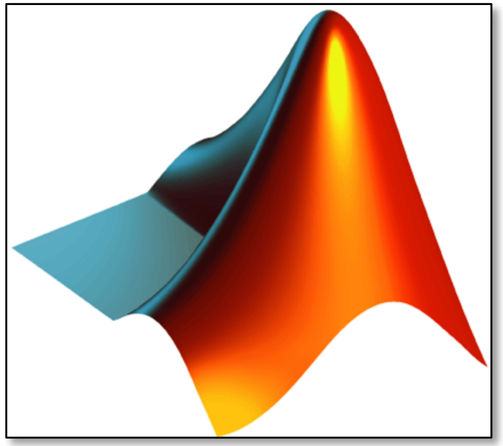
□ Schedule rates

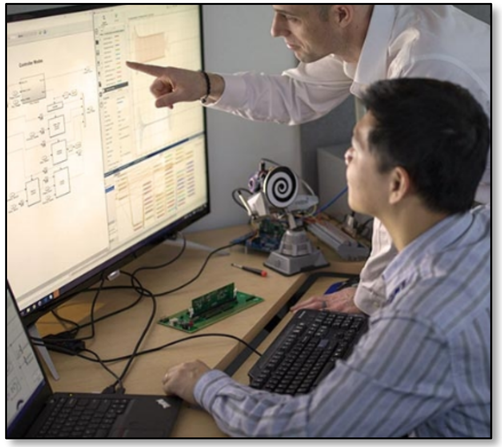


Key takeaways

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

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

○ 1 (easy)

○ 2 (moderate)

○ 3 (difficult)

○ 4 (major challenge)

Please contact us with questions

Eva Pelster
epelster@mathworks.com

MathWorks®