MathWorks Simulation Challenge Year 3

Mark Corless and Lauren Tabolinsky
October 10, 2020

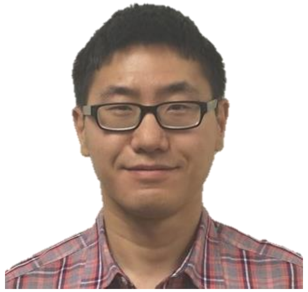
Agenda

- Meet the MathWorks Team
- MathWorks offerings for teams
- Simulation Challenge
- New Products
- Questions

Team Mentors

- Seo-Wook
  - U Waterloo
  - U Toronto
  - spark@mathworks.com

- Shusen
  - Texas AM
  - Virginia Tech
  - szhang@mathworks.com



- Kunal
  - Michigan Tech
  - North Carolina AT
  - kpatil@mathworks.com

- Kim
  - Kettering U
  - Michigan State
  - kmcgarri@mathworks.com

The MathWorks AutoDrive Team

Christoph Hahn

- Technical Lead
- Christoph.Hahn@mathworks.de

Swarooph Seshadri

- Technical Lead
- sseshadr@mathworks.com

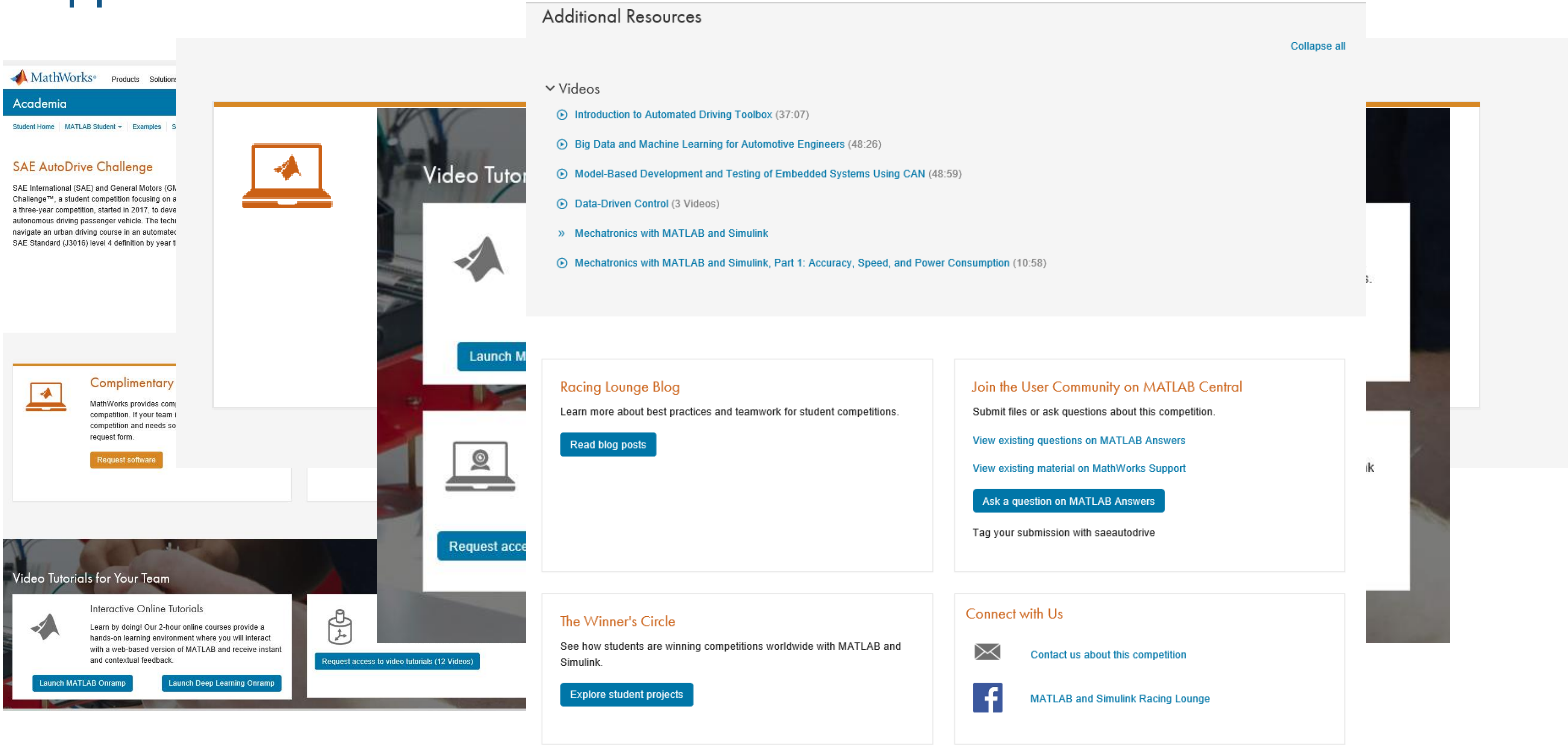
Lauren Tabolinsky

- Academic Programs Manager
- ltabolin@mathworks.com

Mark Corless

- ADAS / AD Industry Marketing Manager
- mcorless@mathworks.com

Support for Teams



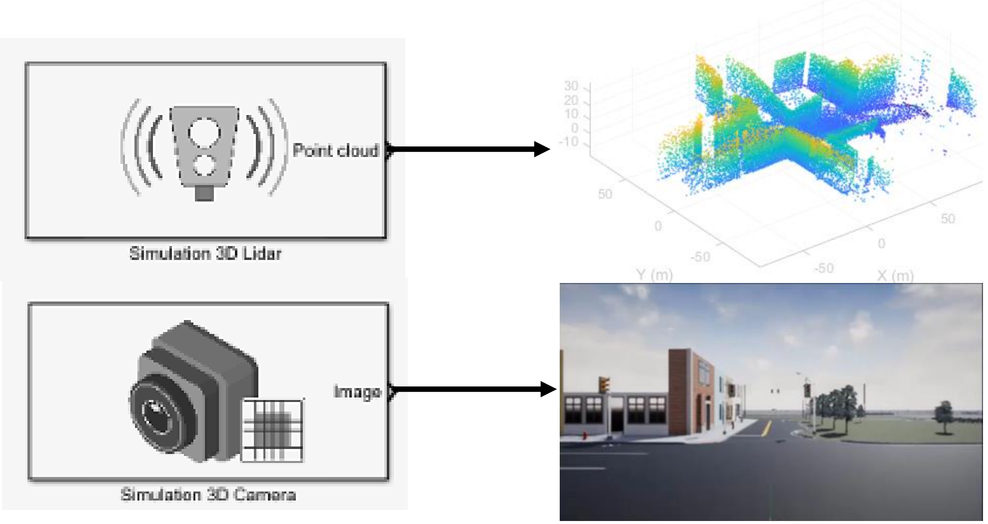
[AutoDrive Webpage]

November Workshop

- MathWorks Workshop (SAE International HQ; Warrendale, PA)
- Dates: Nov. 14-15, 2019
- Agenda is being finalized and registration will open up next week

Simulation Challenge

- Last year's challenge was great!
  - 1st customers to see sensors for Unreal engine
  - Highlights
    - **U Toronto**: Best integration between real-world car development and simulation challenge
    - **Michigan Tech**: Best use of lidar sensor, IPCV and ROS for object detection
    - **Texas A+M**: IPCV and deep learning using MW tools
- We decided to make the challenge more open this year
  - Points (50 available) will be awarded
  - How has your team applied the core concepts using MathWorks' tools to help achieve the overall competition objectives?



MathWorks Simulation Challenge

for SAE AutoDrive year 3 competition

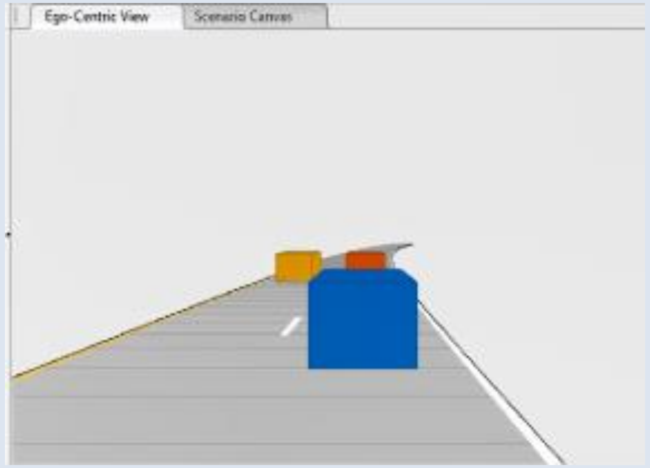
#	Task	Points
1	Synthesize data to test open loop perception algorithm	10
2	Synthesize data to test closed loop controls algorithm	10
3	Generate code from controls algorithm	10
4	Innovate	15
5	Reflect	5

MathWorks Simulation Challenge

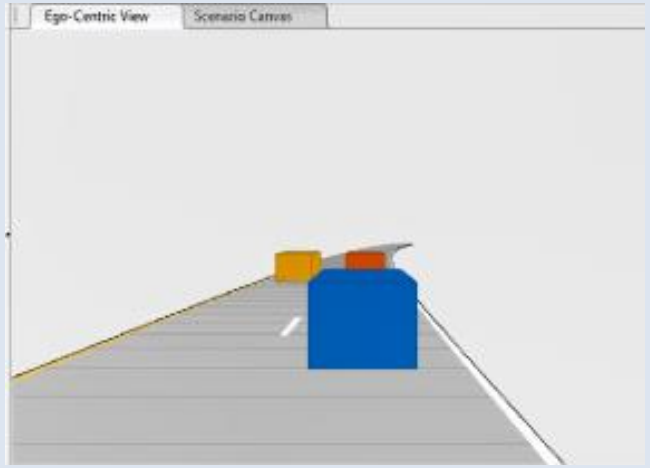
for SAE AutoDrive year 3 competition

#	Task	Points
1	**Synthesize data to test open loop perception algorithm** a. What algorithm did you test and why did you choose it? (i.e. object detection, drivable path, localization, sensor fusion) b. How did you synthesize scenario data? (i.e. Unreal Engine, Driving Scenario Designer, sensor models, customizations) c. How did you assess correctness of the algorithm? (i.e. specify truth, assess metrics, automate testing)	10
2	Synthesize data to test closed loop controls algorithm	10
3	Generate code from controls algorithm	10
4	Innovate	15
5	Reflect	5

How can I design with virtual driving scenarios?

Scenes	**Cuboid** 
Testing	Controls, sensor fusion, planning
Authoring	Driving Scenario Designer App Programmatic API (drivingScenario)
Sensing	Probabilistic radar (detection list) Probabilistic vision (detection list) Probabilistic lane (detection list)

How can I design with virtual driving scenarios?

Scenes	**Cuboid** 	**3D Simulation (Unreal Engine)** 
Testing	Controls, sensor fusion, planning	Controls, sensor fusion, planning, perception
Authoring	Driving Scenario Designer App Programmatic API (drivingScenario)	Unreal Engine Editor
Sensing	Probabilistic radar (detection list) Probabilistic vision (detection list) Probabilistic lane (detection list)	Probabilistic radar (detection list) Monocular camera (image, labels, depth) Fisheye camera (image) Lidar (point cloud)

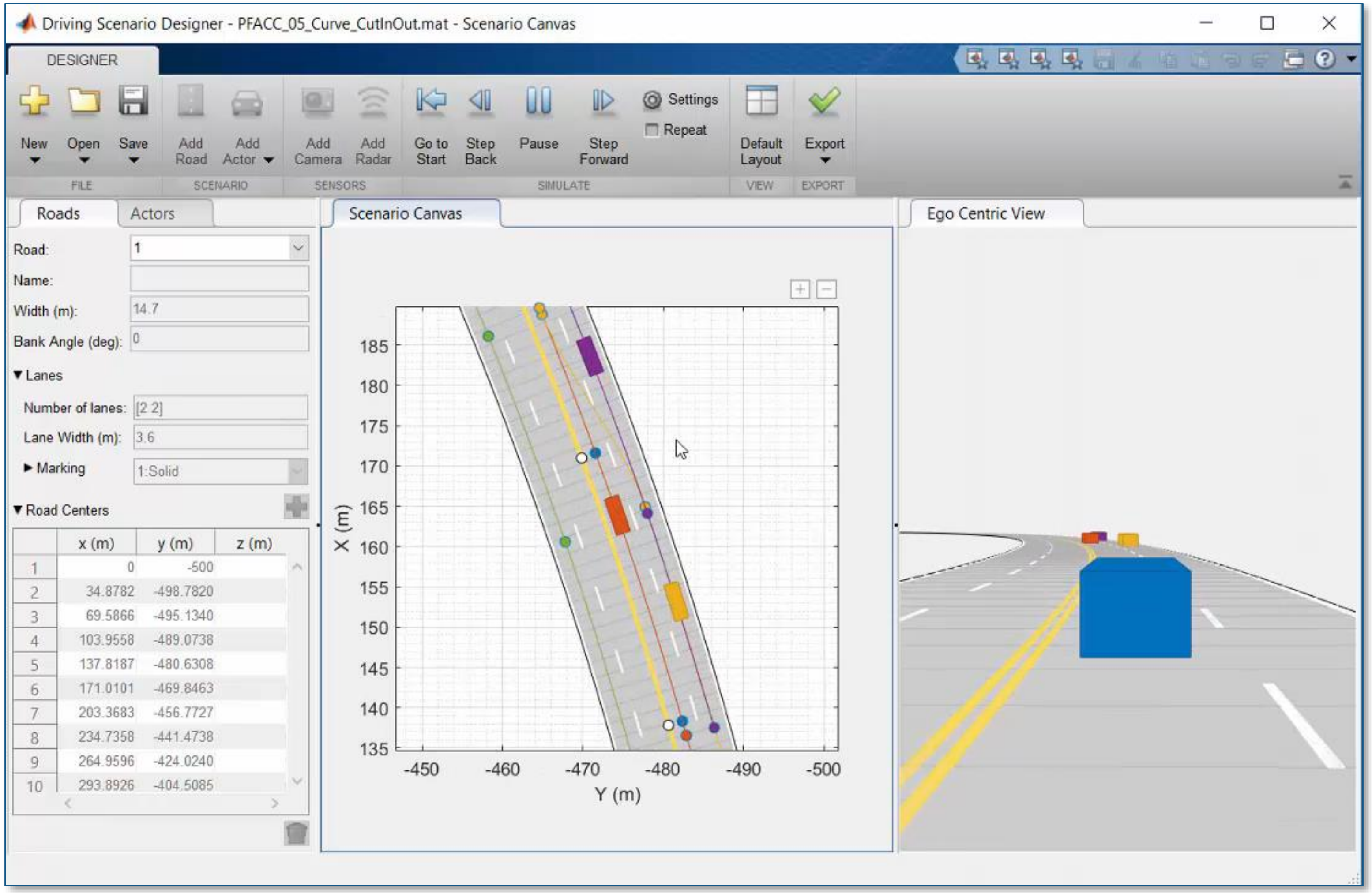
Graphically author driving scenarios

[Driving Scenario Designer](#)

- Create roads and lane markings
- Add actors and trajectories
- Specify actor size and radar cross-section (RCS)
- Explore pre-built scenarios
- Import OpenDRIVE roads

Automated Driving Toolbox™

R2018a



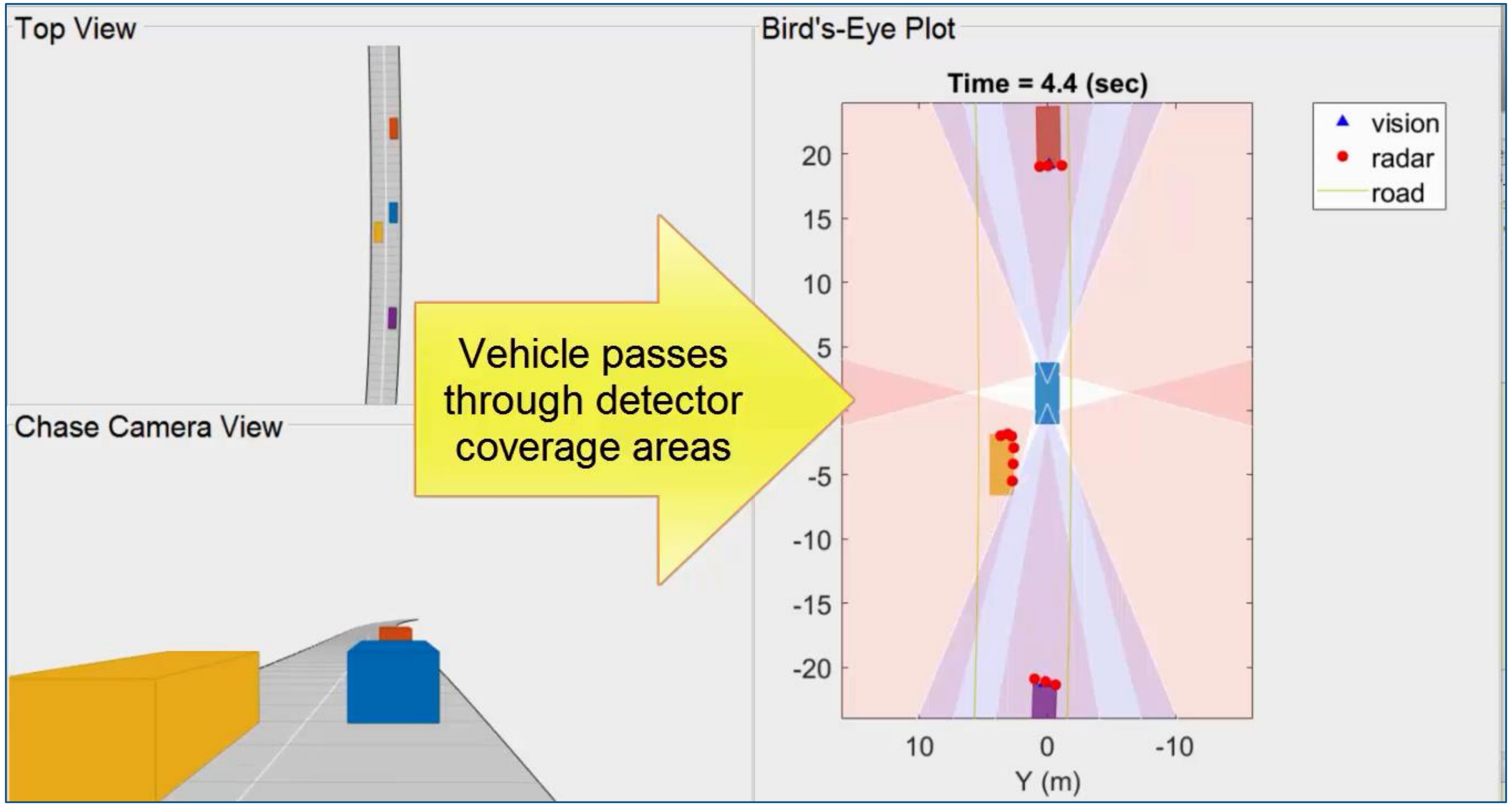
Synthesize scenarios to test sensor fusion algorithms

[Sensor Fusion Using Synthetic Radar and Vision Data]

- Synthesize road and vehicles
- Add probabilistic vision and radar detection sensors
- Fuse and track detections
- Visualize sensor coverage areas, detections, and tracks

Automated Driving Toolbox™

R2017a



Design multi-object trackers

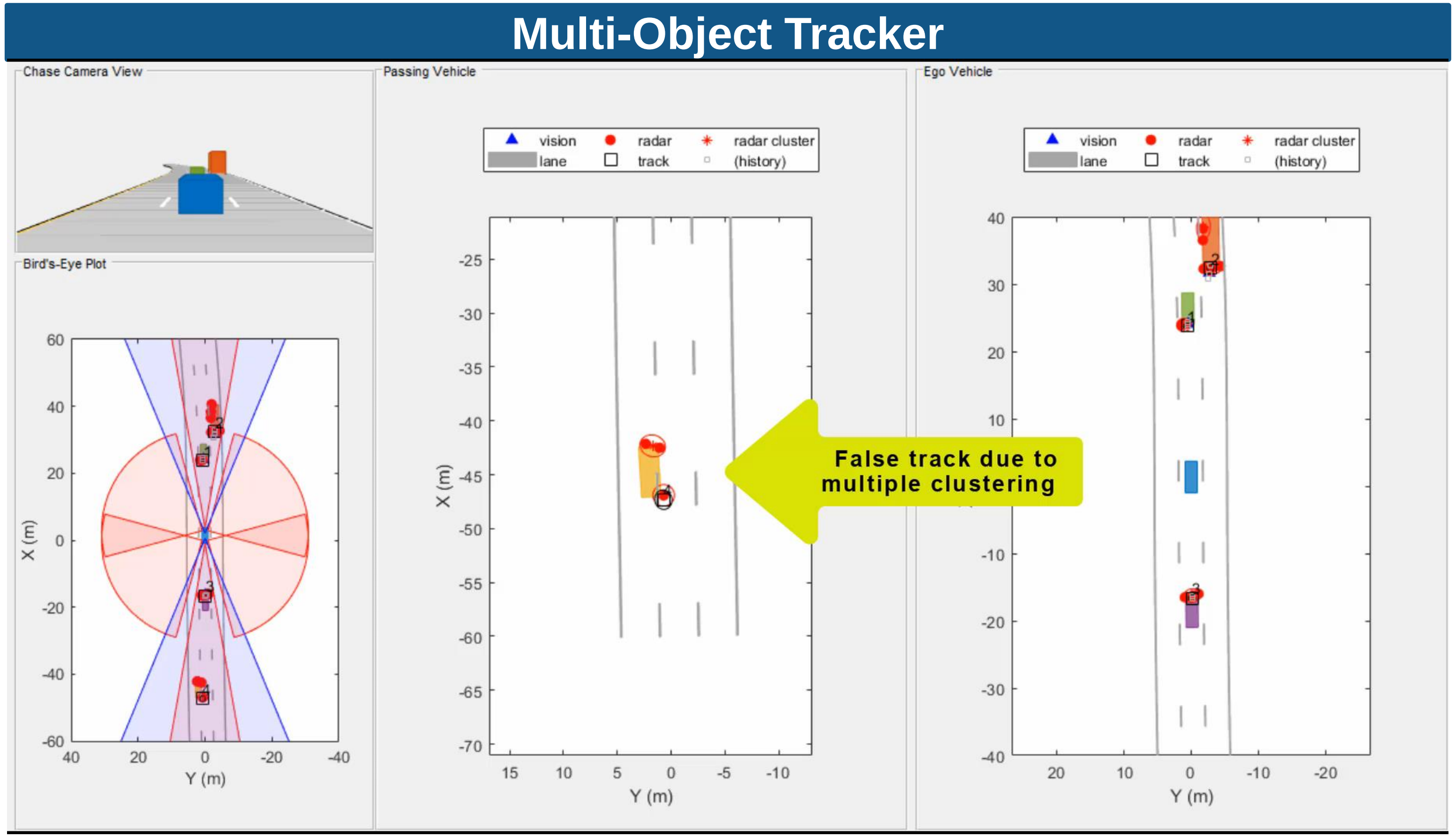
[Extended Object Tracking]()

- Design multi-object tracker
- Design extended object trackers
- Evaluate tracking metrics
- Evaluate error metrics
- Evaluate desktop execution time

Sensor Fusion and Tracking Toolbox™

Automated Driving Toolbox™

Updated R2019b



Design extended object trackers

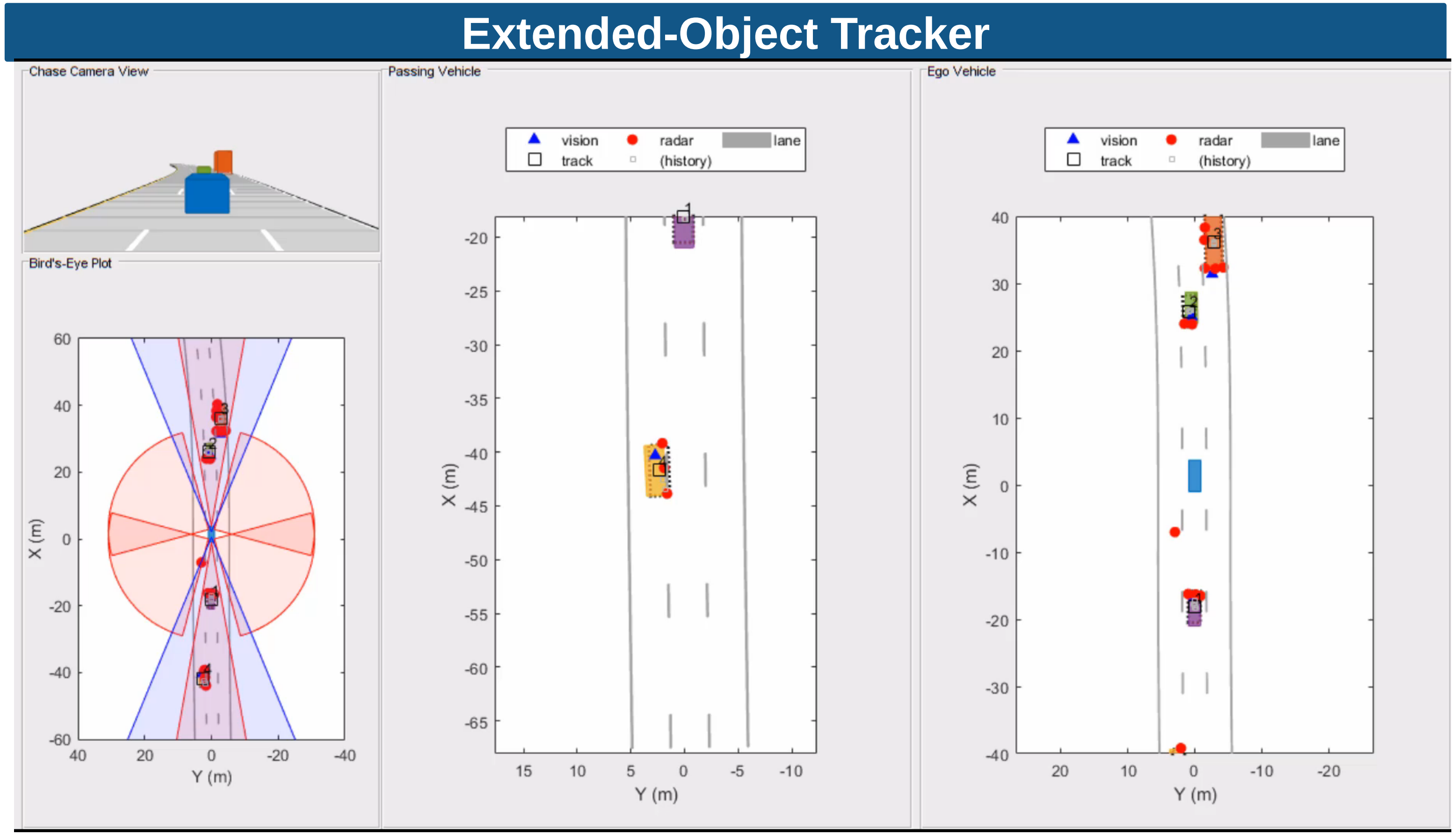
[Extended Object Tracking]

- Design multi-object tracker
- Design extended object trackers
- Evaluate tracking metrics
- Evaluate error metrics
- Evaluate desktop execution time

Sensor Fusion and Tracking Toolbox™

Automated Driving Toolbox™

Updated R2019b



Evaluate tracking performance

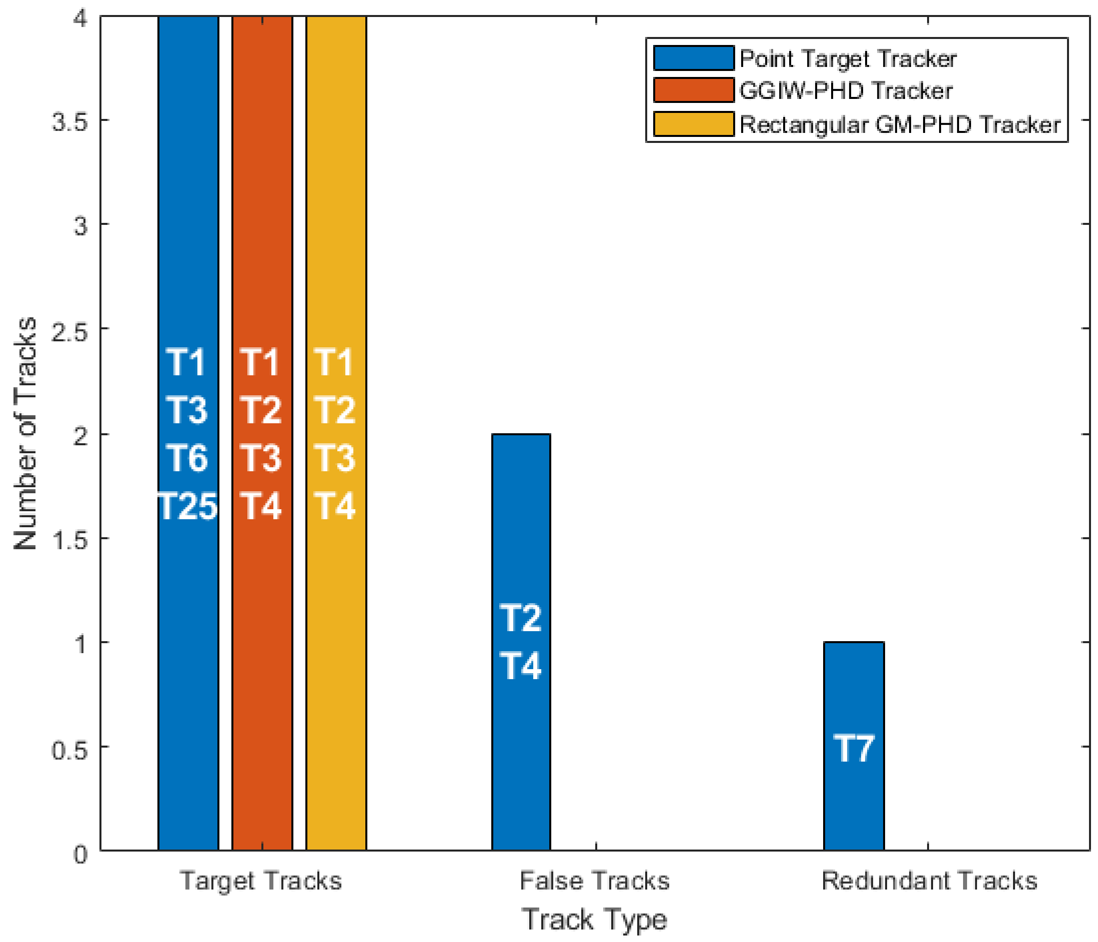
[Extended Object Tracking]

- Design multi-object tracker
- Design extended object trackers
- Evaluate tracking metrics
- Evaluate error metrics
- Evaluate desktop execution time

Sensor Fusion and Tracking Toolbox™

Automated Driving Toolbox™

Updated R2019b



Multi-object tracker

GGIW-Probability Hypothesis Density tracker

Extended object (size and orientation) tracker

Evaluate error metrics

[Extended Object Tracking]()

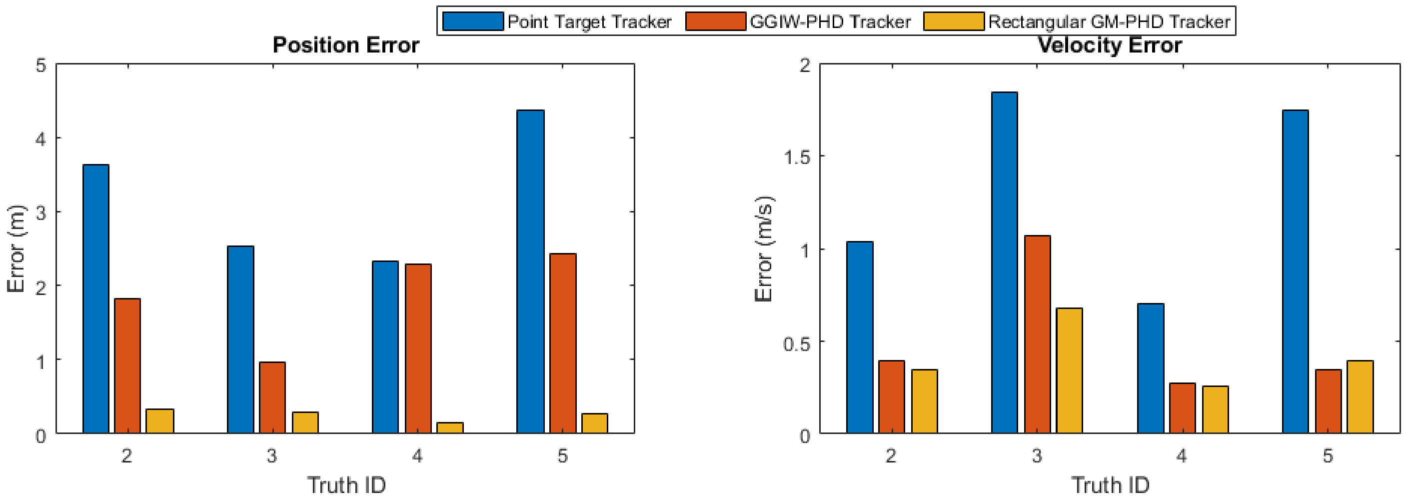
- Design multi-object tracker
- Design extended object trackers
- Evaluate tracking metrics
- Evaluate error metrics
- Evaluate desktop execution time

Sensor Fusion and

Tracking Toolbox™

Automated Driving Toolbox™

Updated R2019b



Multi-object tracker

GGIW-Probability Hypothesis Density tracker

Extended object (size and orientation) tracker

Evaluate OSPA metrics

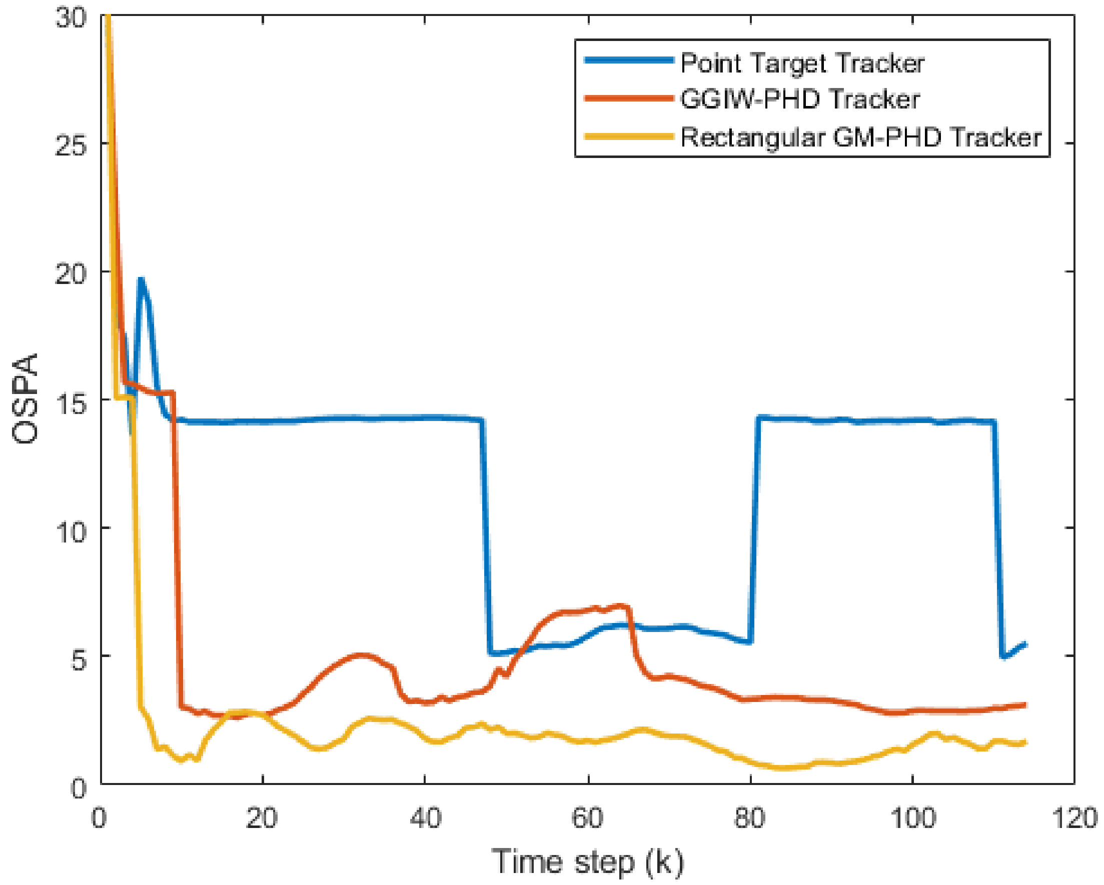
[Extended Object Tracking]

- Design multi-object tracker
- Design extended object trackers
- Evaluate tracking metrics
- Evaluate error metrics
- Evaluate desktop execution time

Sensor Fusion and Tracking Toolbox™

Automated Driving Toolbox™

Updated R2019b



Multi-object tracker

GGIW-Probability Hypothesis Density tracker

Extended object (size and orientation) tracker

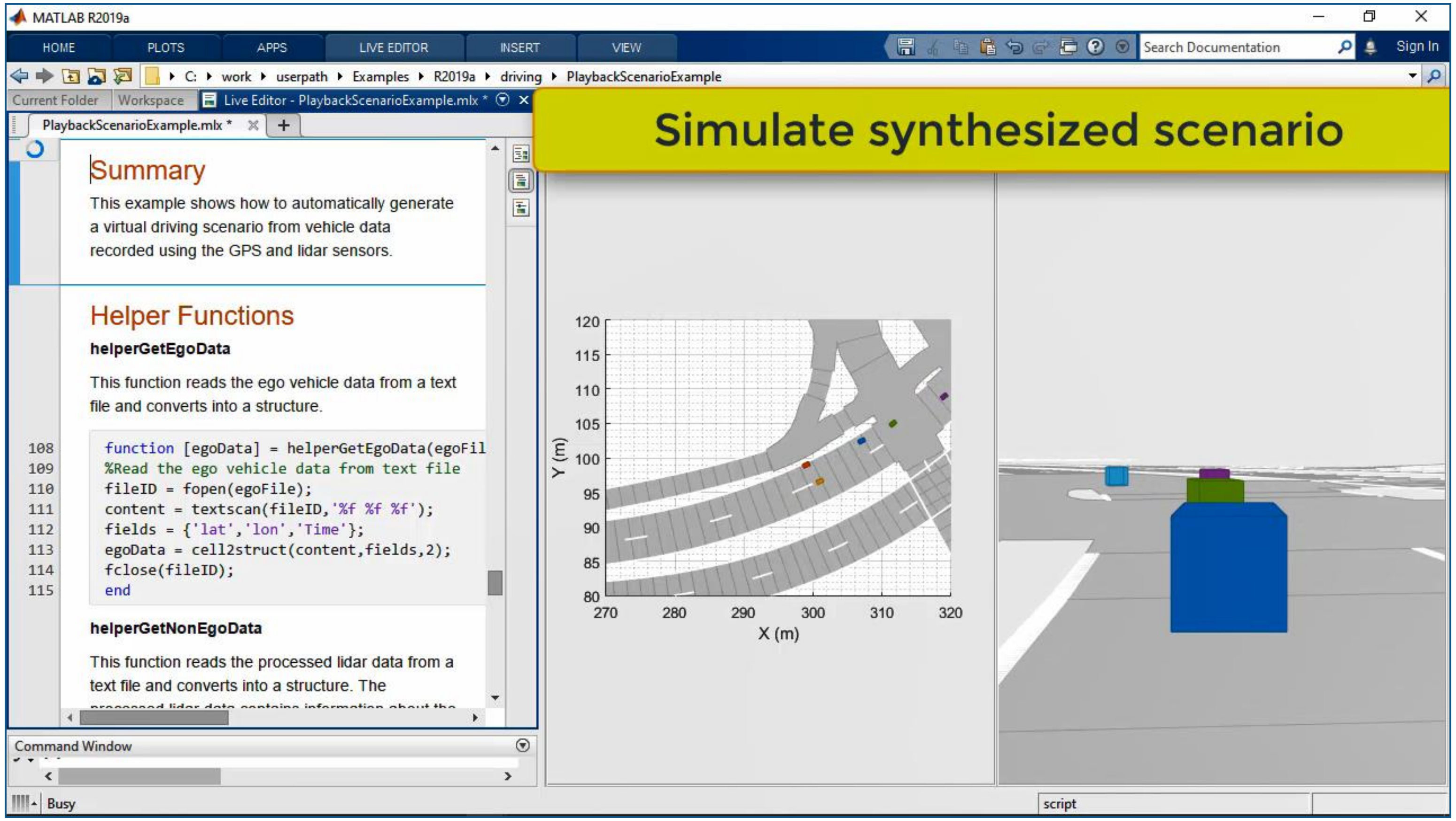
Synthesize driving scenarios from recorded data

[Scenario Generation from Recorded Vehicle Data]()

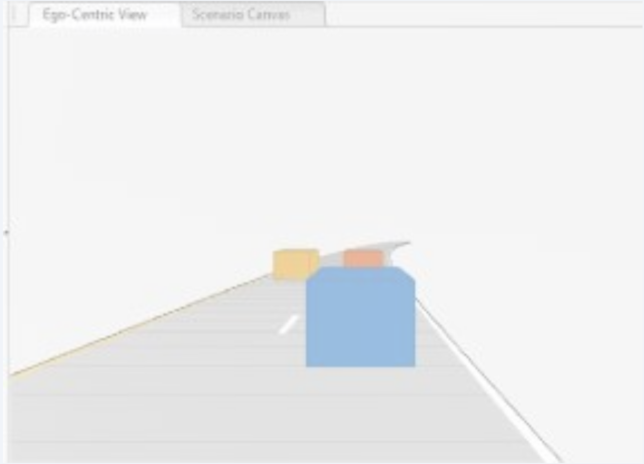
- Visualize video
- Import OpenDRIVE roads
- Import GPS
- Import object lists

Automated Driving Toolbox™

R2019a



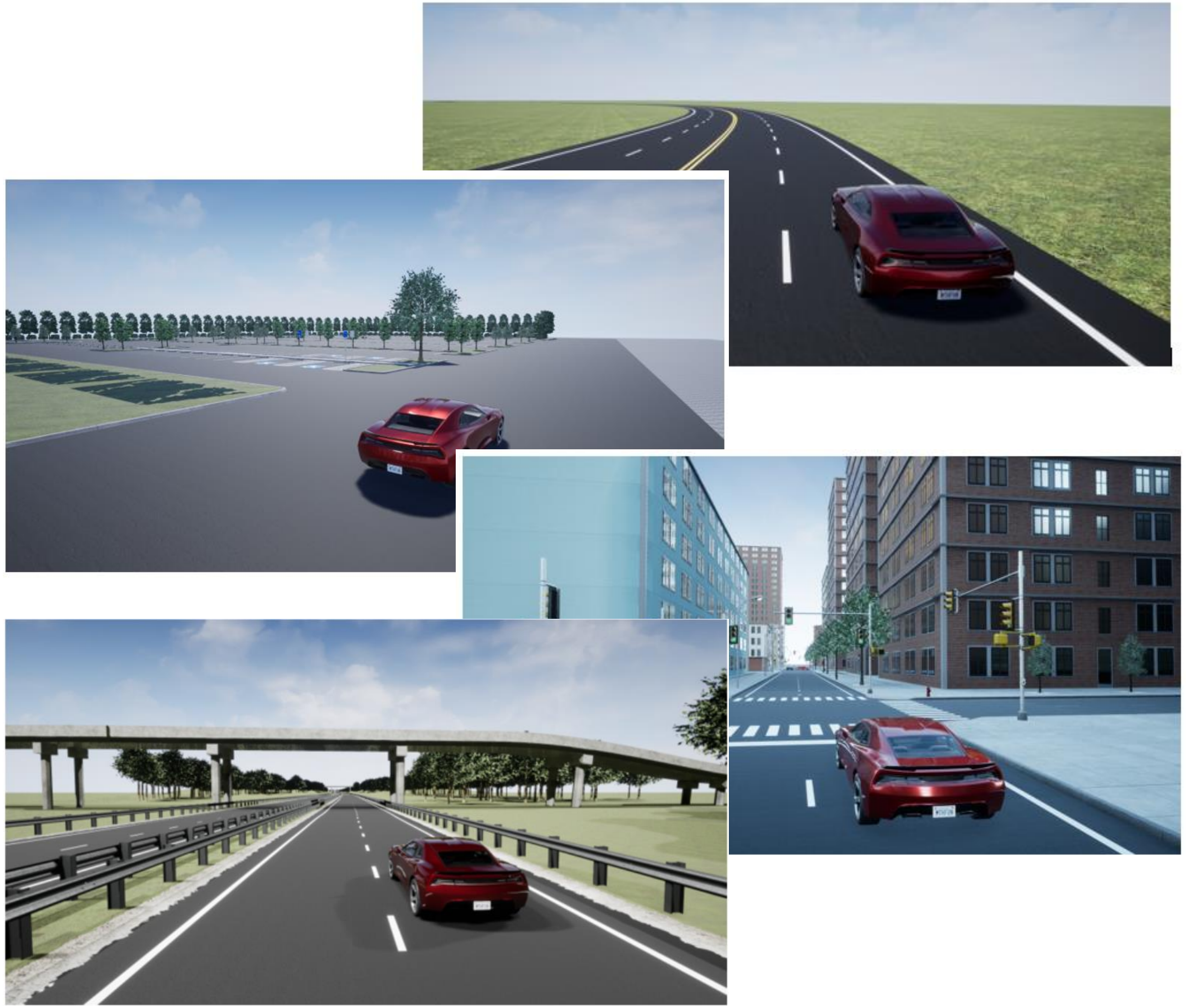
How can I design with virtual driving scenarios?

Scenes	**Cuboid** 	**3D Simulation (Unreal Engine)** 
Testing	Controls, sensor fusion, planning	Controls, sensor fusion, planning, perception
Authoring	Driving Scenario Designer App Programmatic API (drivingScenario)	Unreal Engine Editor
Sensing	Probabilistic radar (detection list) Probabilistic vision (detection list) Probabilistic lane (detection list)	Probabilistic radar (detection list) Monocular camera (image, labels, depth) Fisheye camera (image) Lidar (point cloud)

Select from prebuilt 3D simulation scenes

[3D Simulation for Automated Driving]

- Straight road
- Curved road
- Parking lot
- Double lane change
- Open surface
- US city block
- US highway
- Virtual Mcity



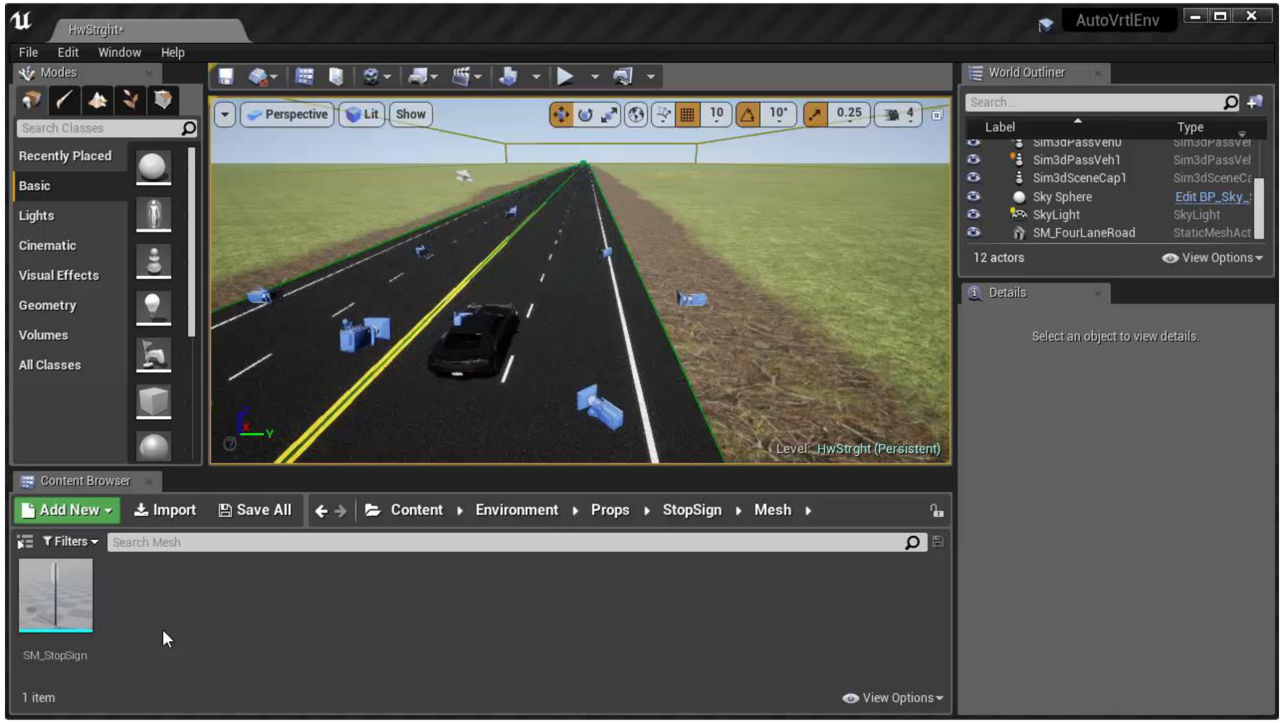
Automated Driving Toolbox™

R2019b

Customize 3D simulation scenes

[Support Package for Customizing Scenes]

- Install Unreal Engine
- Set up environment and open Unreal Editor
- Configure configuration Block for Unreal Editor co-simulation
- Use Unreal Editor to customize scenes
- Create an Unreal Engine project executable file



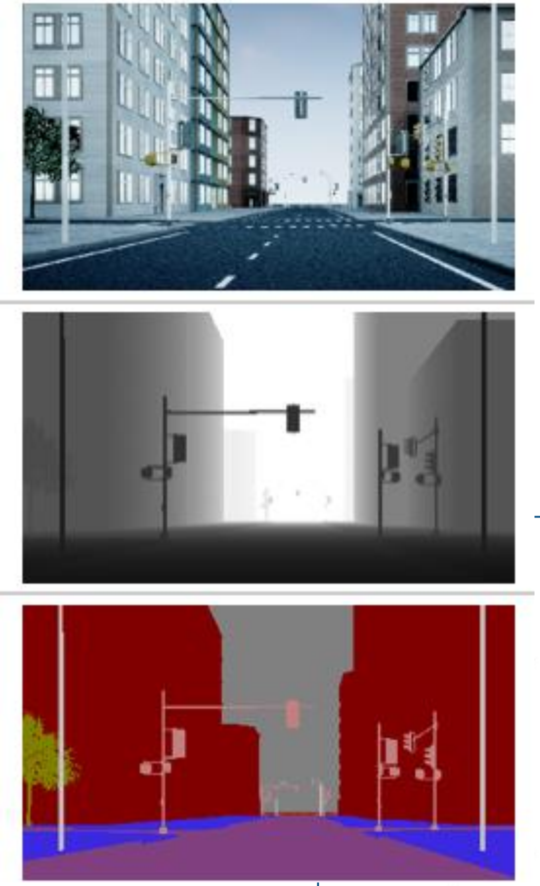
Vehicle Dynamics Blockset™

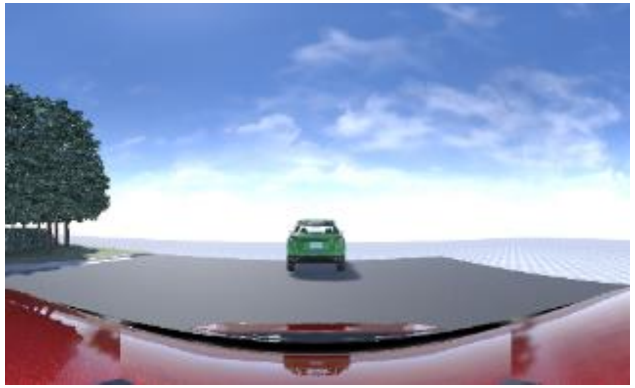
R2019b

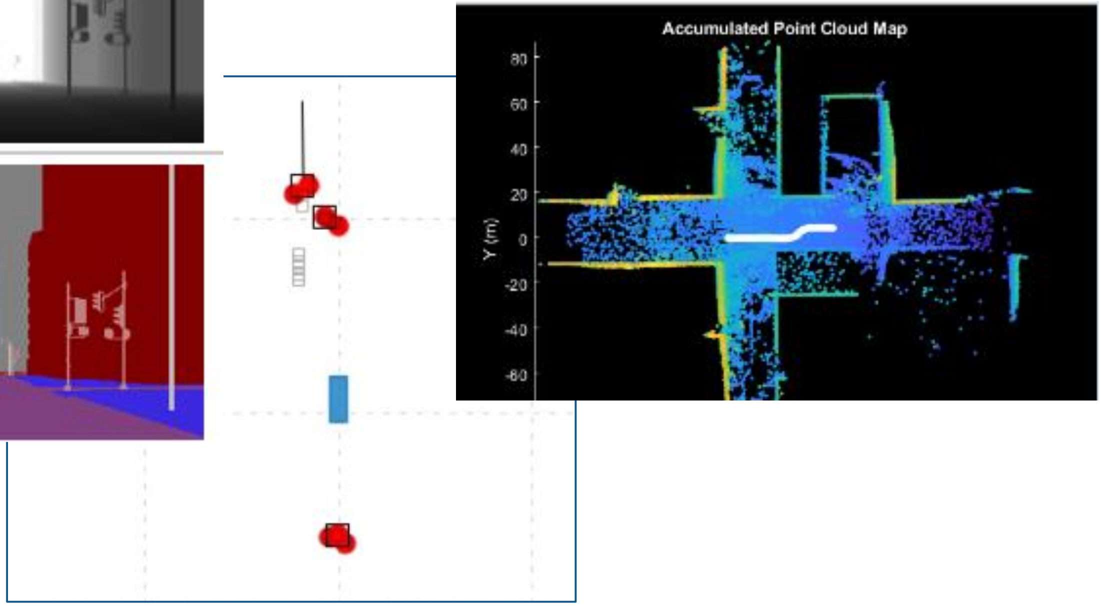
Model sensors in 3D simulation environment

[3D Simulation for Automated Driving]

- Monocular camera
- Fisheye camera
- Lidar
- Probabilistic radar







Automated Driving Toolbox™

R2019b

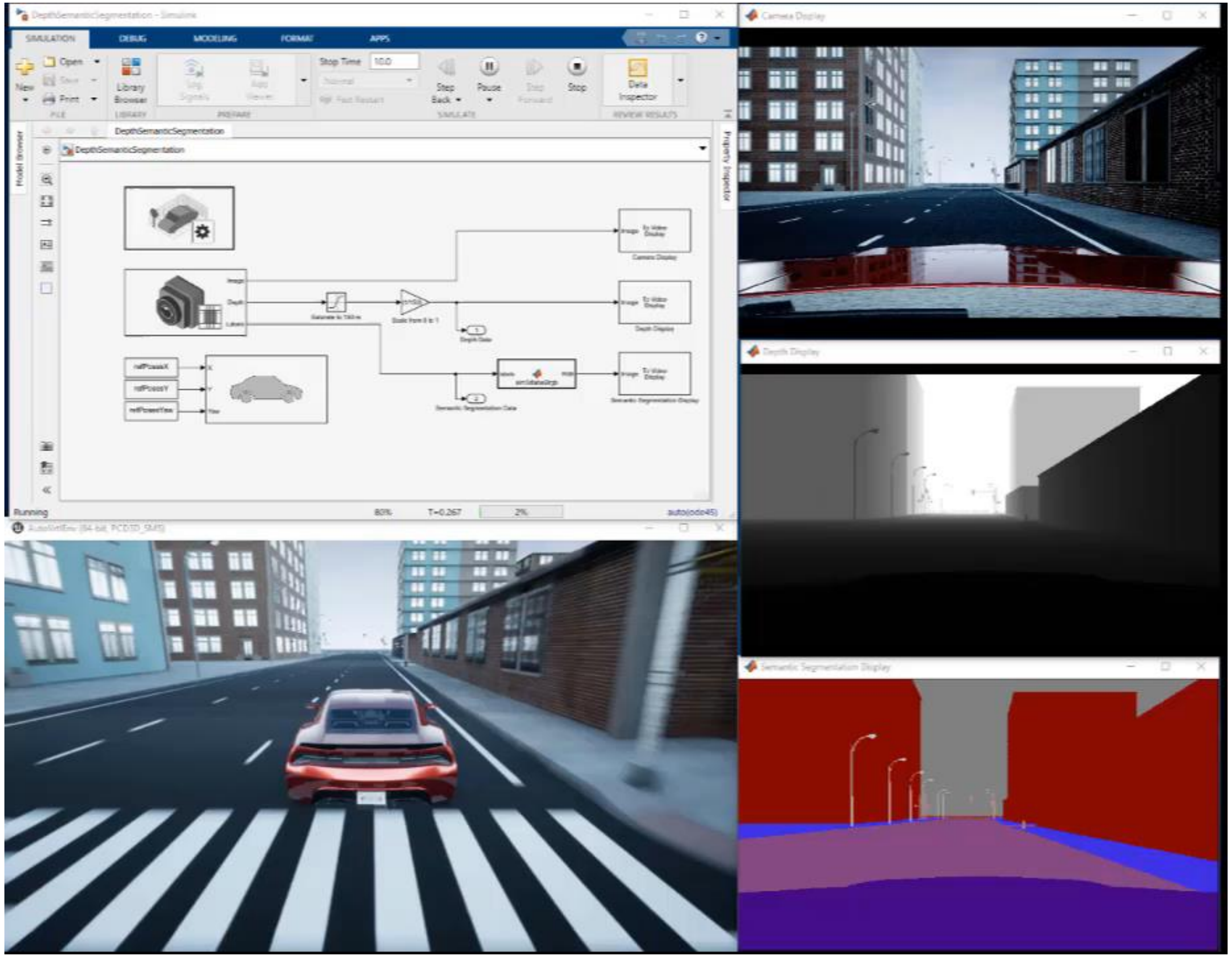
Synthesize monocular camera sensor data

[Visualize Depth and Semantic Segmentation Data in 3D Environment]

- Synthesize RGB image
- Synthesize depth map
- Synthesize sematic segmentation

Automated Driving Toolbox™

R2019b



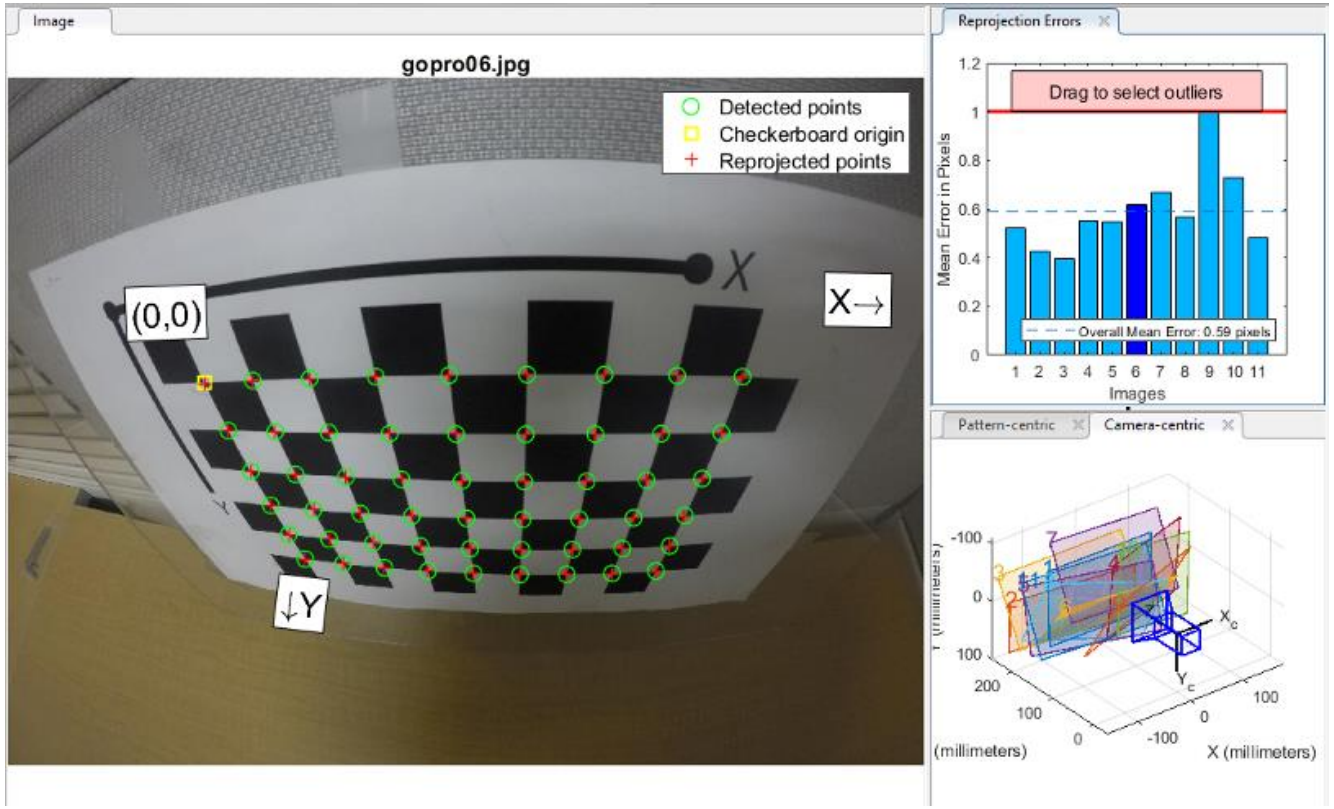
Calibrate monocular camera model

[Single Camera Calibrator App]

- Prepare the Pattern, Camera, and Images
- Add Images and Select Camera Model
- Calibrate
- Evaluate Calibration Results

Computer Vision Toolbox™

R2013b



Synthesize fisheye camera sensor data

[Simulate a Simple Driving Scenario and Sensor in 3D Environment]

- Explore camera model (Scaramuzza)
- Configure distortion center, image size and mapping coefficients
- Visualize results

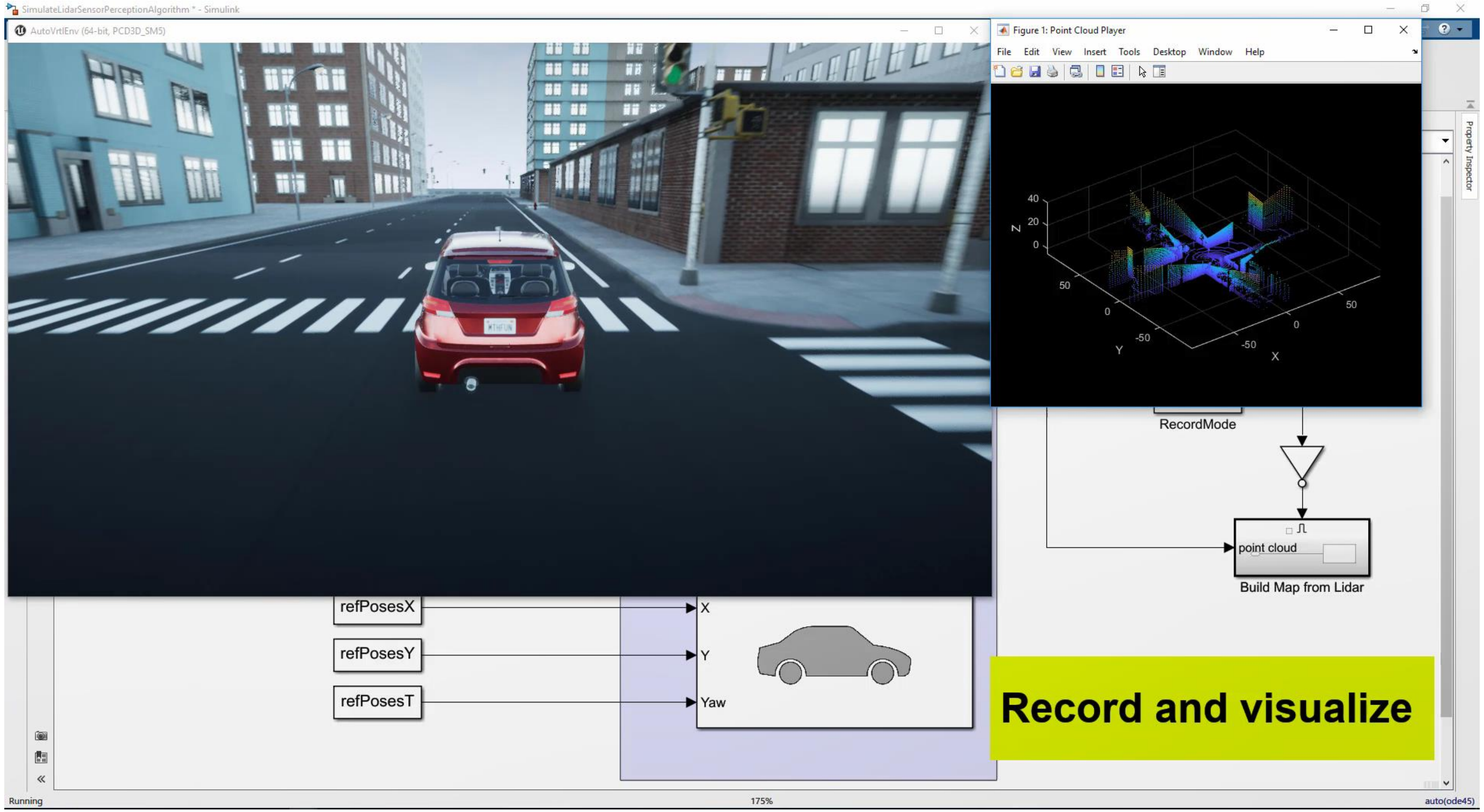
Automated Driving Toolbox™

R2019b

Synthesize lidar sensor data

[Simulate Lidar Sensor Perception Algorithm]

- Record and visualize
- Develop algorithm
- Build a 3D map
- Use algorithm within simulation environment



Automated Driving Toolbox™

R2019b

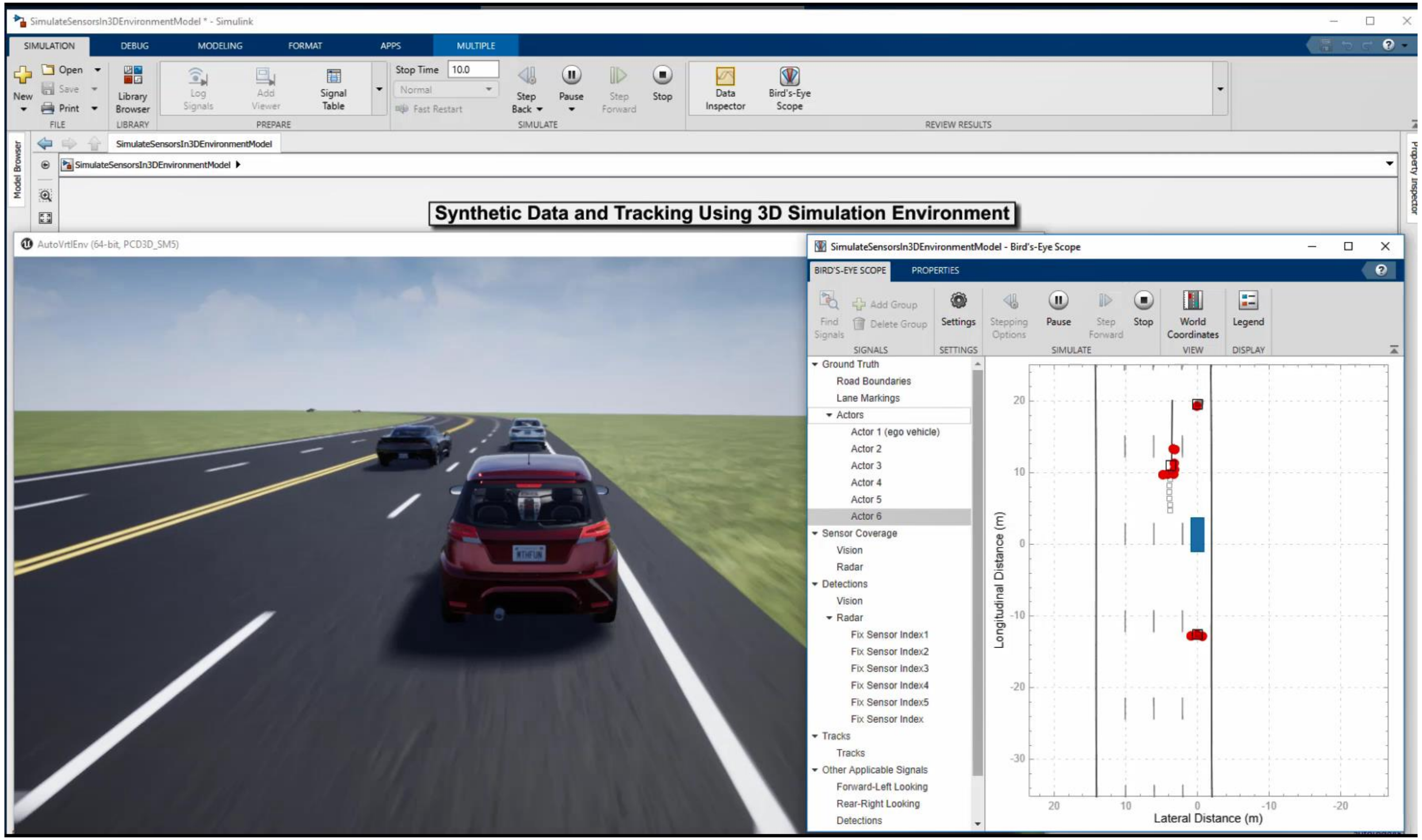
Synthesize radar sensor data

[Simulate Radar Sensors in 3D Environment]

- Extract the center locations
- Use center location for road creation using driving scenario
- Define multiple moving vehicles
- Export trajectories from app
- Configure multiple probabilistic radar models
- Calculate confirmed track

Automated Driving Toolbox™

R2019b



Communicate with the 3D simulation environment

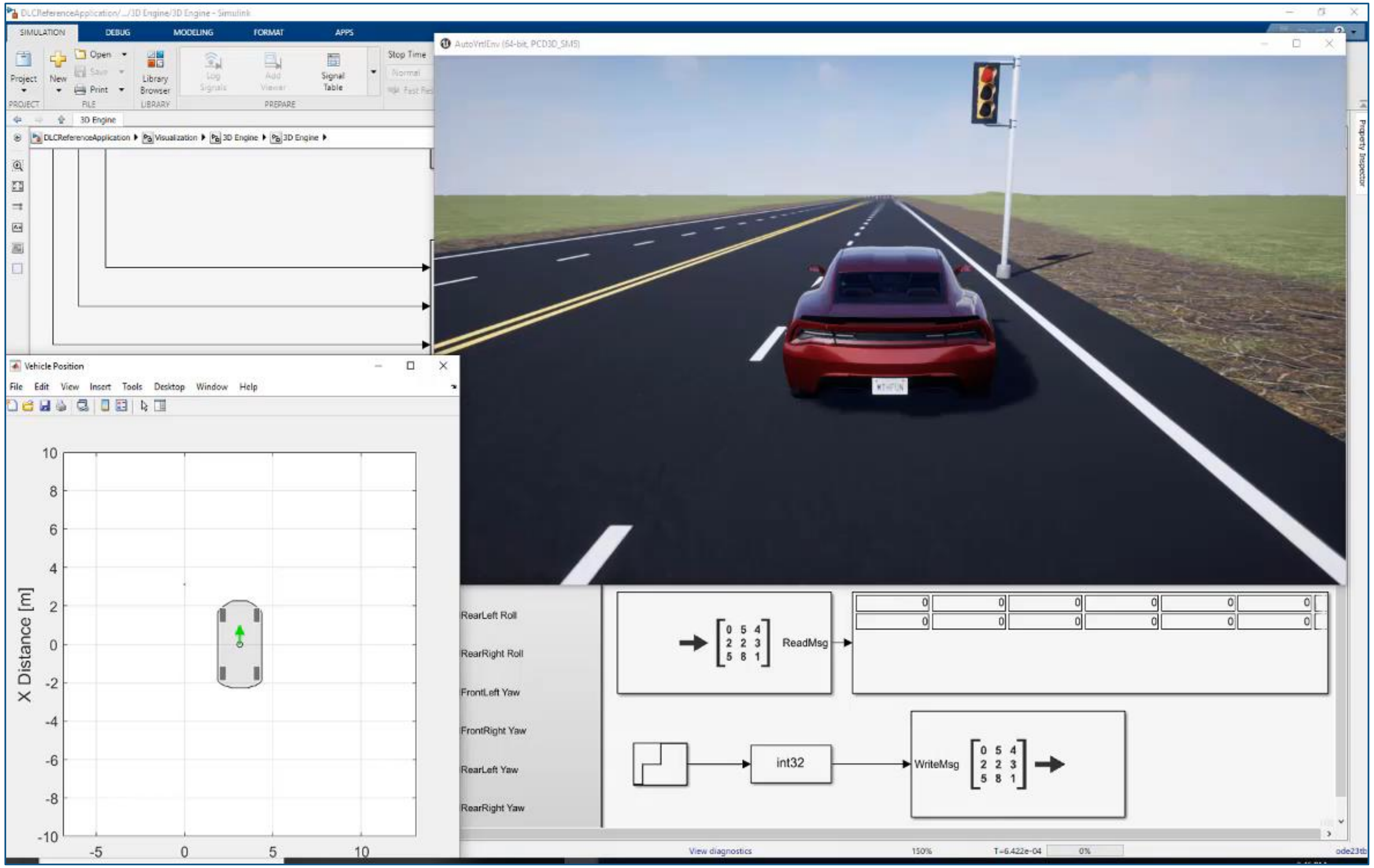
[Send and Receive Double-Lane Change Scene Data]()

- Simulation 3D Message Set
  - Send data to Unreal Engine
  - Traffic light color

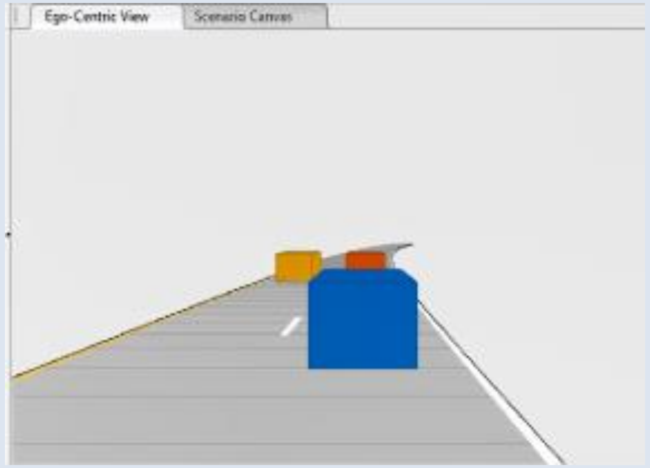
- Simulation 3D Message Get
  - Retrieve data from Unreal Engine
  - Number of cones hit

Vehicle Dynamics BlocksetTM

R2019b



How can I design with virtual driving scenarios?

Scenes	**Cuboid**	**3D Simulation (Unreal Engine)**
		
Testing	Controls, sensor fusion, planning	Controls, sensor fusion, planning, perception
Authoring	Driving Scenario Designer App Programmatic API (drivingScenario)	Unreal Engine Editor
Sensing	Probabilistic radar (detection list) Probabilistic vision (detection list) Probabilistic lane (detection list)	Probabilistic radar (detection list) Monocular camera (image, labels, depth) Fisheye camera (image) Lidar (point cloud)

MathWorks Simulation Challenge

for SAE AutoDrive year 3 competition

#	Task	Points
1	Synthesize data to test open loop perception algorithm	10
2	**Synthesize data to test closed loop controls algorithm** a. What algorithm did you test and why did you chose it? (i.e. lateral controls, longitudinal controls, decision logic, planning) b. How did you synthesize scenarios? (i.e. Unreal Engine, Driving Scenario Designer, sensors, vehicle dynamics, customizations) c. How did you assess correctness of the algorithm? (i.e. specify expected behavior, assess metrics, automated testing)	10
3	Generate code from controls algorithm	10
4	Innovate	15
5	Reflect	5

Integrate driving scenario into closed loop simulation

[Lane Following Control with Sensor Fusion]

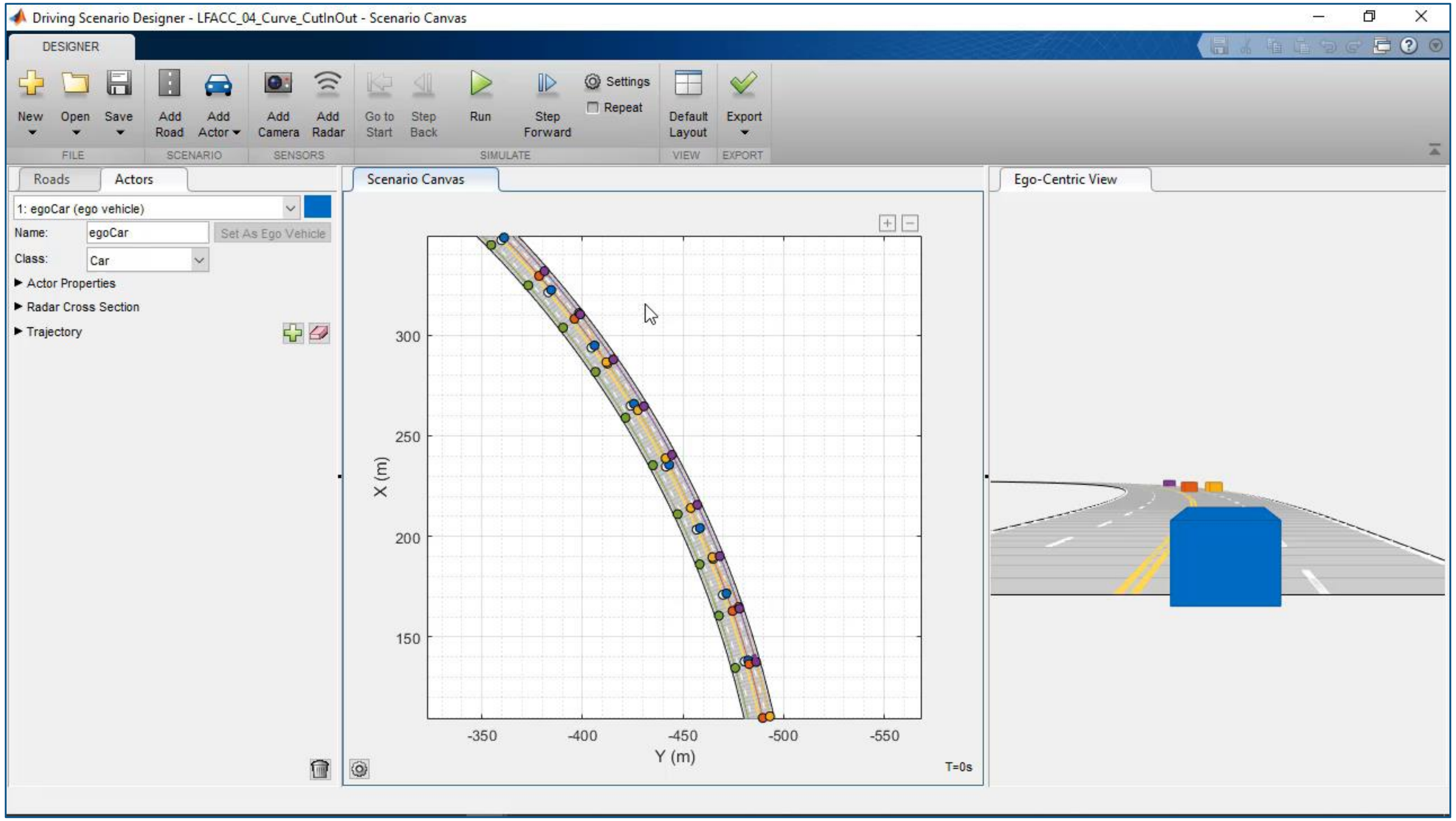
- Integrate scenario into system
- Design lateral (lane keeping) and longitudinal (lane spacing) model predictive controllers
- Visualize sensors and tracks
- Generate C/C++ code
- Test with software in the loop (SIL) simulation

Model Predictive Control Toolbox™

Automated Driving Toolbox™

Embedded Coder®

R2018b



Design lateral and longitudinal controls

[Lane Following Control with Sensor Fusion]

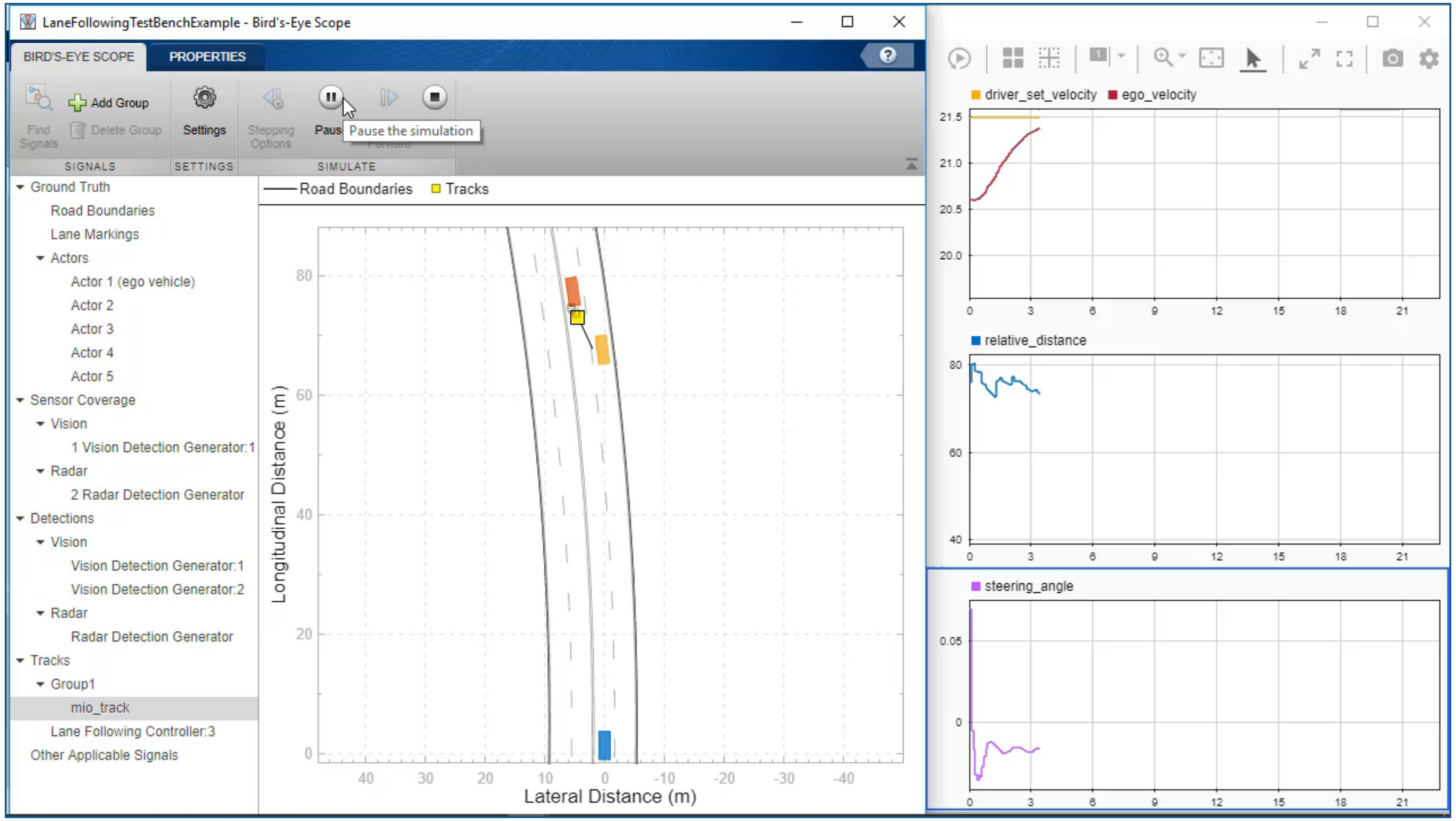
- Integrate scenario into system
- Design lateral (lane keeping) and longitudinal (lane spacing) model predictive controllers
- Visualize sensors and tracks
- Generate C/C++ code
- Test with software in the loop (SIL) simulation

Model Predictive Control Toolbox™

Automated Driving Toolbox™

Embedded Coder®

R2018b



Visualize sensor detections and tracks

[Lane Following Control with Sensor Fusion]

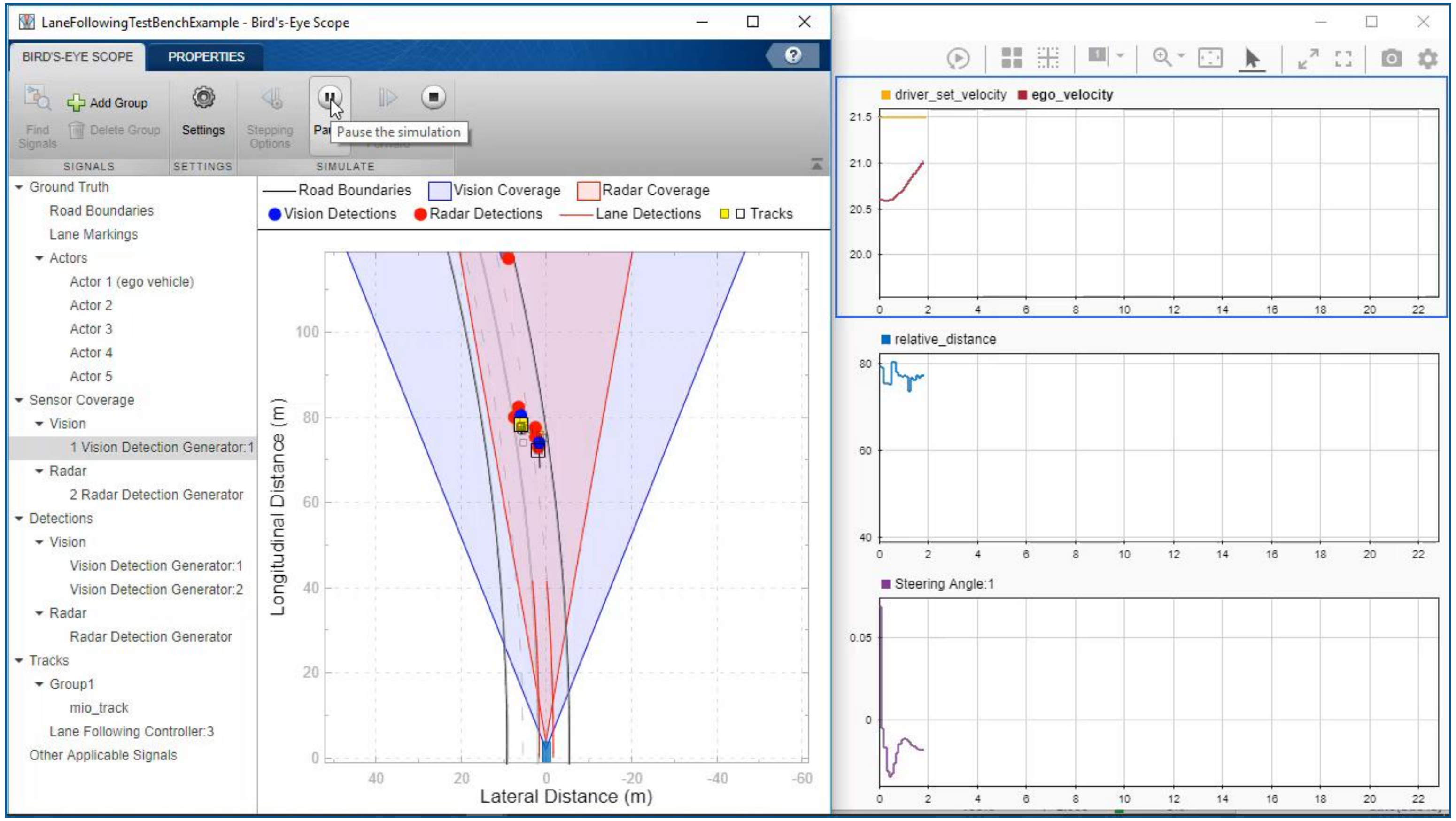
- Integrate scenario into system
- Design lateral (lane keeping) and longitudinal (lane spacing) model predictive controllers
- Visualize sensors and tracks
- Generate C/C++ code
- Test with software in the loop (SIL) simulation

Model Predictive Control Toolbox™

Automated Driving Toolbox™

Embedded Coder®

R2018b



Automate testing against driving scenarios

[Testing a Lane Following Controller with Simulink Test]

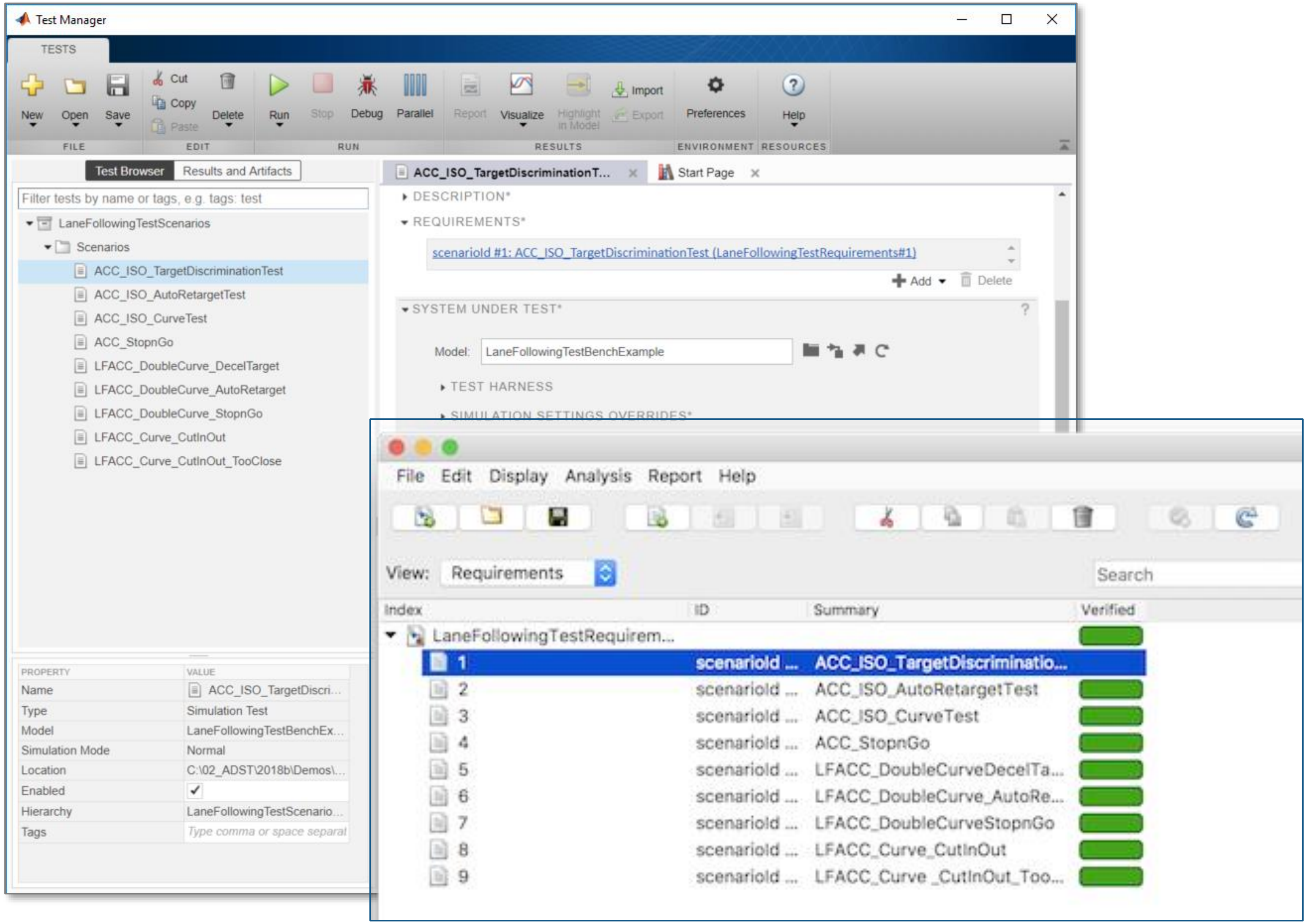
- Author high level requirements
- Synthesize driving scenarios
- Specify assessment criteria
- Run interactive simulation
- Automate regression testing
- Review verification status

Simulink Test™

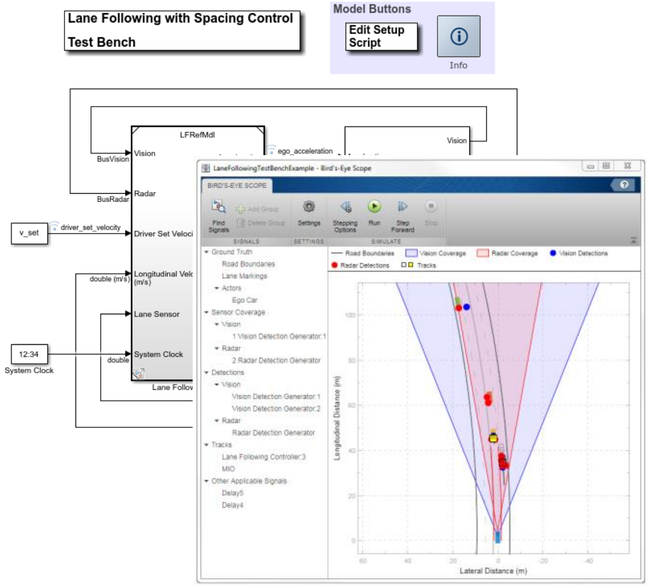
Automated Driving Toolbox™

Model Predictive Control Toolbox™

R2018b



Synthesize scenarios to test your design



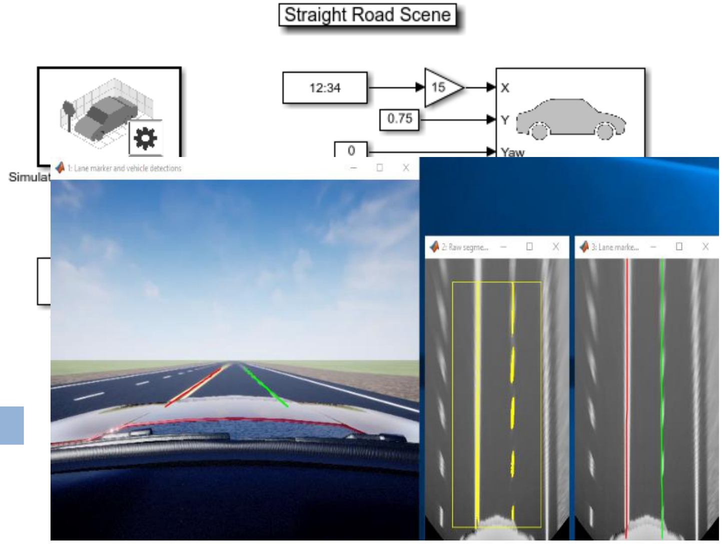
[Lane Following Control with Sensor Fusion](#)

Model Predictive Control Toolbox™

Automated Driving Toolbox™

Embedded Coder®

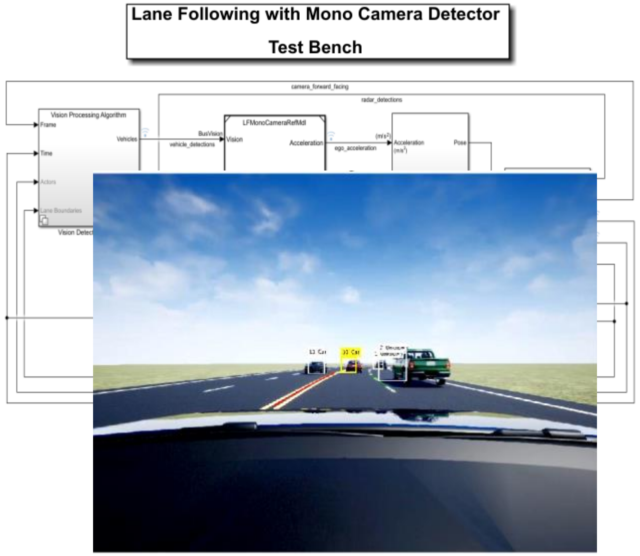
R2018b



[Design of Lane Marker Detector in 3D Simulation Environment](#)

Automated Driving Toolbox™

R2019b



[Lane-Following Control with Monocular Camera Perception](#)

Model Predictive Control Toolbox™

Automated Driving Toolbox™

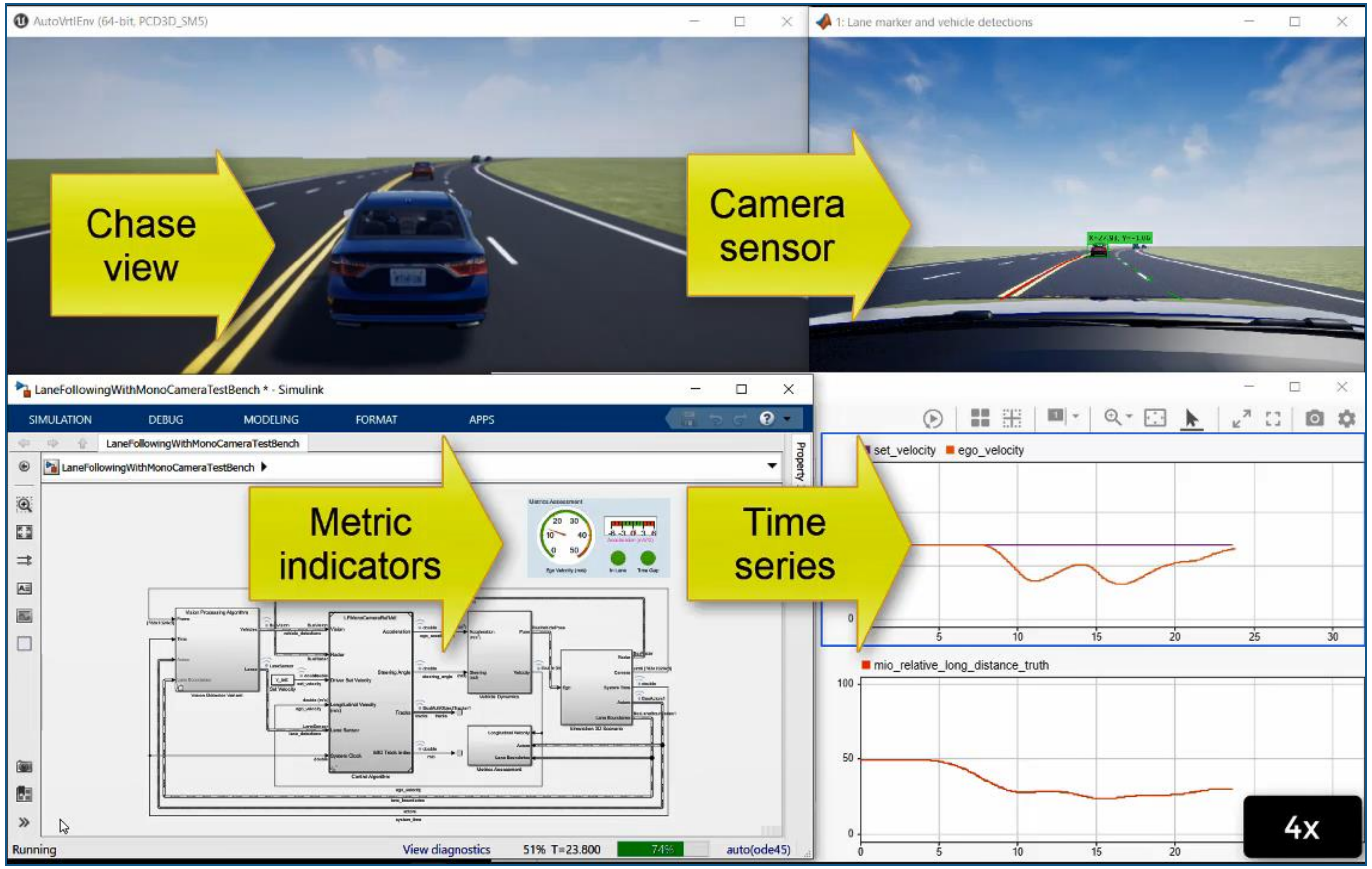
Vehicle Dynamics Blockset™

Updated R2019b

Simulate controls with perception

[Lane-Following Control with Monocular Camera Perception]

- Author target vehicle trajectories
- Synthesize monocular camera and probabilistic radar sensors
- Model lane following and spacing control in Simulink
- Model lane boundary and vehicle detectors in MATLAB code



Model Predictive Control Toolbox™

Automated Driving Toolbox™

Vehicle Dynamics Blockset™

Updated R2019b

Visualize logged simulation detection and camera data

[Lane-Following Control with Monocular Camera Perception]

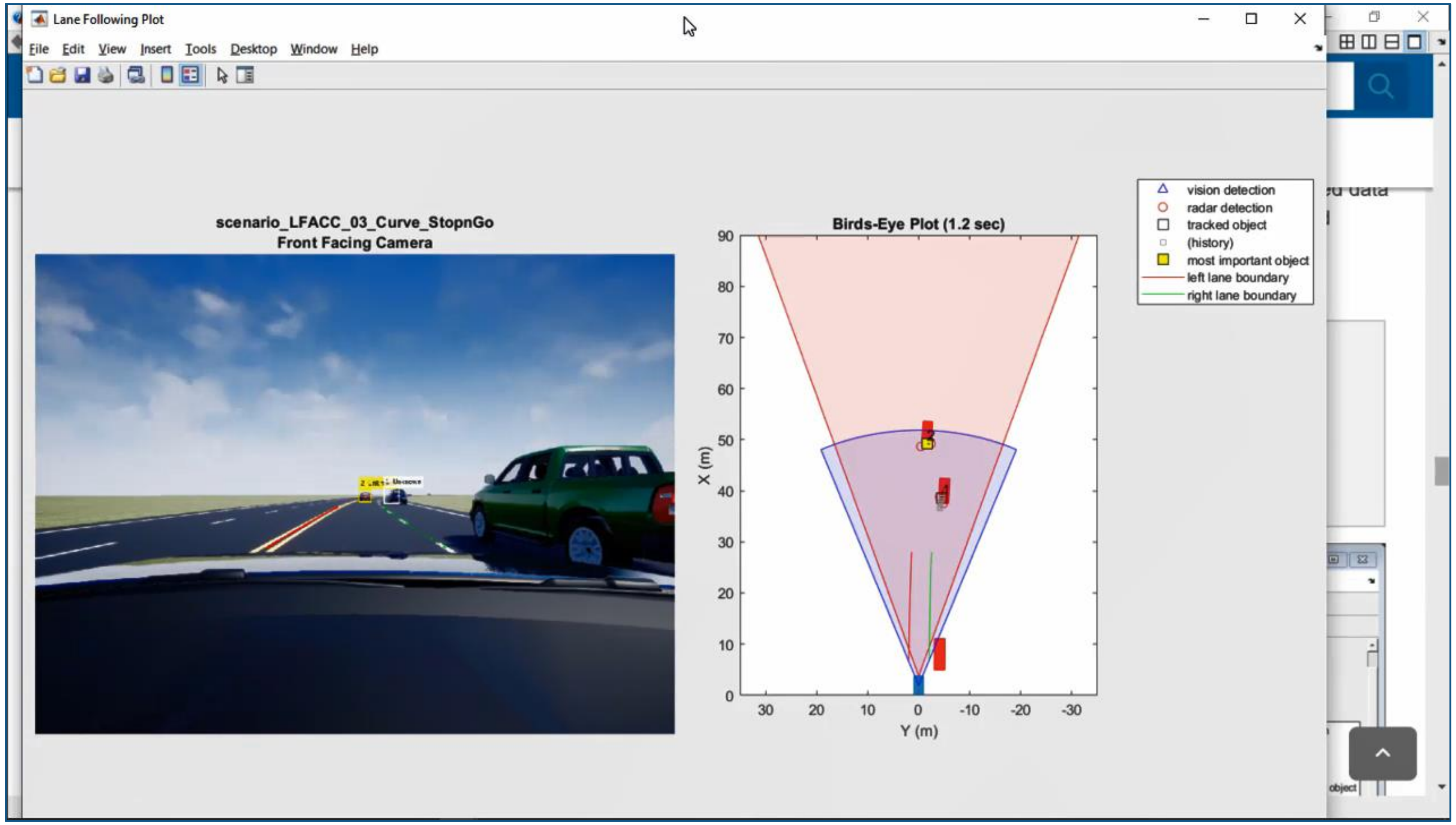
- Author target vehicle trajectories
- Synthesize monocular camera and probabilistic radar sensors
- Model lane following and spacing control in Simulink
- Model lane boundary and vehicle detectors in MATLAB code

Model Predictive Control Toolbox™

Automated Driving Toolbox™

Vehicle Dynamics Blockset™

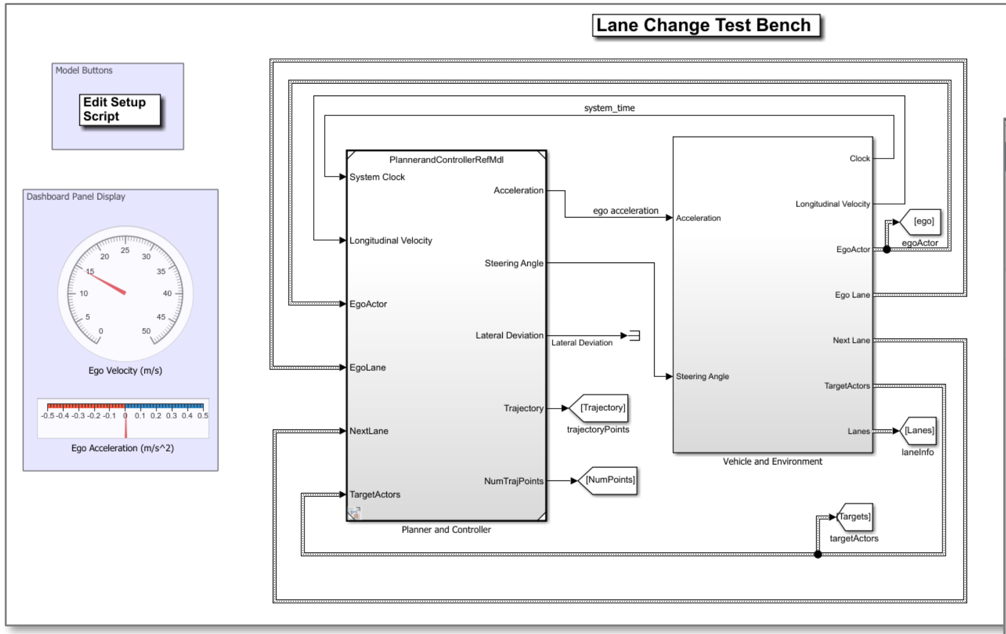
Updated R2019b

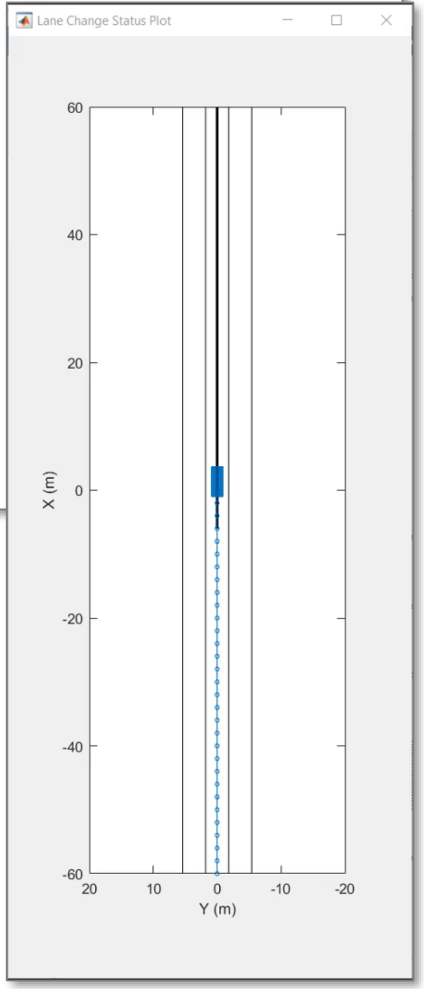


Design highway automated lane change maneuver

[Lane Change for Highway Driving]

- Find most important objects
- Generate optimal trajectory for collision-free lane change
- Extract path from trajectory
- Follow path with Model Predictive Control (MPC)





Navigation Toolbox™

Model Predictive Control Toolbox™

Automated Driving Toolbox™

R2019b

MathWorks Simulation Challenge

for SAE AutoDrive year 3 competition

#	Task	Points
1	Synthesize data to test open loop perception algorithm	10
2	Synthesize data to test closed loop controls algorithm	10
3	**Generate code from controls algorithm** a. What algorithm did you choose to generate code and why did you chose it? (i.e. lateral controls, longitudinal controls, decision logic, planning) b. Did you run software in the loop (SIL) simulation and what insight did you gain? (i.e. equivalent numeric results, measured execution time, code coverage) c. Did you integrate and assess the generated code running in your vehicle and what insight did you gain? (i.e. C++, ROS node, assessment of functionality compared with simulation)	10
4	Innovate	15
5	Reflect	5

Generate C/C++ code for path planner and controller

[Code Generation for Path Planning and Vehicle Control]

- Simulate system
- Configure for code generation
- Generate C/C++ code
- Test using Software-In-the-Loop
- Measure execution time of generated code

Automated Driving Toolbox™

Embedded Coder

R2019a

```
// model step function
void step0();

// model step function
void step1();

// model terminate function
void terminate();

// Constructor
AutomatedParkingValetModelClass();

// Destructor
~AutomatedParkingValetModelClass();

// Root inport: '<Root>/Costmap' set method
void setCostmap(costmapBus localArgInput);

// Root inport: '<Root>/GoalPose' set method
void setGoalPose(real_T localArgInput[3]);
```

Step0 = 50 msec rate
Step1 = 100 msec rate

Methods to access inputs and outputs

Measure execution time of generated code

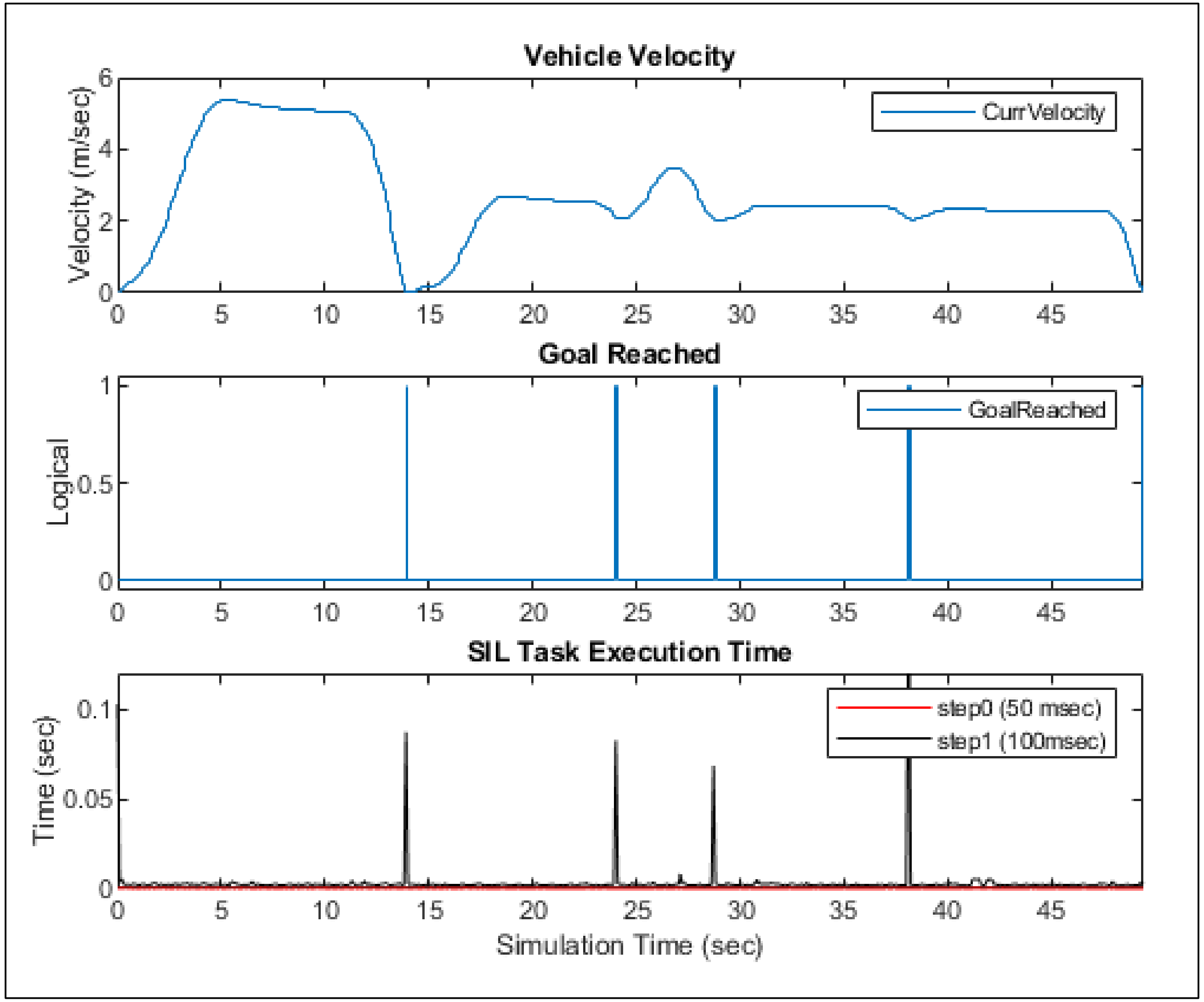
[Code Generation for Path Planning and Vehicle Control]()

- Simulate system
- Configure for code generation
- Generate C/C++ code
- Test using Software-In-the-Loop
- Measure execution time of generated code

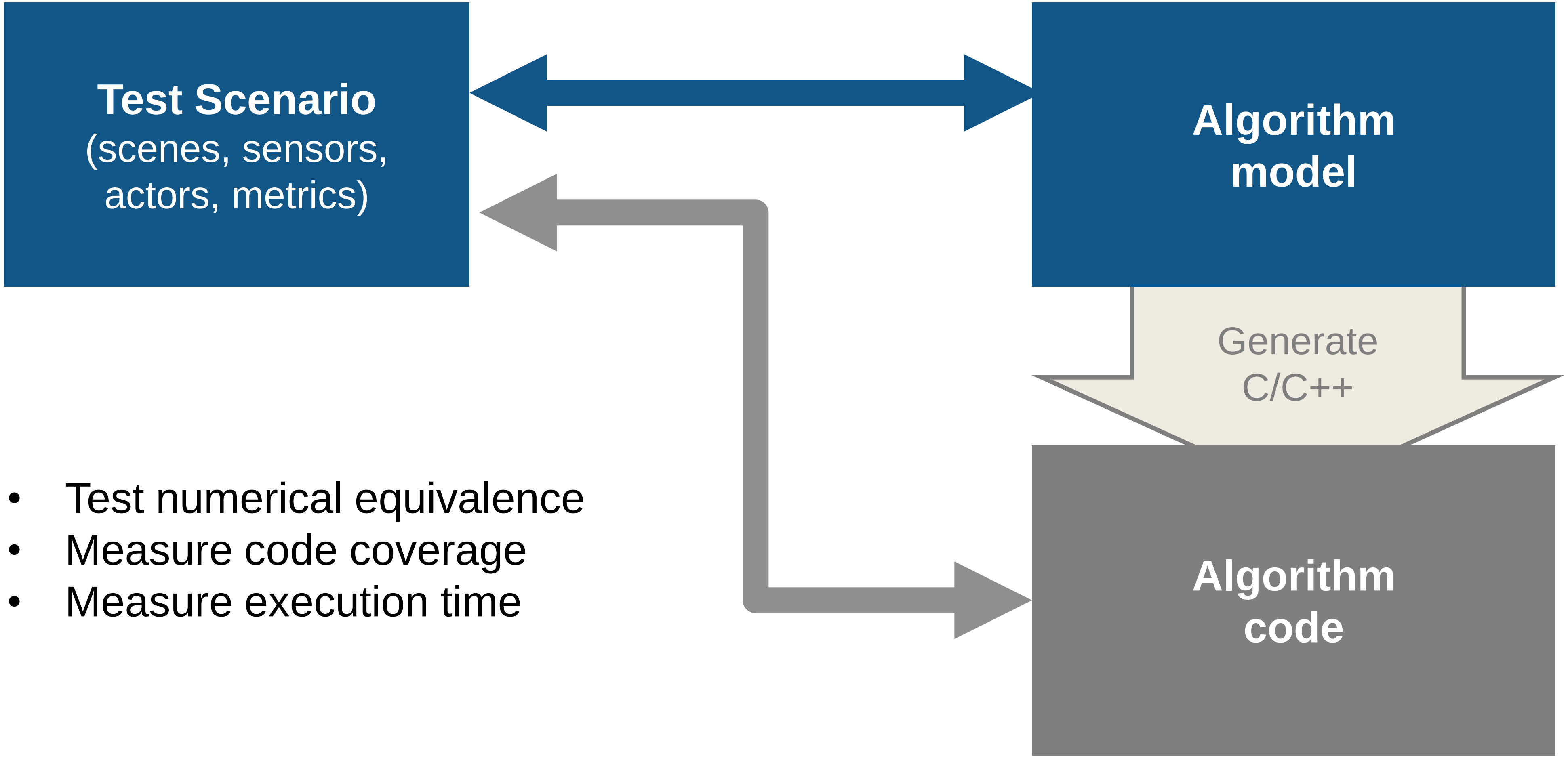
Automated Driving Toolbox™

Embedded Coder

R2019a



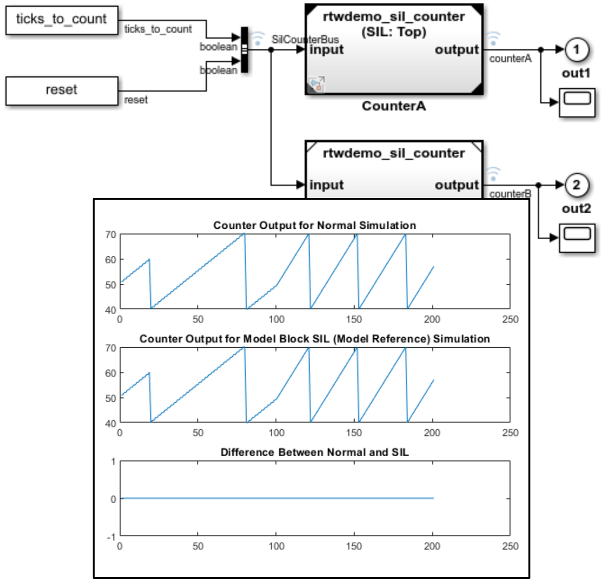
Evaluate generated code with software-in-the-loop (SIL) simulation



- Test numerical equivalence
- Measure code coverage
- Measure execution time

Evaluate generated code with software-in-the-loop (SIL) simulation

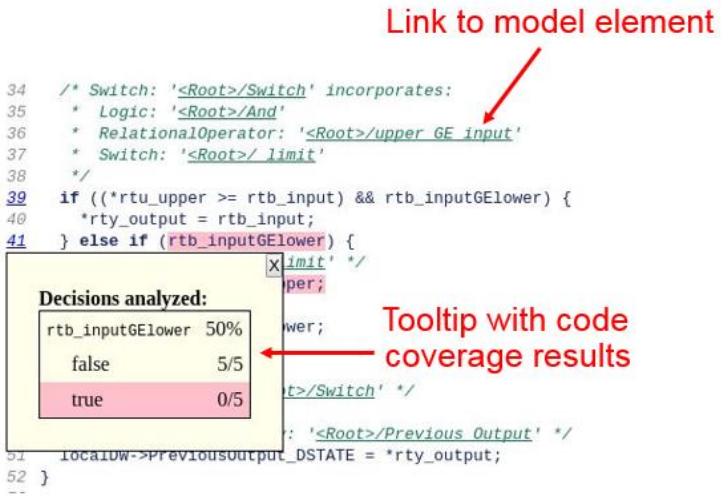
Test numerical equivalence



[Software-in-the-Loop Simulation]

Embedded Coder®

Measure code coverage



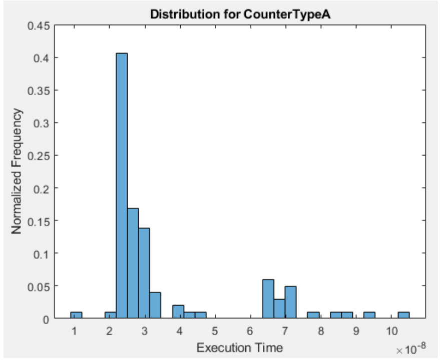
[Code Coverage for Models in Software-in-the-Loop (SIL) Mode]

Embedded Coder®

Profile code execution time

2. Profiled Sections of Code

Section	Maximum Execution Time in ns	Average Execution Time in ns
[+] rtwdemo_sil_topmodel_initialize	80	80
[+] rtwdemo_sil_topmodel_step [0.1 0]	358	129

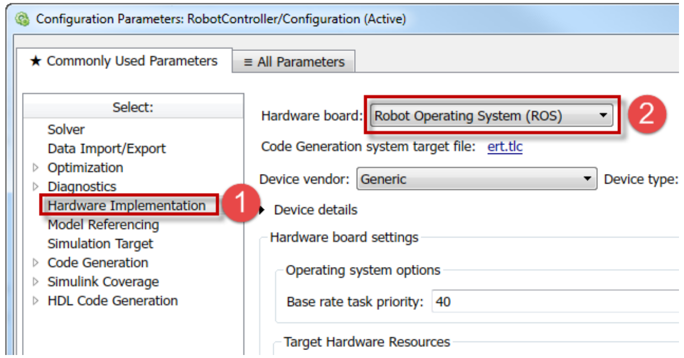


[View and Compare Code Execution Times]

Embedded Coder®

Deploy to ROS node

Generate standalone ROS node



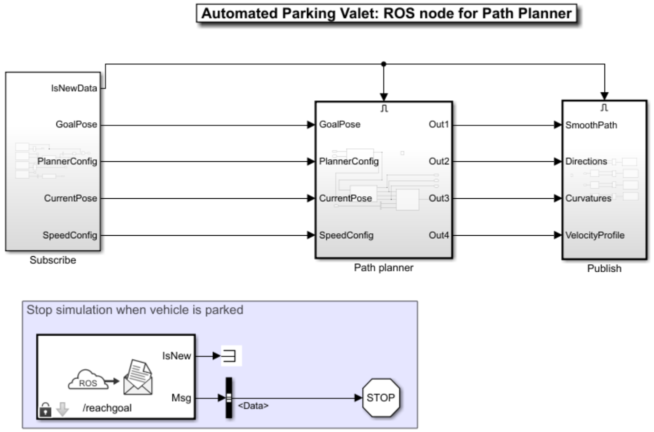
[Generate a Standalone ROS Node from Simulink]

ROS Toolbox™

Embedded Coder®

R2019b

Generate ROS nodes for parking valet



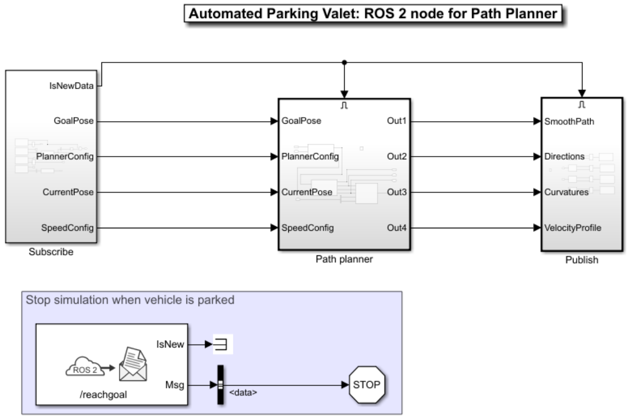
[Automated Parking Valet with ROS in Simulink]

ROS Toolbox™

Embedded Coder®

R2019b

Generate ROS 2.0 nodes for parking valet



[Automated Parking Valet with ROS 2 in Simulink]

ROS Toolbox™

Embedded Coder®

R2019b

MathWorks Simulation Challenge

for SAE AutoDrive year 3 competition

#	Task	Points
1	Synthesize data to test open loop perception algorithm	10
2	Synthesize data to test closed loop controls algorithm	10
3	Generate code from controls algorithm	10
4	**Innovate** a. What did you do with MathWorks tools that differentiates you from other teams? (i.e. Analyze recorded ROS/CAN data, label recorded data, train deep learning network, build a custom App, share work through projects with revision control) b. What insight did you gain while doing this?	15
5	Reflect	5

Connect to CAN and CAN-FD data

CAN platform support

- Connect to live data
- Read logged data

Vehicle Network Toolbox™

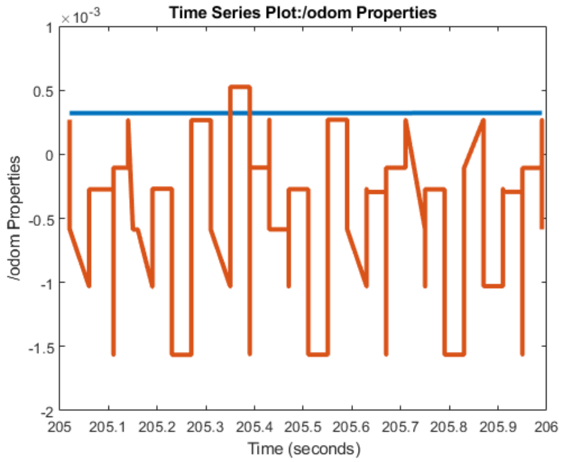
Updated R2019b

Vendor	Windows®	Linux®
MathWorks® virtual channels	✓	✓
Vector	✓	
PEAK-System	✓	✓
Kvaser	✓	✓
National Instruments®	✓	

File Format	Windows	Linux
BLF	✓	
CDF	✓	✓
MDF	✓	

Integrate with ROS 1.0 and ROS 2.0

Replay logged ROS data

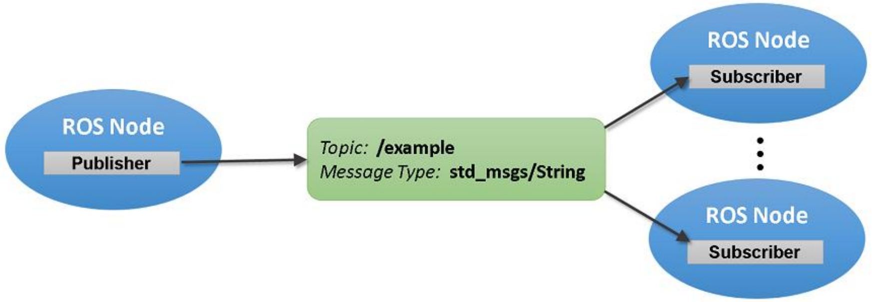


[Work with rosbag Logfiles](#)

ROS Toolbox™

R2019b

Connect to live ROS data

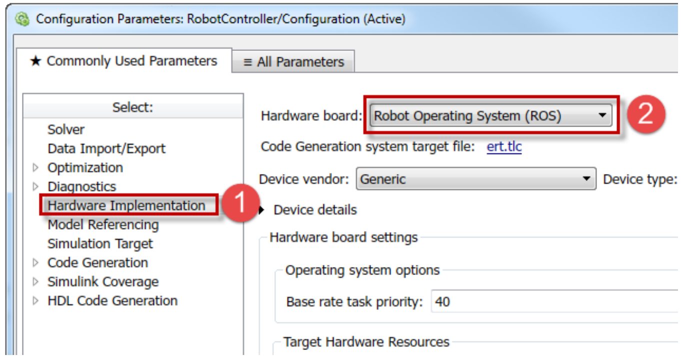


[Exchange Data with ROS Publishers and Subscribers](#)

ROS Toolbox™

R2019b

Generate standalone ROS node



[Generate a Standalone ROS 2 Node from Simulink](#)

ROS Toolbox™

Simulink Coder™

R2019b

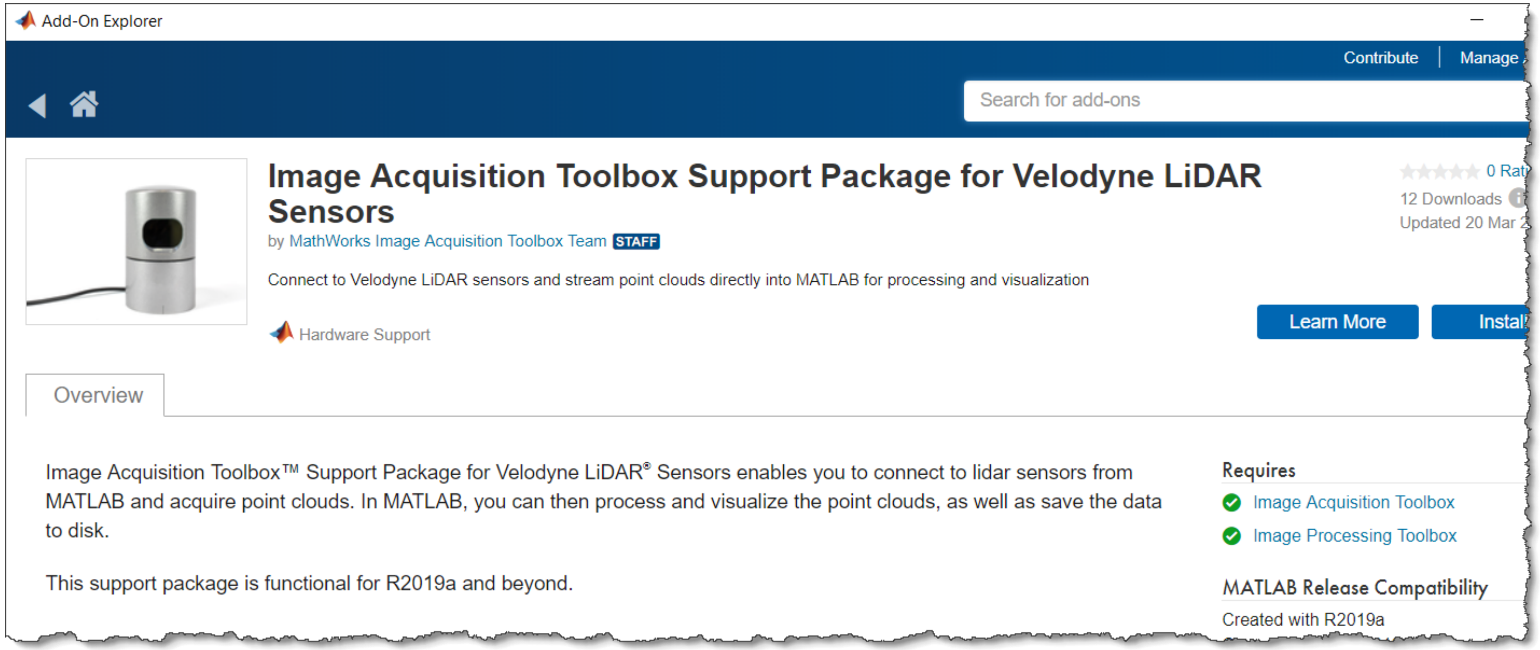
Read point cloud from Velodyne log file

- Read point cloud data from Velodyne packet capture (PCAP) file
- The reader supports the following Velodyne LiDAR models:
  - VLP-16, Puck LITE, Puck Hi-Res, VLP-32C, HDL-32E, and HDL-64E
  - VLS-128 support package is available per request
- User can provide device specific calibration XML file

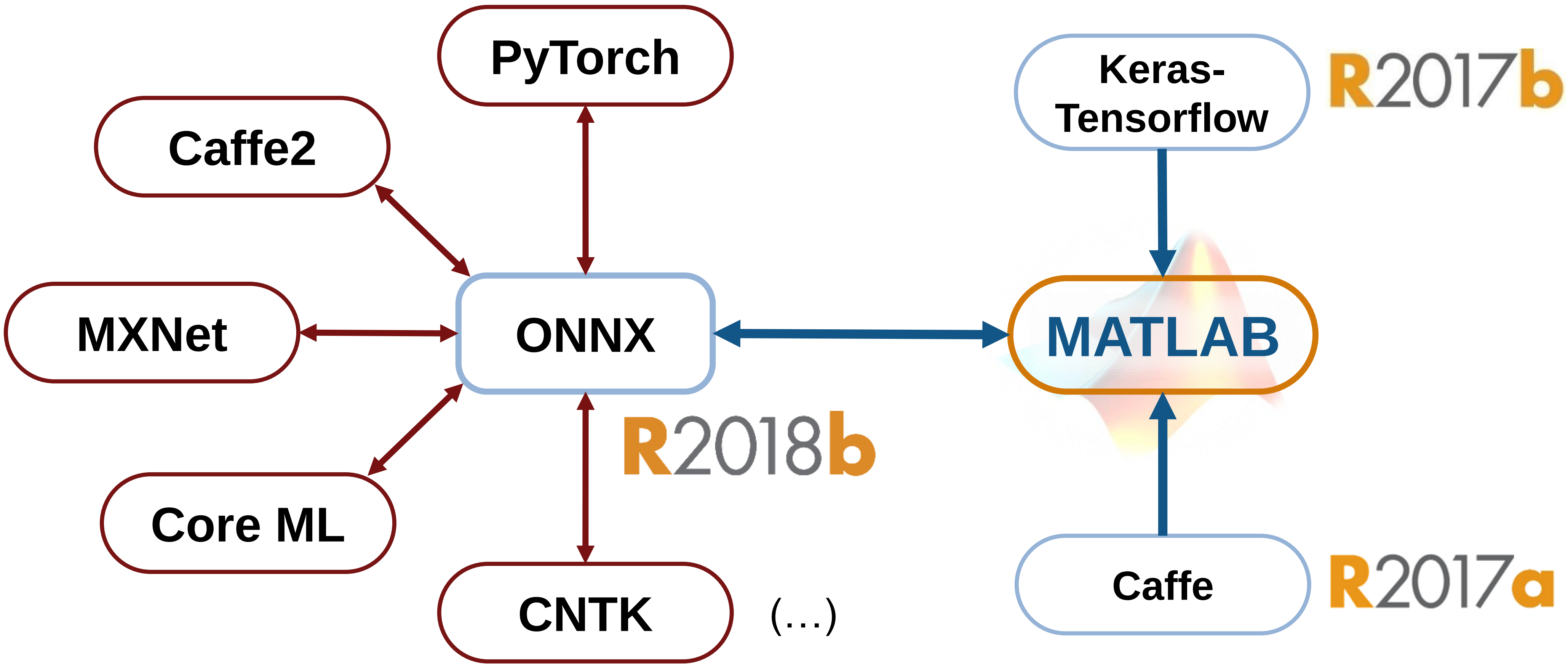
```
veloReader = velodyneFileReader(fileName,deviceModel,'CalibrationFile',calibFile);
```

Stream live Velodyne point cloud data

- You can connect to and stream point clouds from the following Velodyne LiDAR models:
  - HDL-32E sensor
  - VLP-32C Ultra Puck sensor
  - VLP-16 Puck sensor
  - VLP-16 Puck Lite sensor
  - VLP-16 Puck Hi-Res sensor



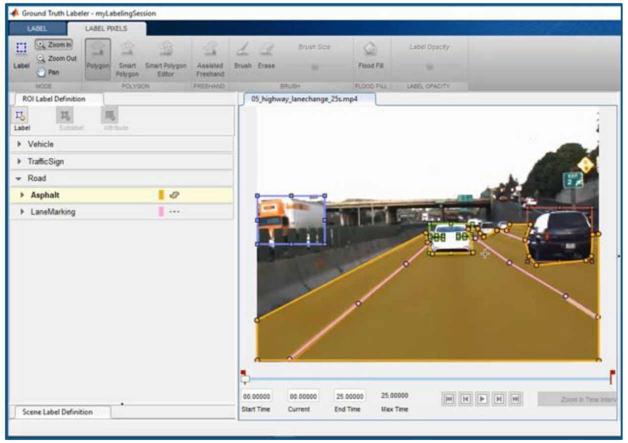
Interoperate with neural network frameworks



Open Neural Network Exchange

Design vision perception systems

Label recorded data

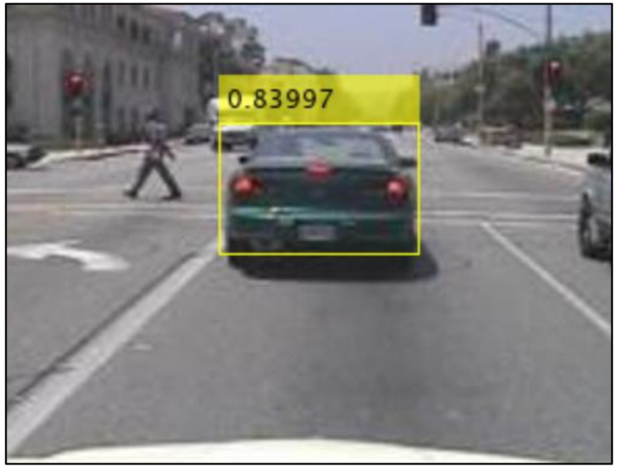


[Get Started with the Ground Truth Labeler]()

Automated Driving ToolboxTM
Computer Vision ToolboxTM

Updated R2019a

Train deep learning networks

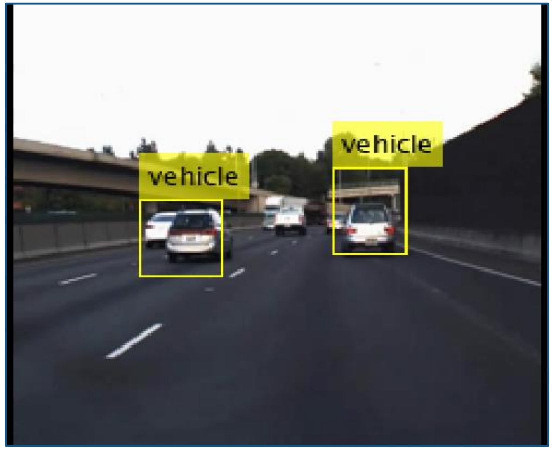


[Object Detection Using YOLO v2 Deep Learning]()

Computer Vision ToolboxTM
Deep Learning ToolboxTM

R2019a

Generate code



[Generate C++ Code for Object Detection Using YOLO v2 and Intel MKL-DNN]()

Deep Learning ToolboxTM

MATLAB Coder

R2019a

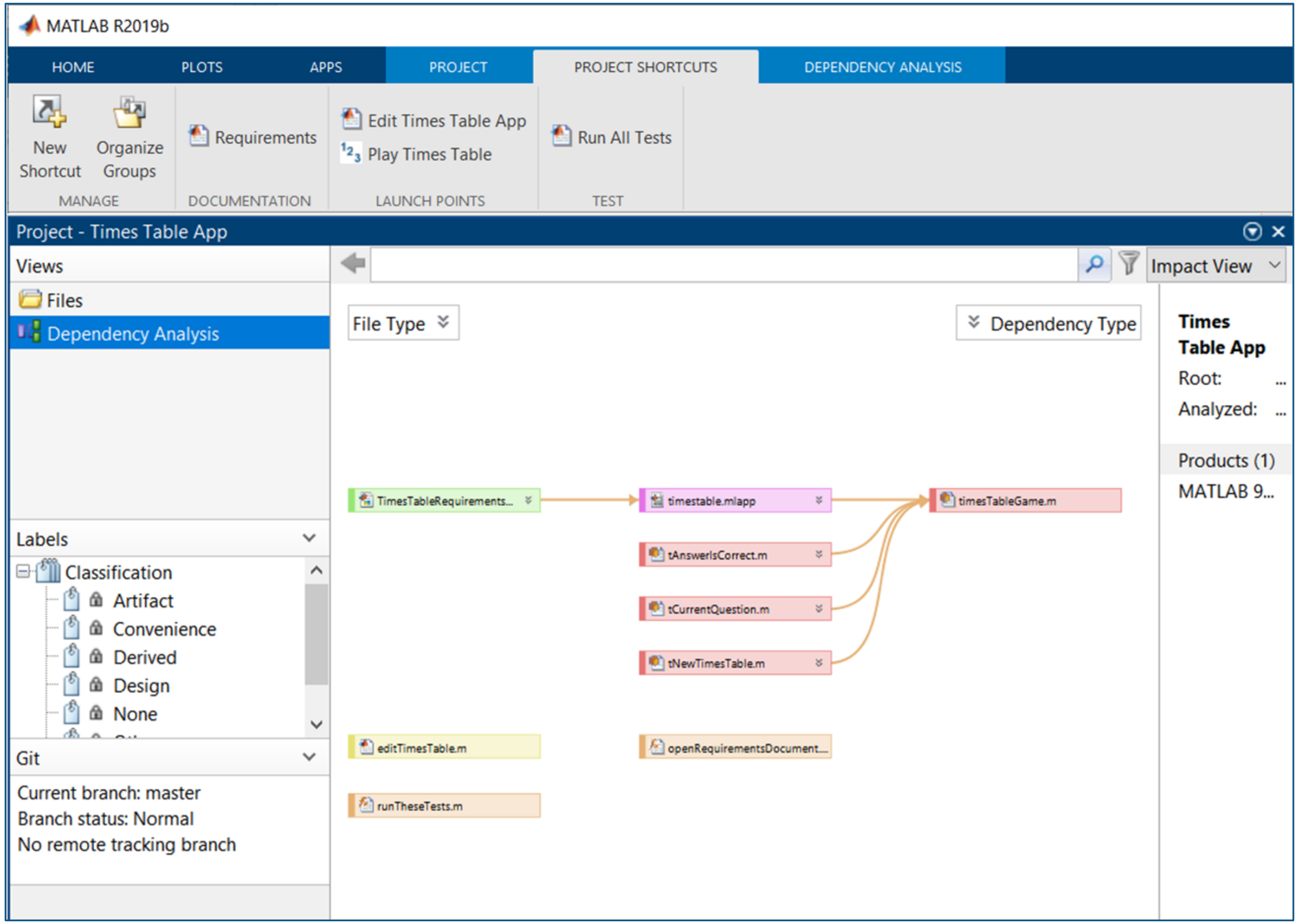
Organize your work and collaborate with projects

[Explore an example project]()

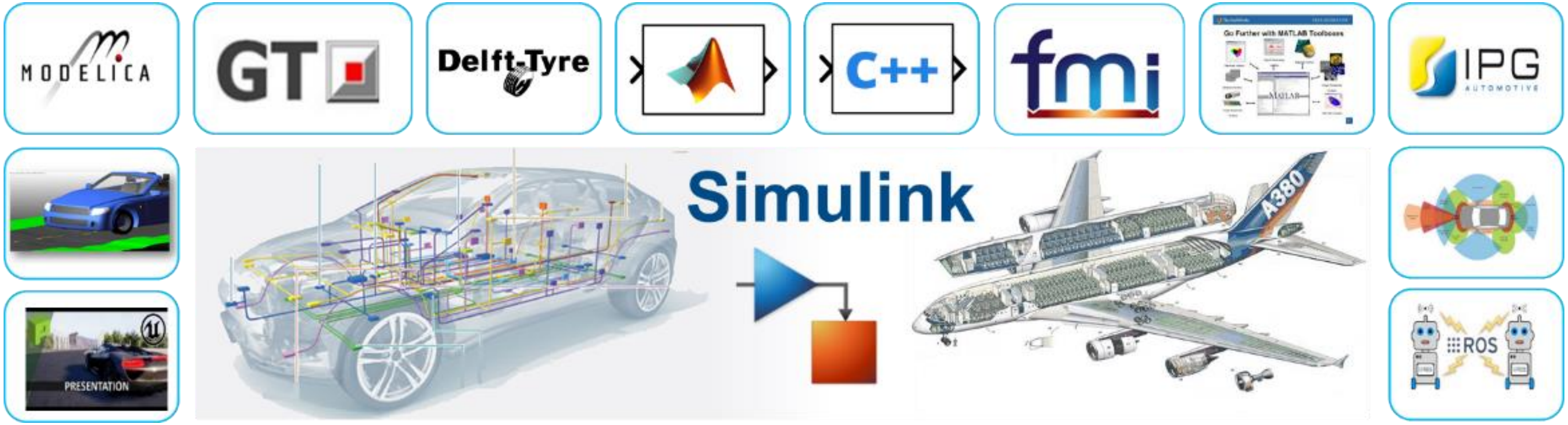
- View, search, and sort files
- Run frequently used files
- Integrate with source control
- Review changes
- Analyze dependencies
- Commit modified files

MATLAB®

R2019a



Connect to third party tools



152 Interfaces to 3rd Party Modeling and Simulation Tools
(as of March 2019)

MathWorks
Partner

MathWorks Simulation Challenge

for SAE AutoDrive year 3 competition

#	Task	Points
1	Synthesize data to test open loop perception algorithm	10
2	Synthesize data to test closed loop controls algorithm	10
3	Generate code from controls algorithm	10
4	Innovate	15
5	**Reflect** a. Would you do something different next time? b. Is there anything missing from the tools that would have helped you?	5

MathWorks Simulation Challenge

for SAE AutoDrive year 3 competition

#	Task	Points
1	Synthesize data to test open loop perception algorithm	10
2	Synthesize data to test closed loop controls algorithm	10
3	Generate code from controls algorithm	10
4	Innovate	15
5	Reflect	5

Additional clarification of tasks and scoring will be provided at November training