MATLAB EXPO 2019

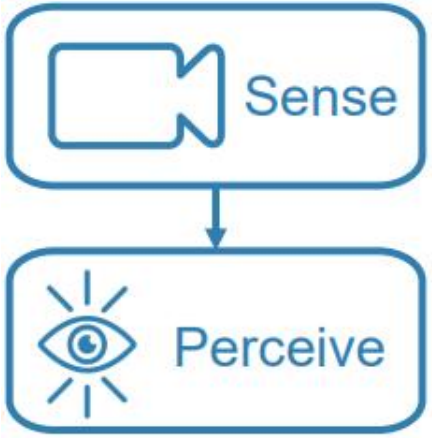
Automated Driving with MATLAB and Simulink

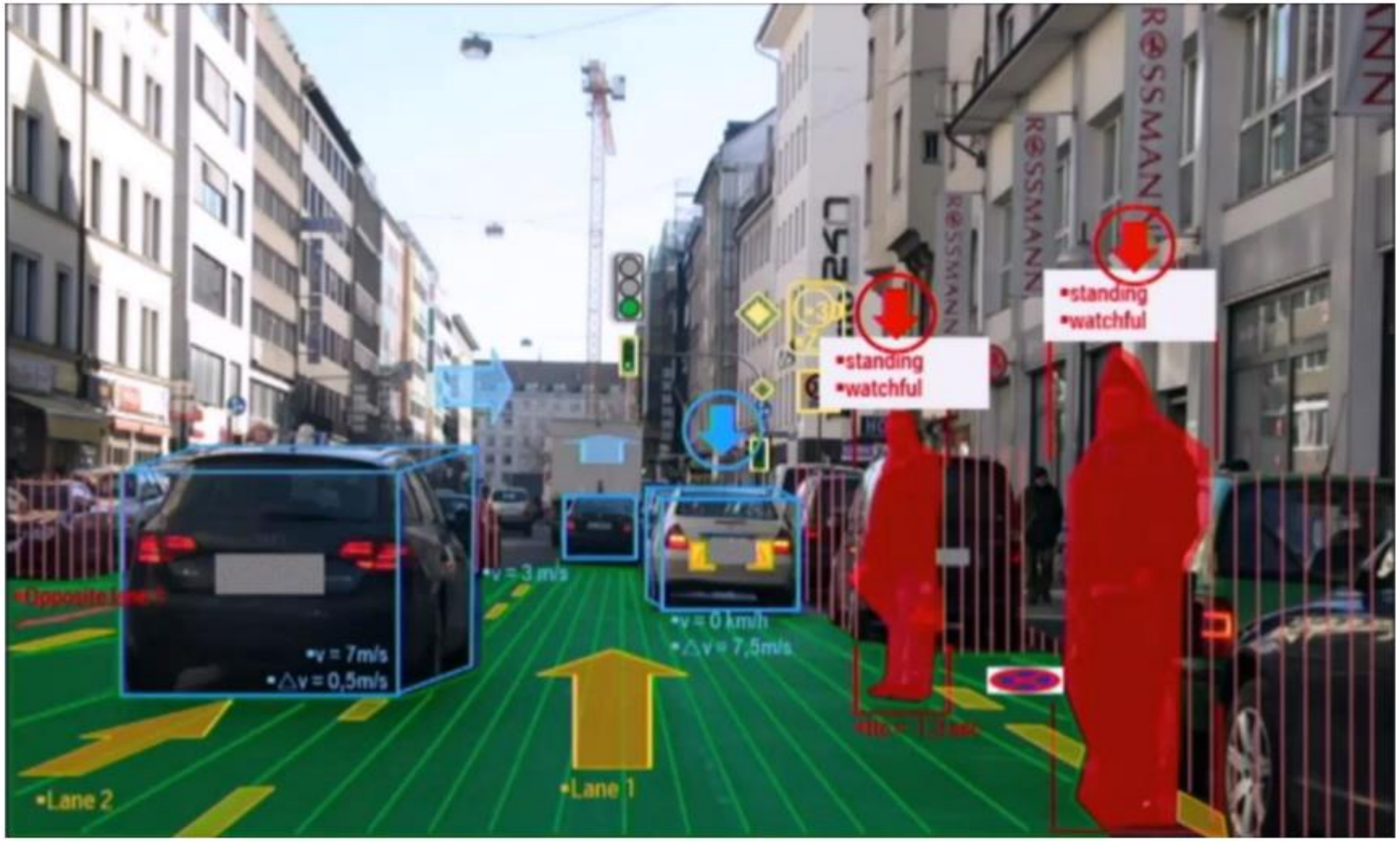
GianCarlo Pacitti
Senior Application Engineer, MathWorks

Capabilities of an Autonomous Vehicle

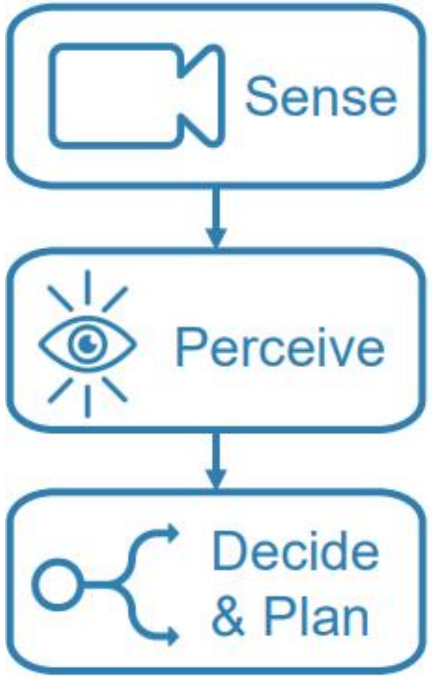
Sense

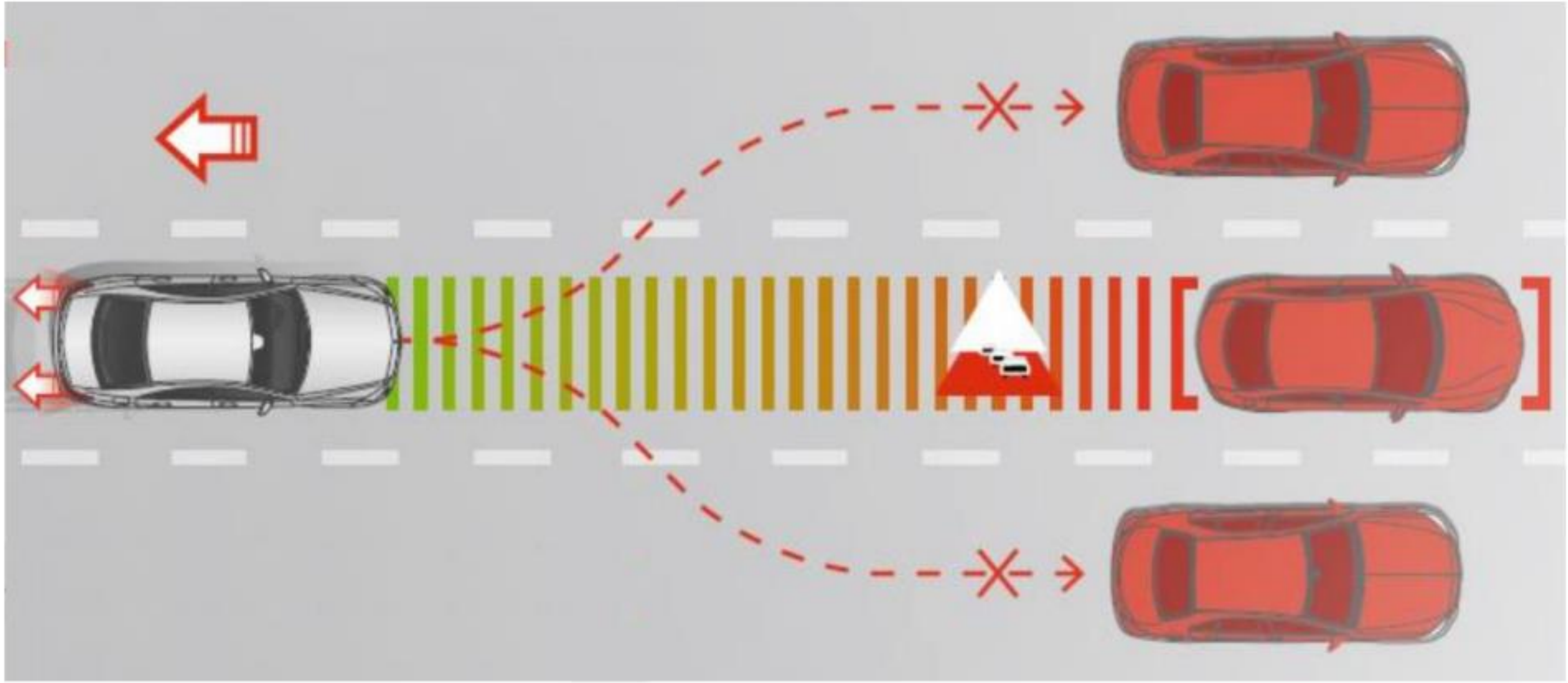
Capabilities of an Autonomous Vehicle



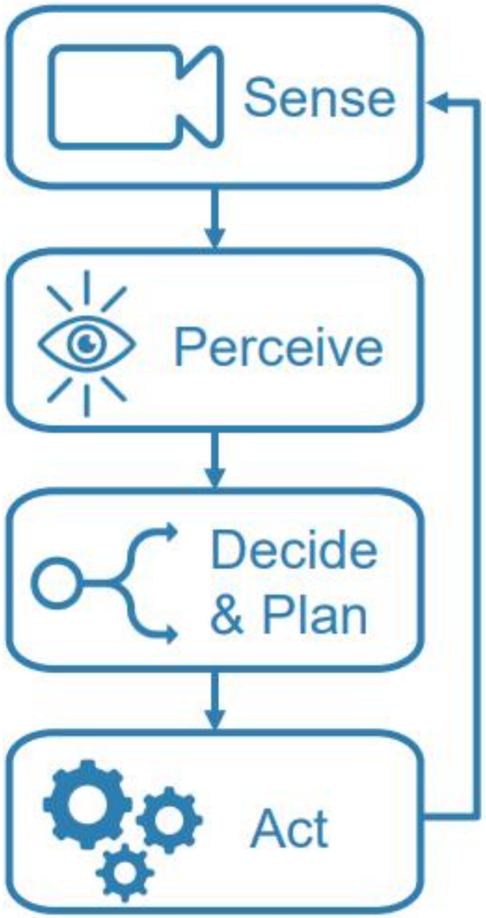


Capabilities of an Autonomous Vehicle

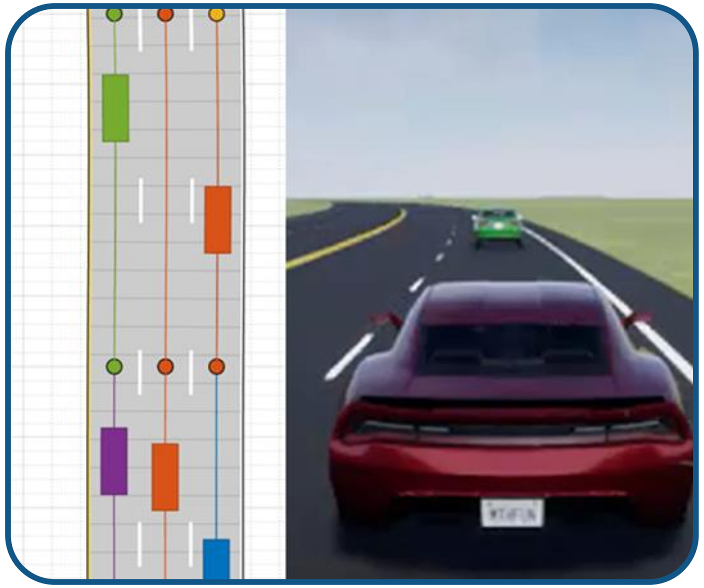




Capabilities of an Autonomous Vehicle



Some common questions from automated driving engineers



How can I
synthesize scenarios
to test my designs?

Perception

Planning

Control

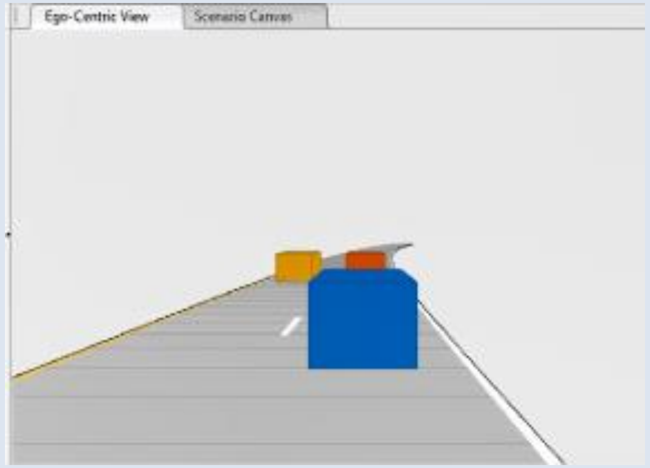
How can I
discover and design
in multiple domains?

Simulation Integration

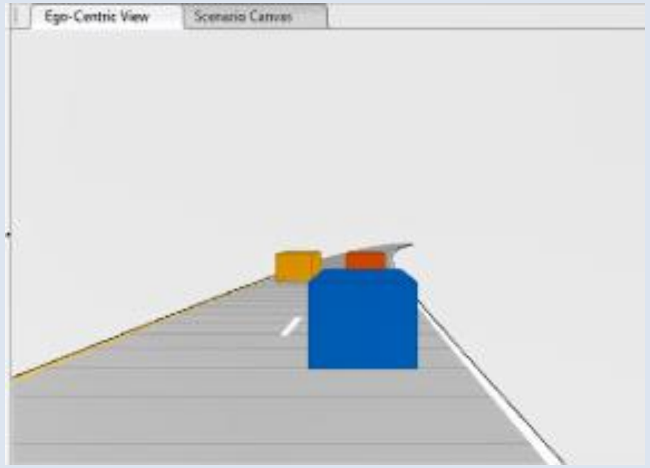
ROS
CAN
C/C++
Python
Cross Release
Third Party

How can I
integrate
with other environments?

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**
	 	
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)

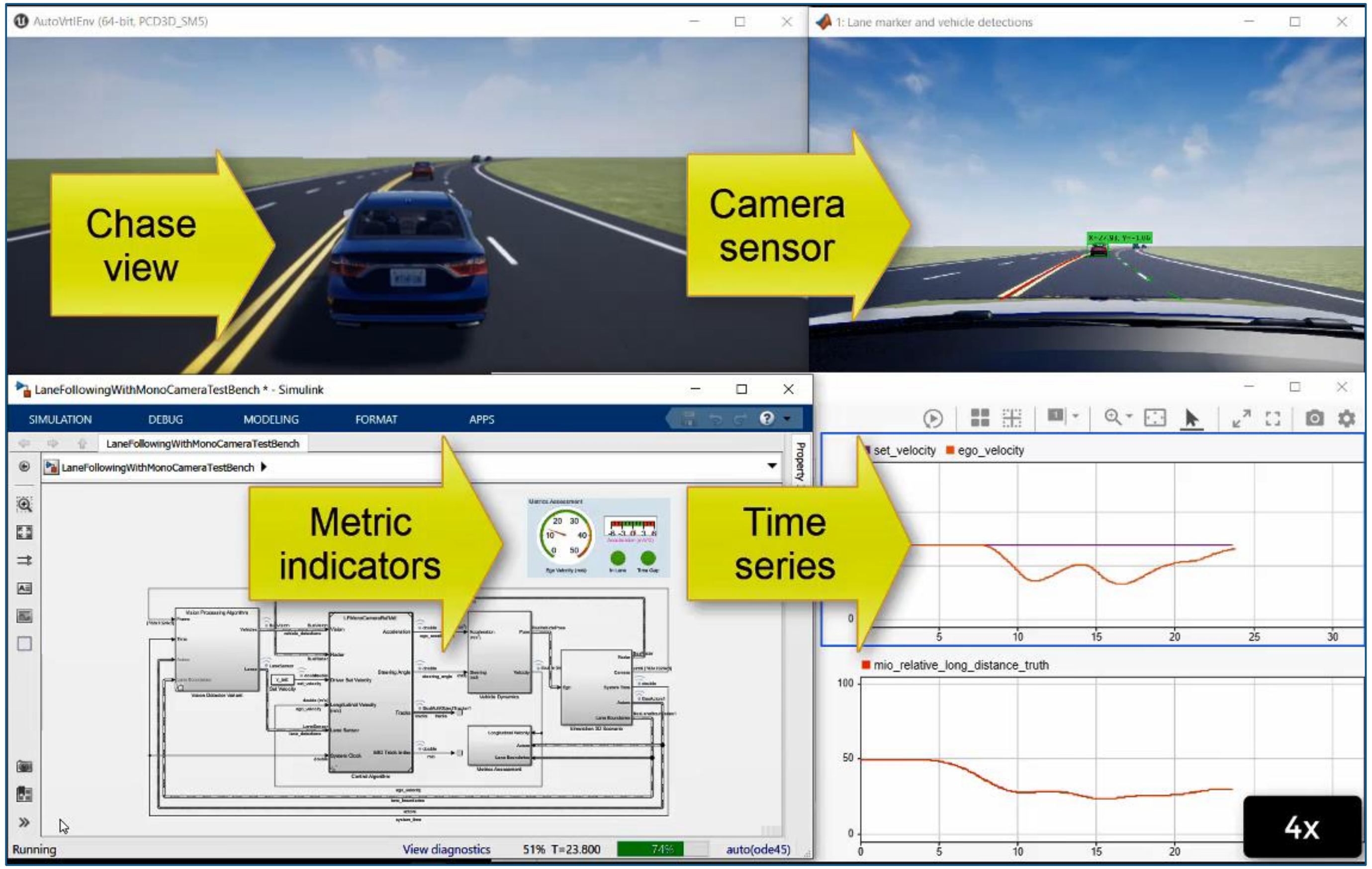
Simulate controls with perception

[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

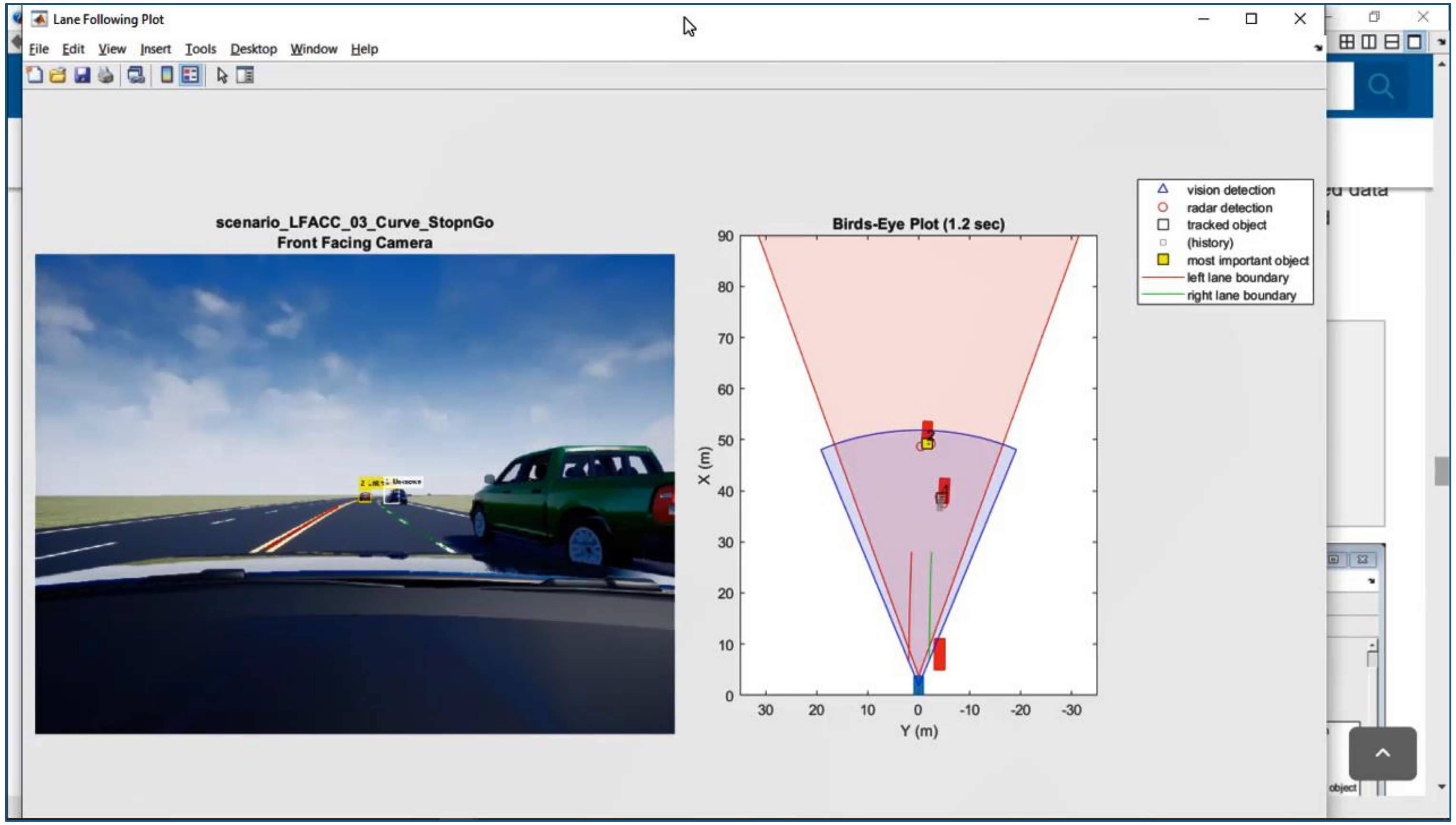


Visit the Demo Station to see more…

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



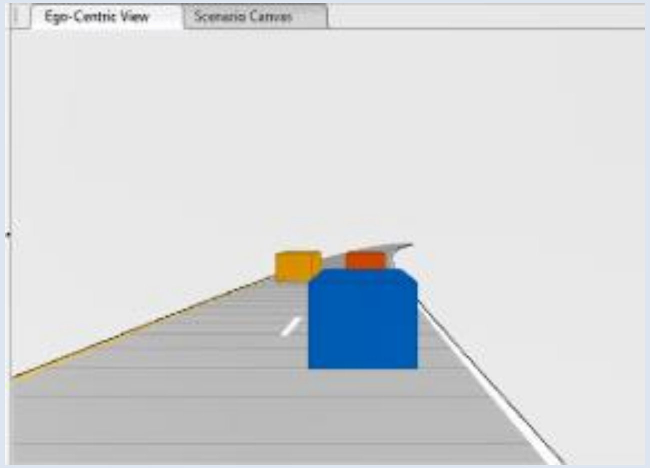
Model Predictive Control Toolbox™

Automated Driving Toolbox™

Vehicle Dynamics Blockset™

Updated R2019b

How can I design with virtual driving scenarios?

Scenes	**Cuboid**	**3D Simulation**
		
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)

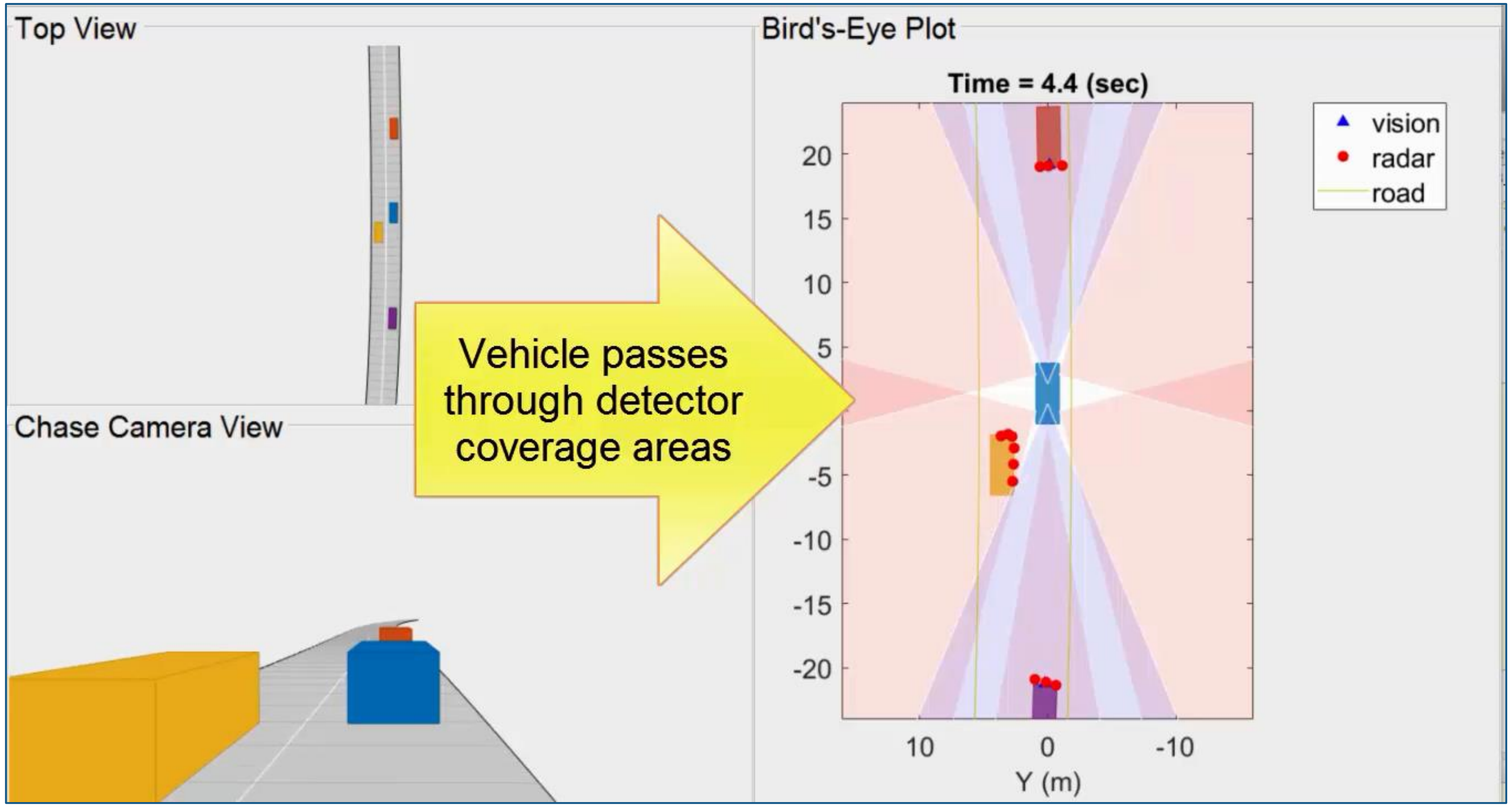
Synthesize driving scenarios to test sensor fusion algorithms

Sensor Fusion Using Synthetic Radar and Vision Data

- Create scenario
- Add probabilistic radar and vision sensors
- Create tracker
- Visualize coverage area, detections, and tracks

Automated Driving Toolbox™

R2017a



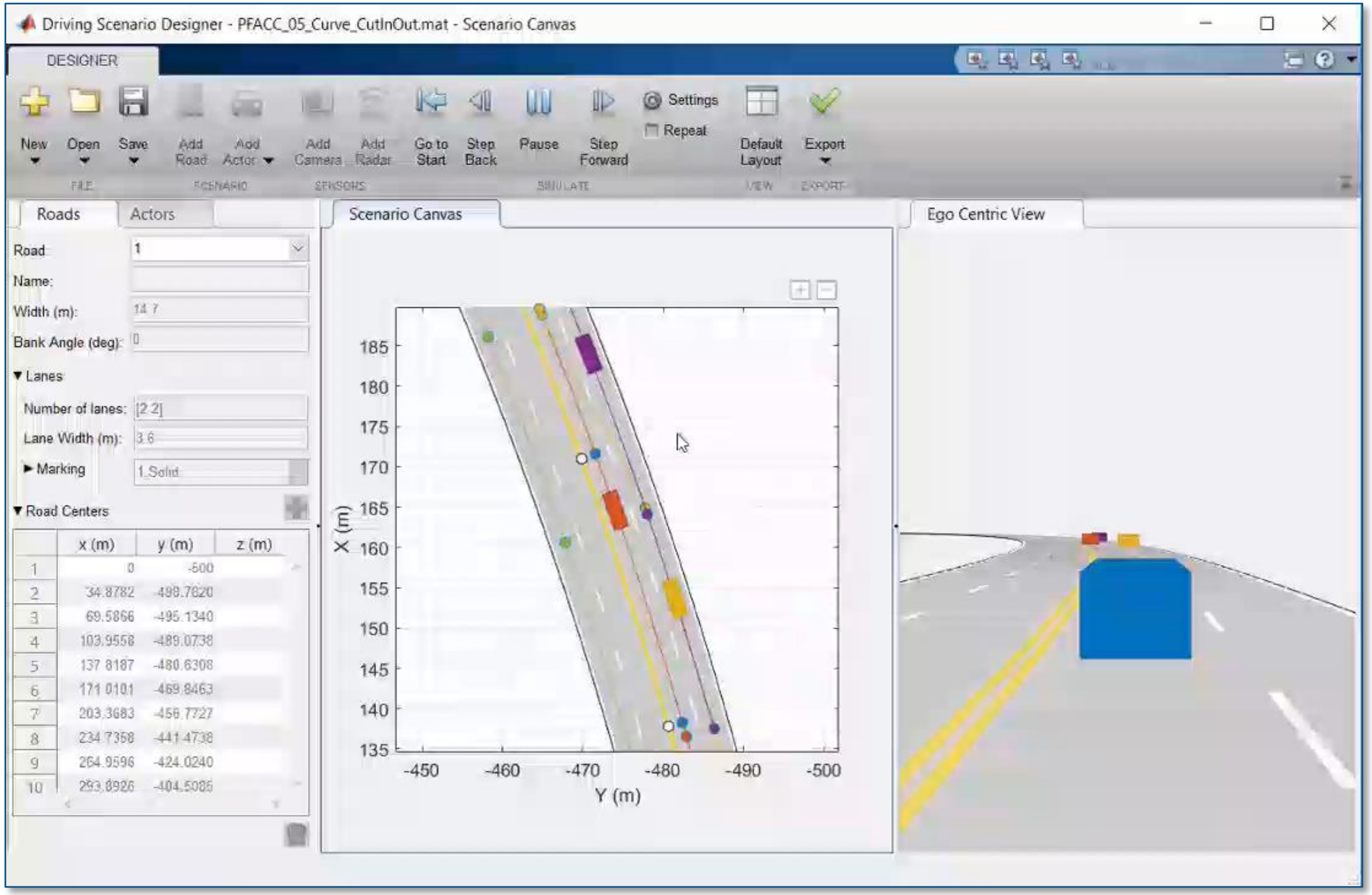
Graphically author driving scenarios

[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

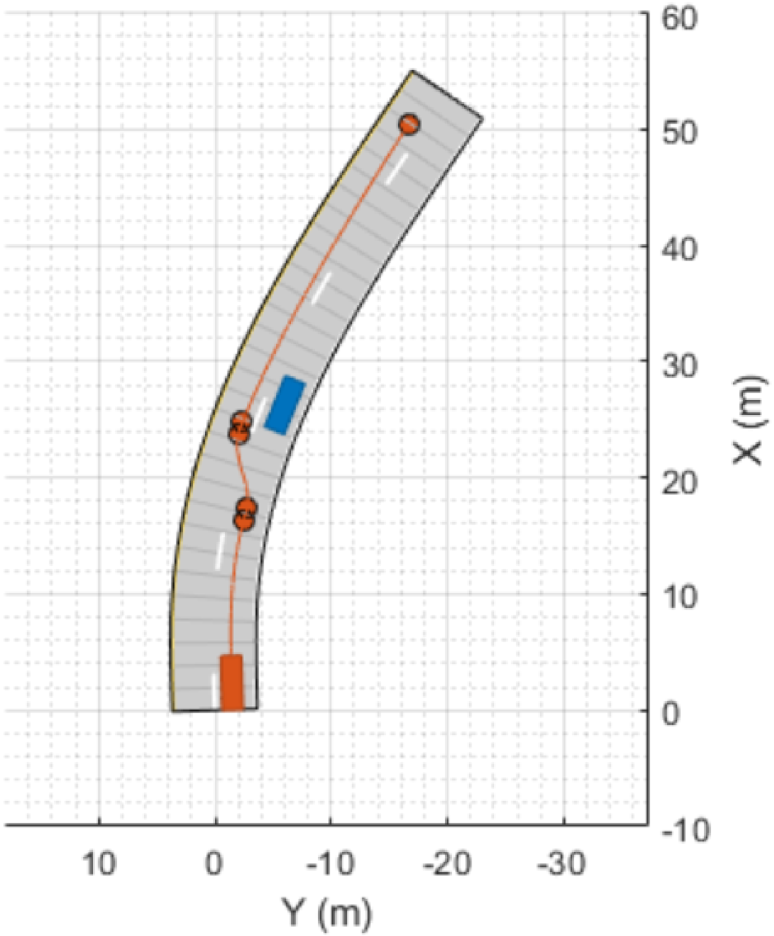
Automated Driving Toolbox™

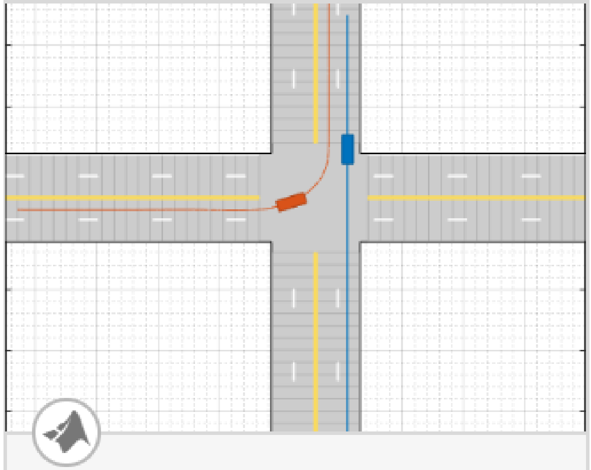
R2018a



Programmatically author driving scenarios

```
scenario = drivingScenario;

road( scenario, [0 0; 10 0; 53 -20], ...
      'lanes', lanespec(2) );

plot( scenario,'Waypoints','on' );

idleCar = vehicle( scenario, ...
                   'Position',[25 -5.5 0], ...
                   'Yaw',-22 );

passingCar = vehicle( scenario, 'ClassID', 1 );

waypoints = [1 -1.5; 16.36 -2.5; 17.35 -2.765; ...
             23.83 -2.01; 24.9 -2.4; 50.5 -16.7];

velocity = 15;

trajectory( passingCar, waypoints, velocity );
```





Create Driving Scenario Variations Programmatically

Programmatically create variations of a driving scenario that was built using the Driving Scenario Designer app.

Open Live Script

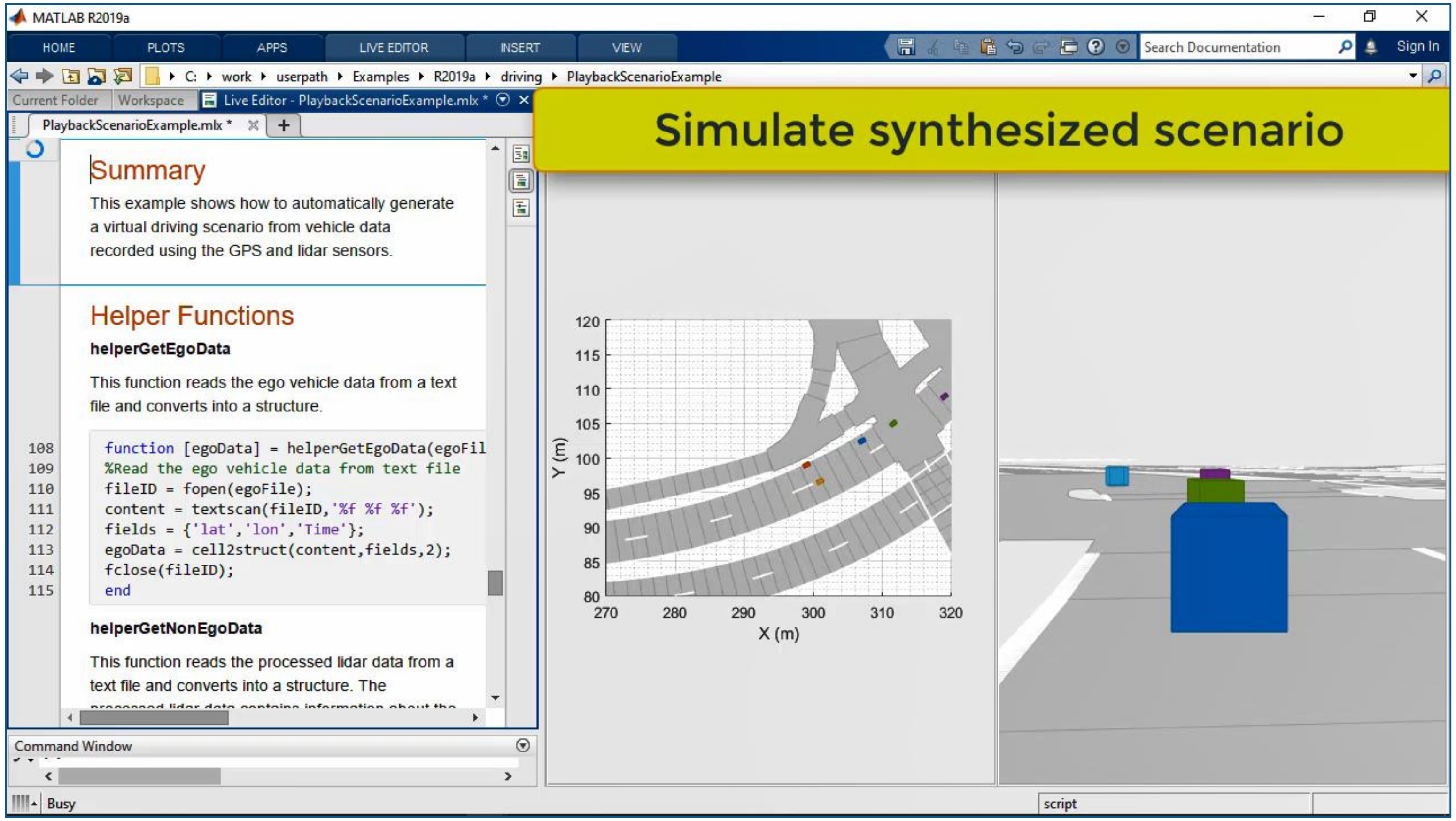
Synthesize driving scenarios from recorded data

[Scenario Generation from Recorded Vehicle Data]

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

Automated Driving Toolbox™

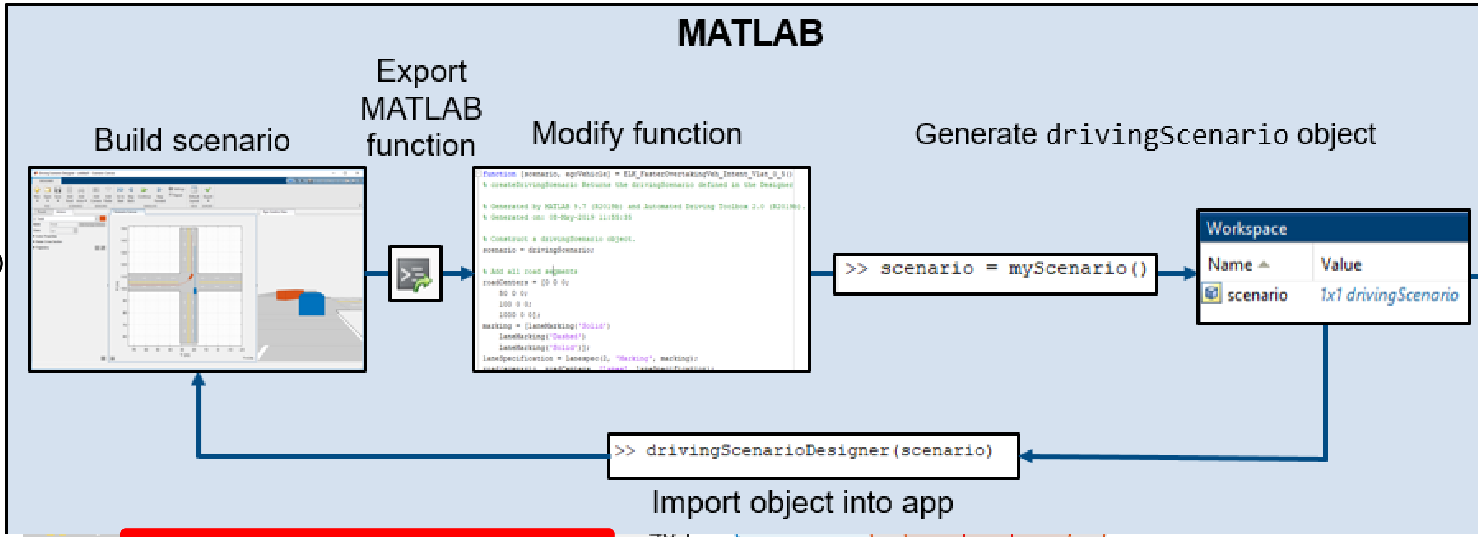
R2019a

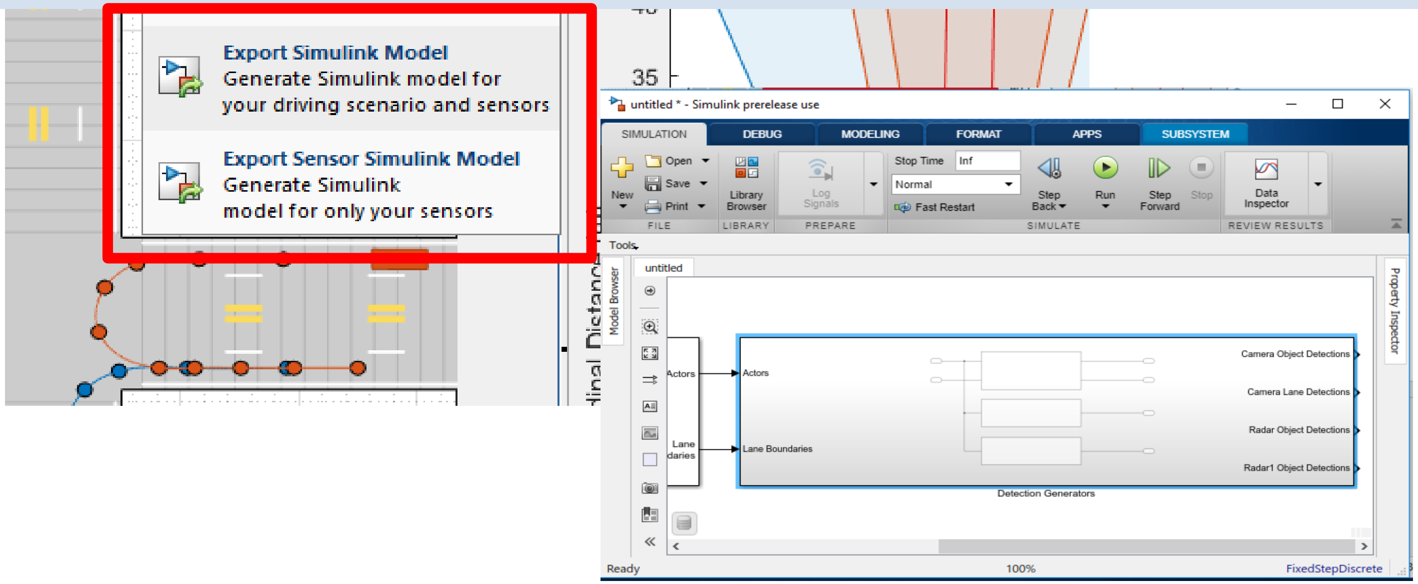


Enhancements to driving scenarios

[Create Driving Scenario Variations Programmatically]()

- Export the scenario code to MATLAB® and generate scenario variations programmatically
- Export the scenario and sensors to Simulink® and use them to test your driving algorithms.





Automated Driving Toolbox™

R2019b

Integrate driving scenario into closed loop simulation

[Lane Following Control with Sensor Fusion]()

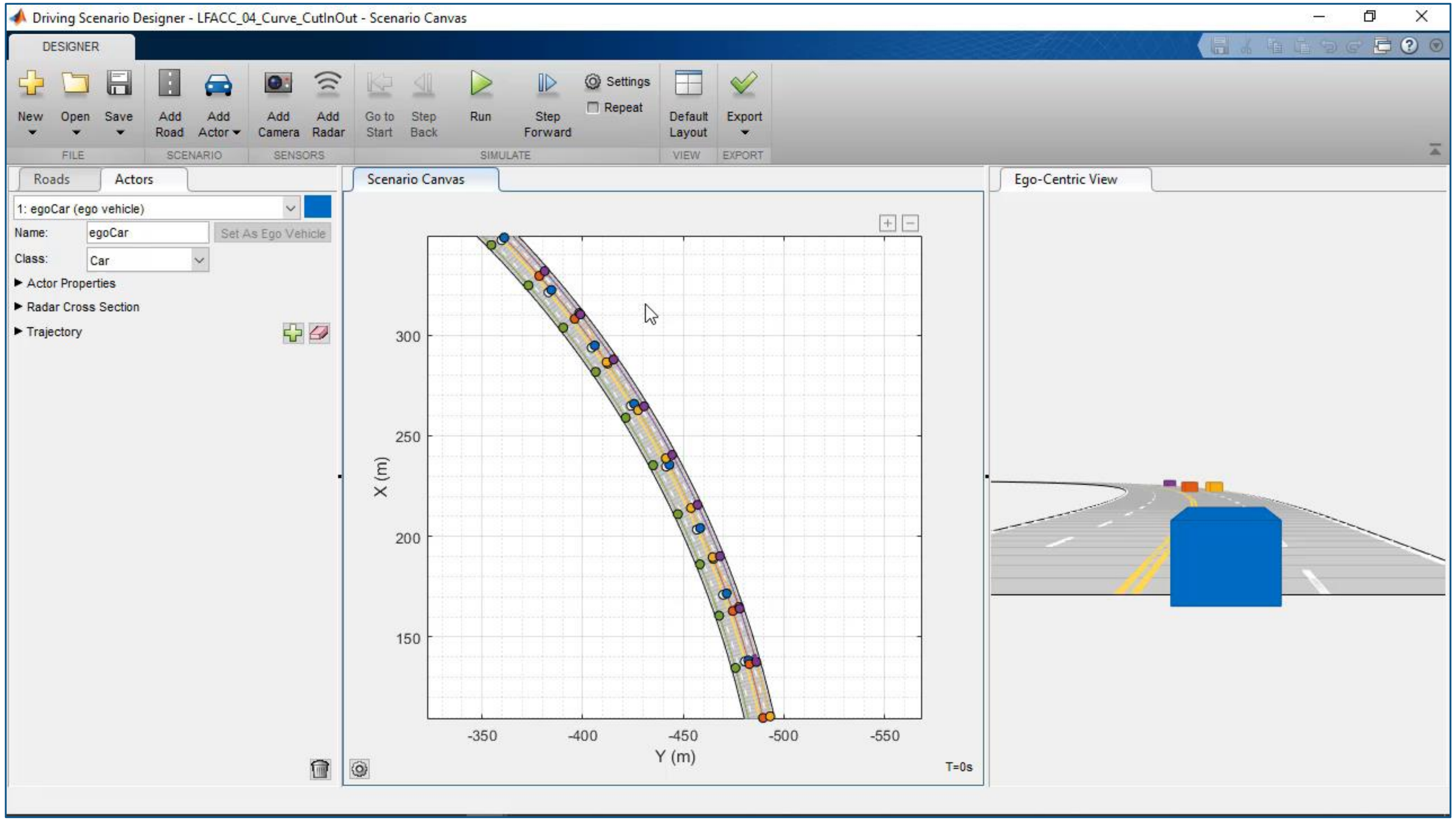
- Integrate scenario into system
- Design lateral (lane keeping) and longitudinal (lane spacing) model predictive controllers
- Visualize sensors and tracks
- Generate C/C++ code
- Test with software in the loop (SIL) simulation

Model Predictive Control Toolbox™

Automated Driving Toolbox™

Embedded Coder®

R2018b



Design lateral and longitudinal controls

Lane Following Control with Sensor Fusion

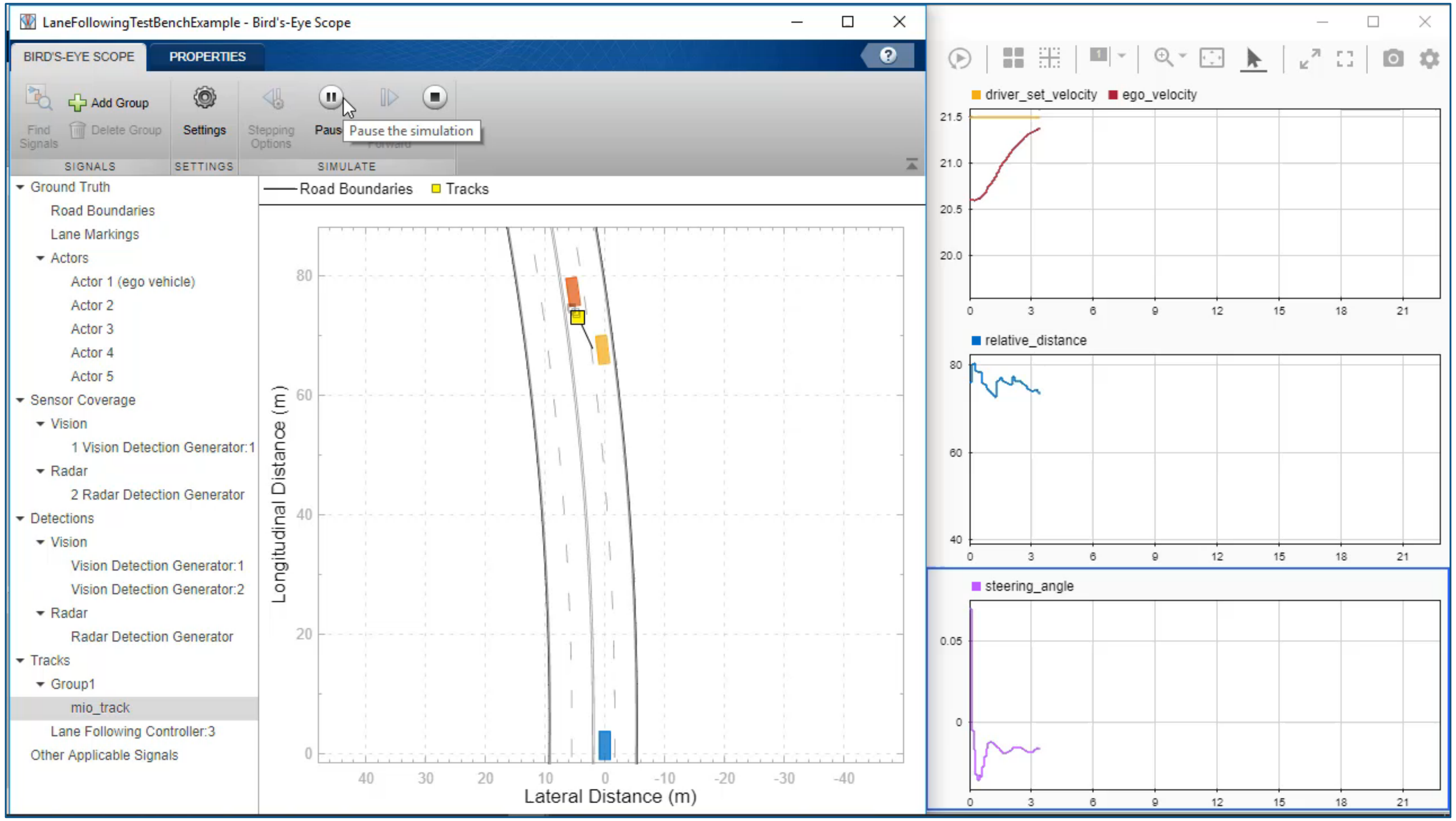
- Integrate scenario into system
- Design lateral (lane keeping) and longitudinal (lane spacing) model predictive controllers
- Visualize sensors and tracks
- Generate C/C++ code
- Test with software in the loop (SIL) simulation

Model Predictive Control Toolbox™

Automated Driving Toolbox™

Embedded Coder®

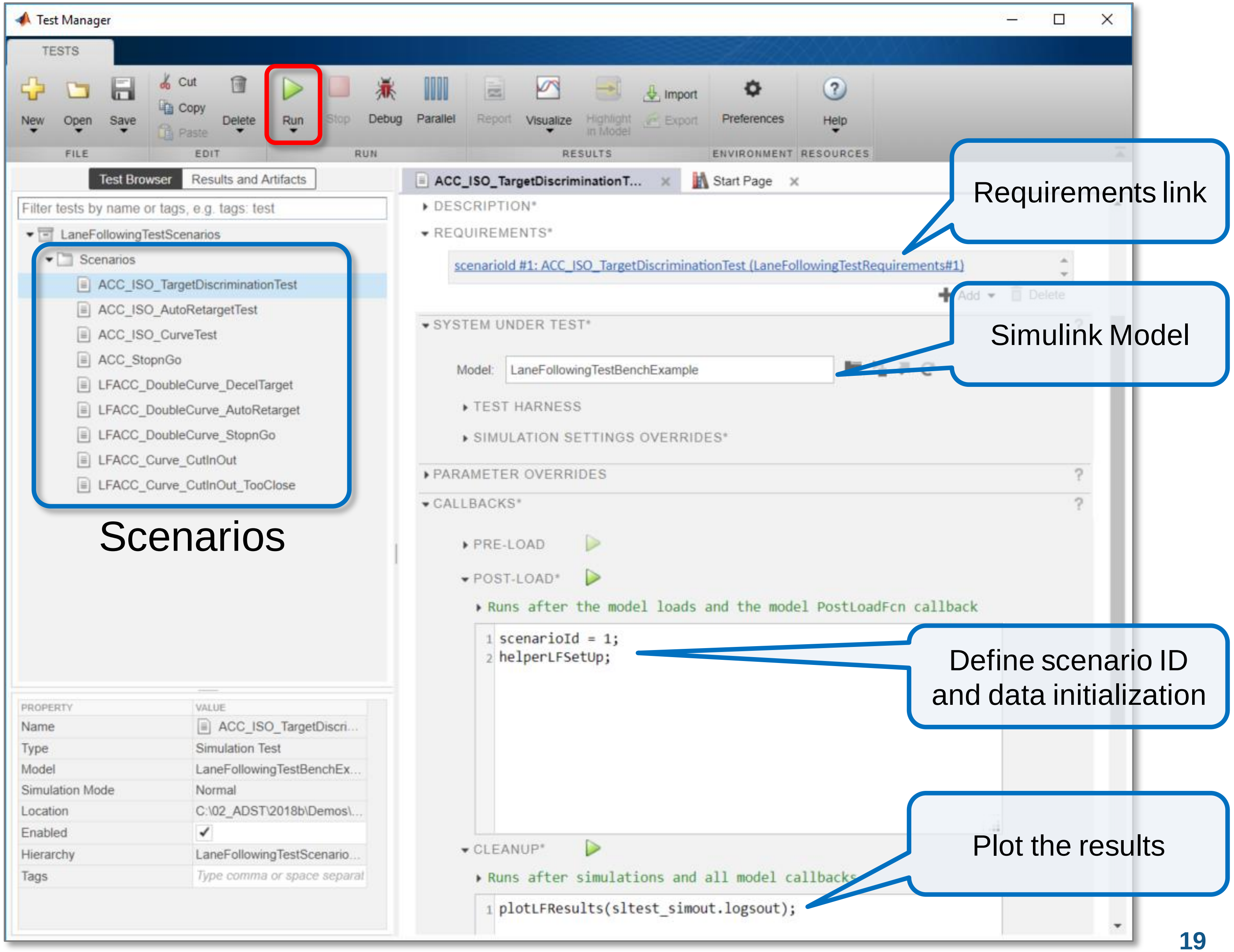
R2018b



Automate testing against driving scenarios

[Testing a Lane Following Controller with Simulink Test]

- Specify driving scenario



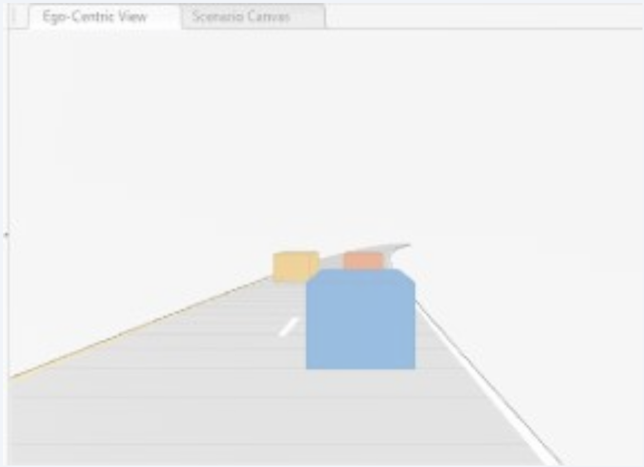
Simulink Test™

Automated Driving Toolbox™

Model Predictive Control Toolbox™

R2018b

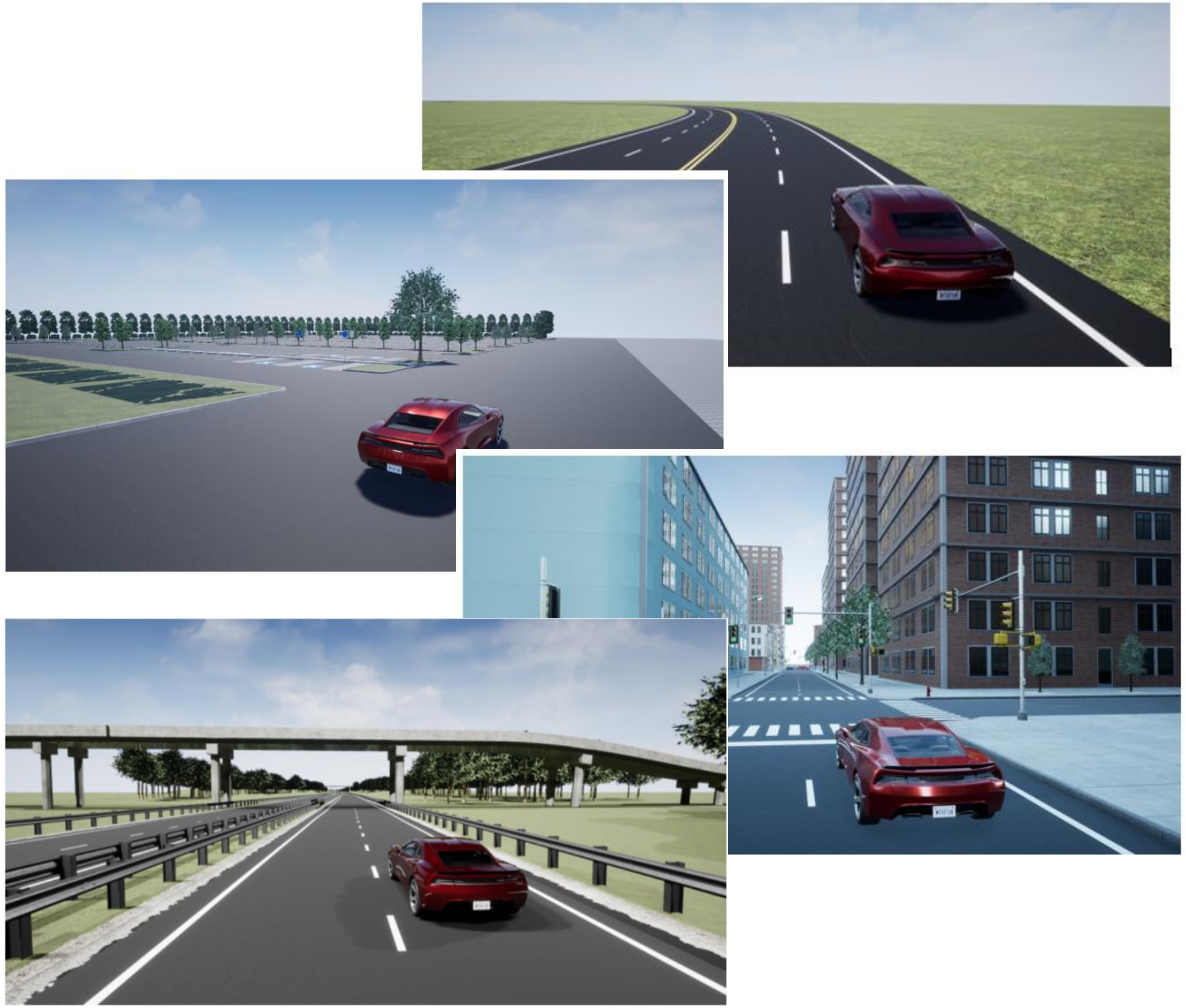
How can I design with virtual driving scenarios?

Scenes	**Cuboid**	**3D Simulation**
		
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



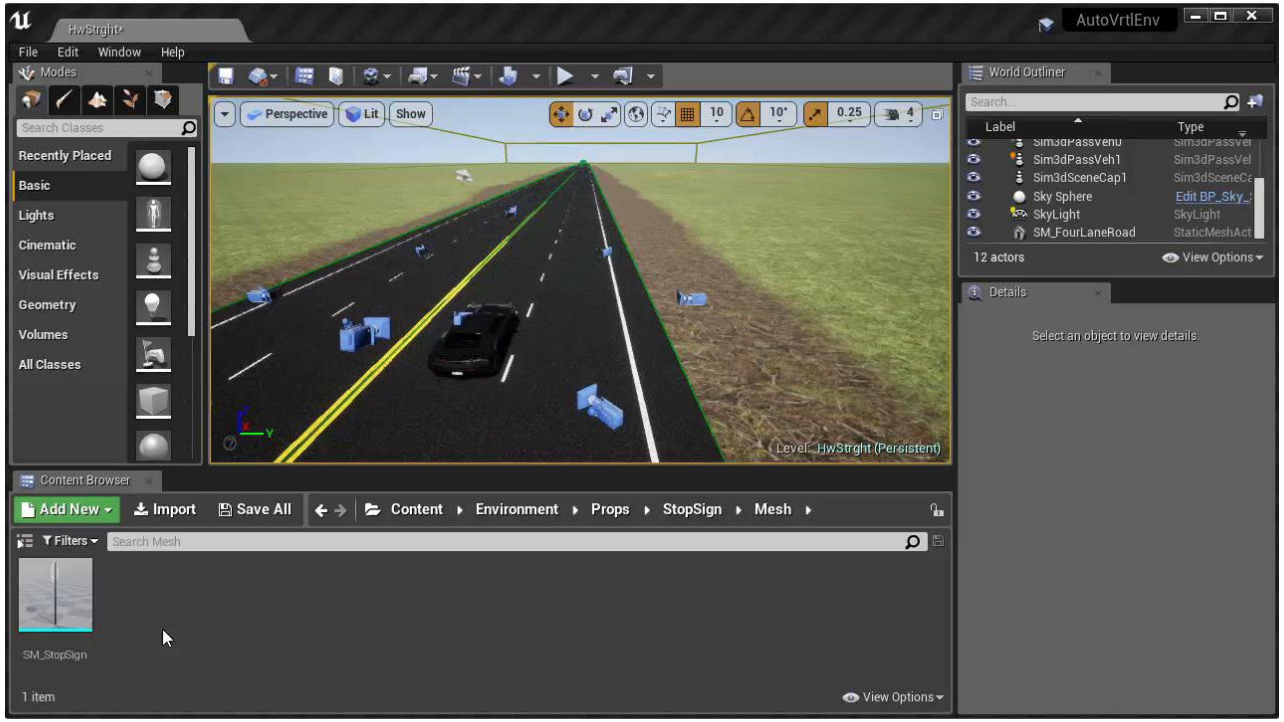
Automated Driving Toolbox™

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



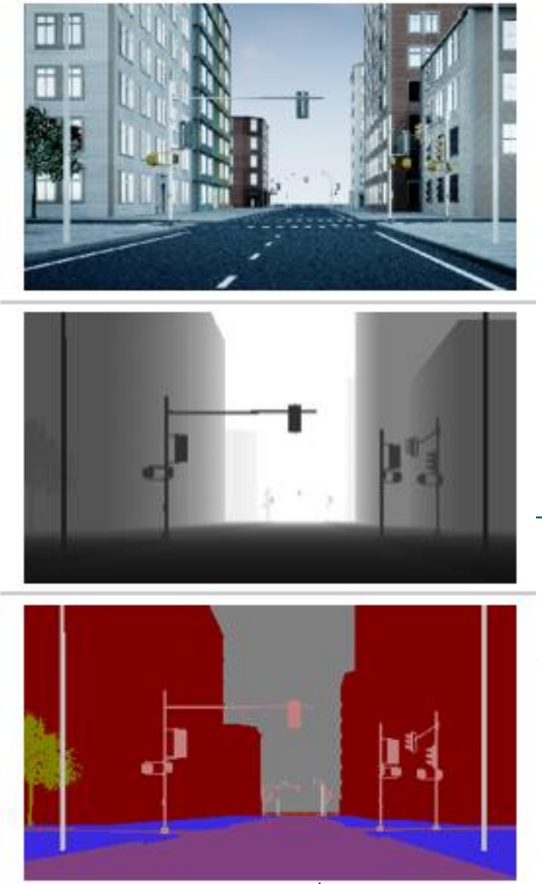
Vehicle Dynamics Blockset™

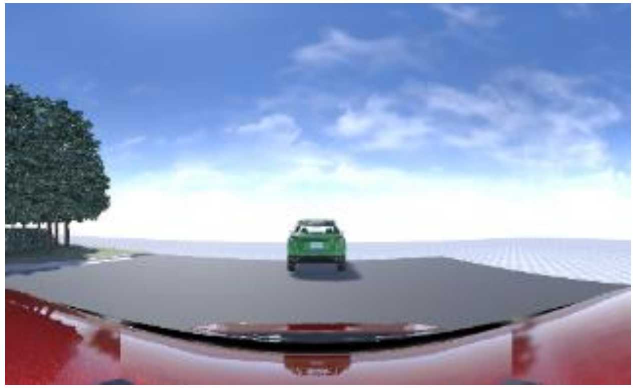
R2019b

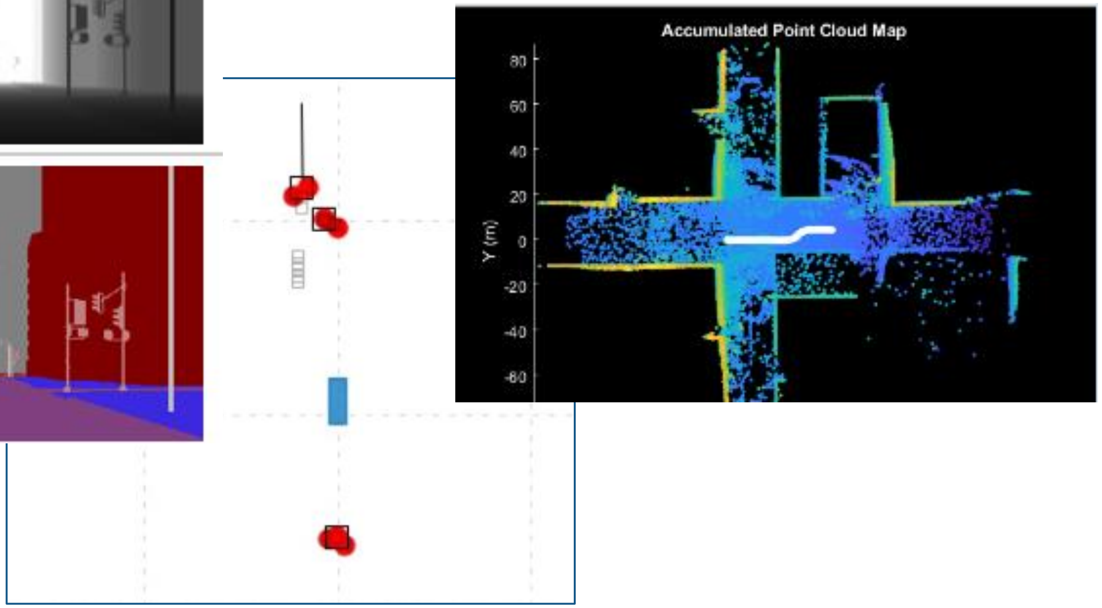
Model sensors in 3D simulation environment

[3D Simulation for Automated Driving](#)

- Monocular camera
- Fisheye camera
- Lidar
- Probabilistic radar







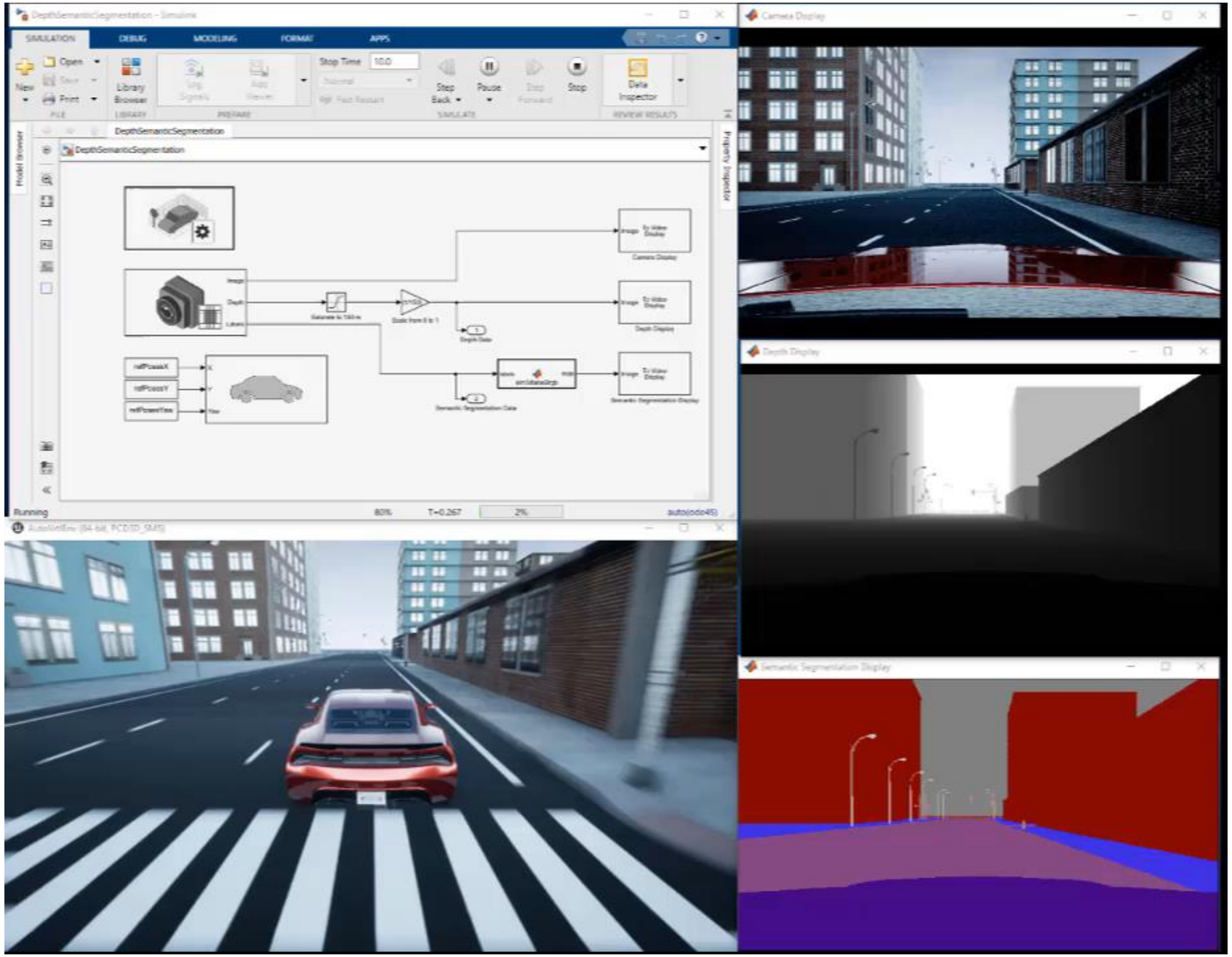
Automated Driving Toolbox™

R2019b

Synthesize monocular camera sensor data

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

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



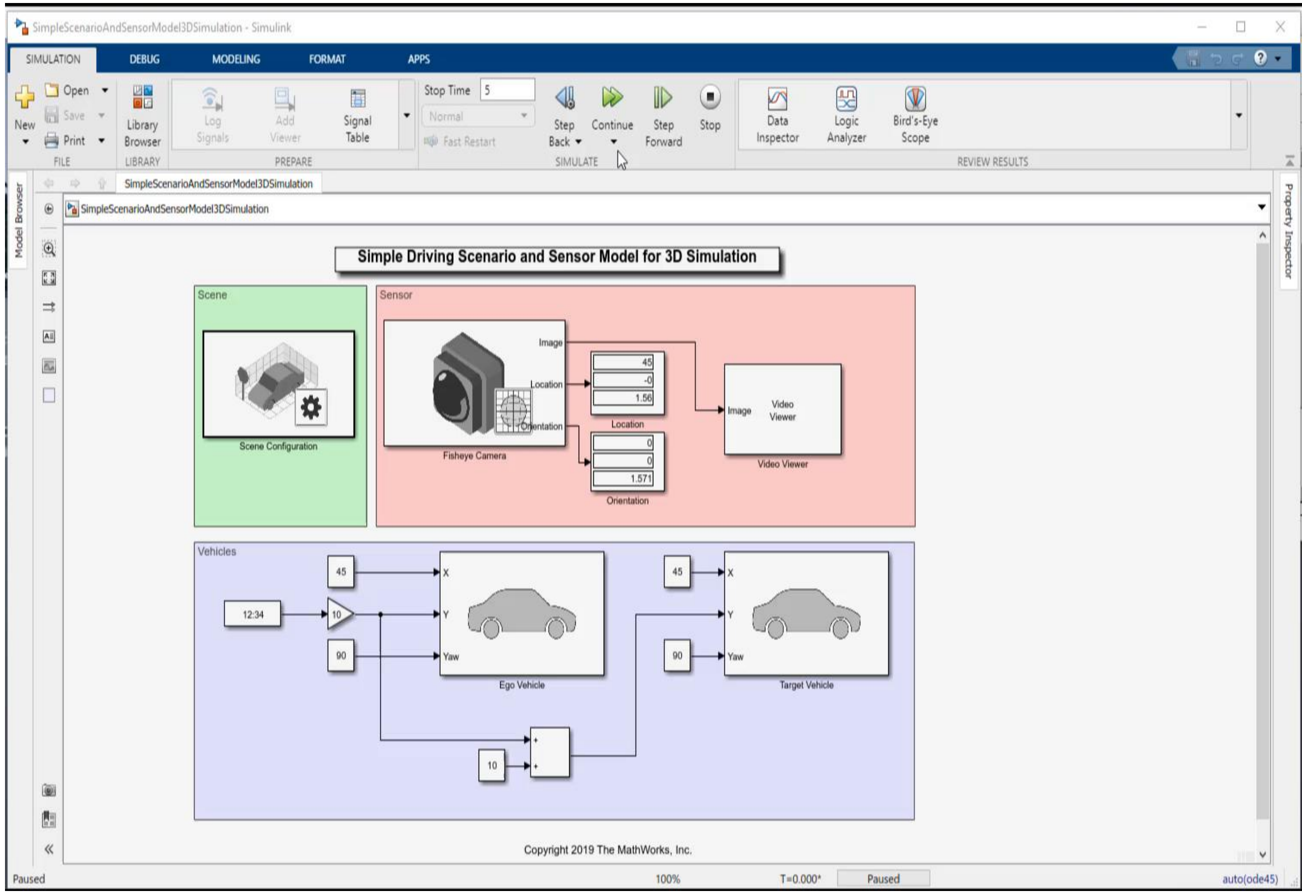
Automated Driving Toolbox™

R2019b

Synthesize fisheye camera sensor data

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

- Scaramuzza camera model
  - parameters for distortion center, image size and mapping coefficients



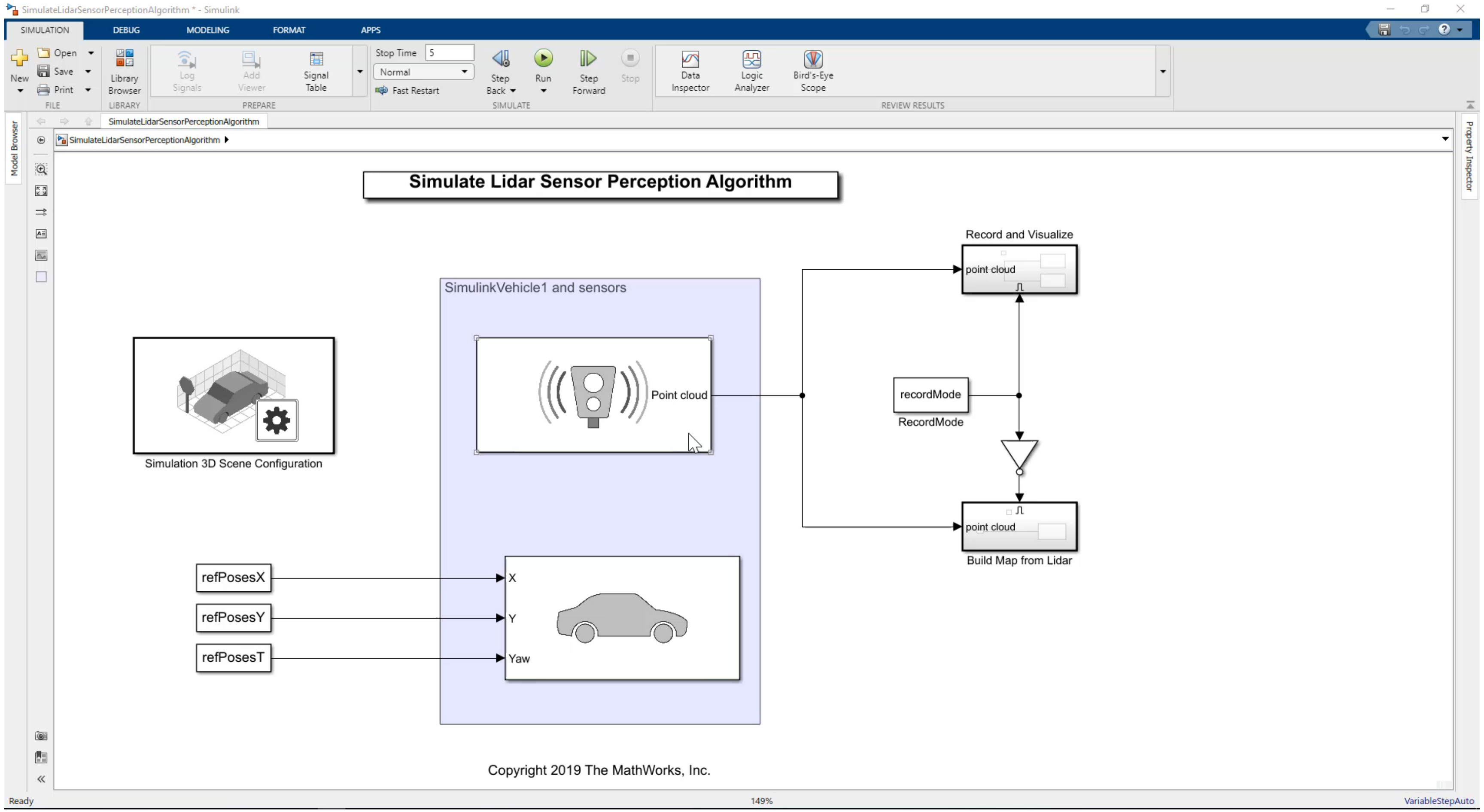
Automated Driving Toolbox™

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

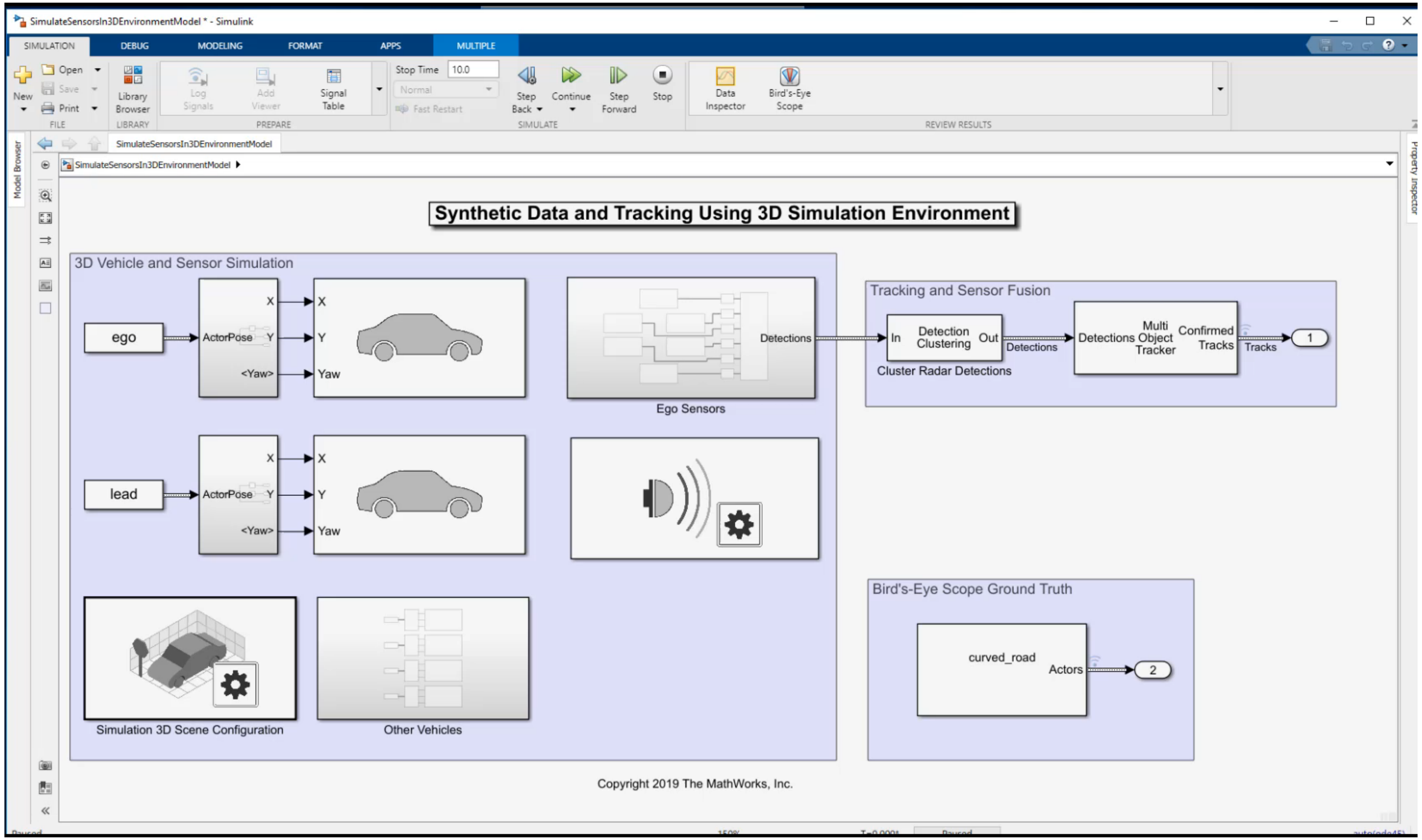
Synthesize radar sensor data

[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

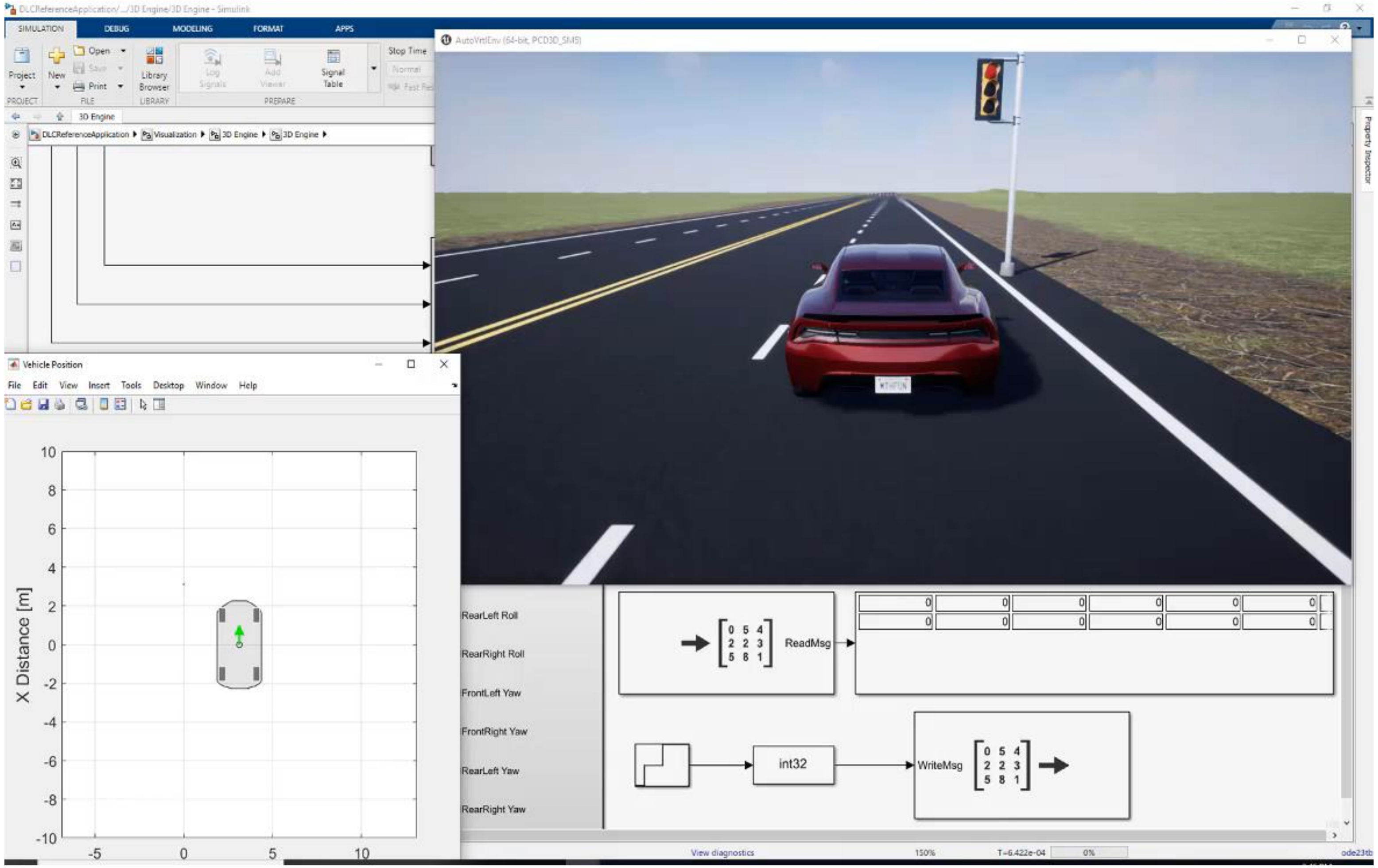
[Send and Receive Double-Lane Change Scene Data]

- 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 Blockset™

R2019b



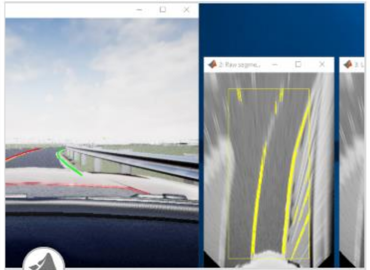
New Examples for 3D Simulation in Automated Driving Toolbox

Unreal Engine Driving Scenario Simulation

Select Waypoints for 3D Simulation

Select waypoints from a scene and visualize the path of a vehicle following these waypoints in a 3D simulation environment.

Open Script



Design of Lane Marker Detector in 3D Simulation Environment

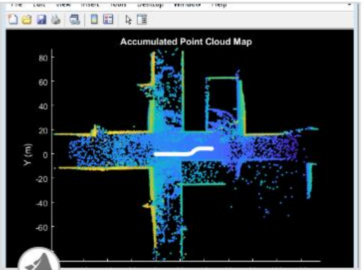
Use a 3D simulation environment to record synthetic sensor data and develop and test a lane marker detection system.

Open Script

Visualize Automated Parking Valet Using 3D Simulation

Visualize vehicle motion in a 3D simulation environment using an automated parking valet system constructed in Simulink.

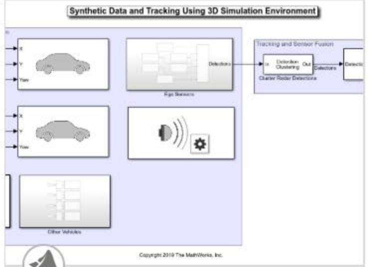
Open Script



Simulate Lidar Sensor Perception Algorithm

Develop a lidar perception algorithm using data recorded from a 3D simulation environment, and simulate within that environment.

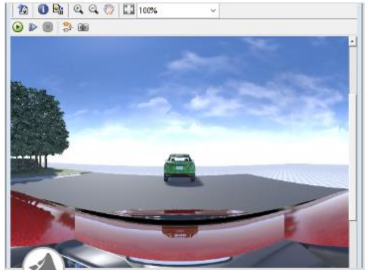
Open Script



Simulate Radar Sensors in 3D Environment

Implement a synthetic data simulation for tracking and sensor fusion using Simulink and a 3D simulation environment.

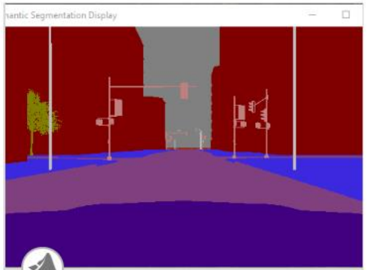
Open Model



Simulate a Simple Driving Scenario and Sensor in 3D Environment

Learn the basics of configuring and simulating scenes, vehicles, and sensors in a 3D environment powered by the Unreal Engine from

Open Model

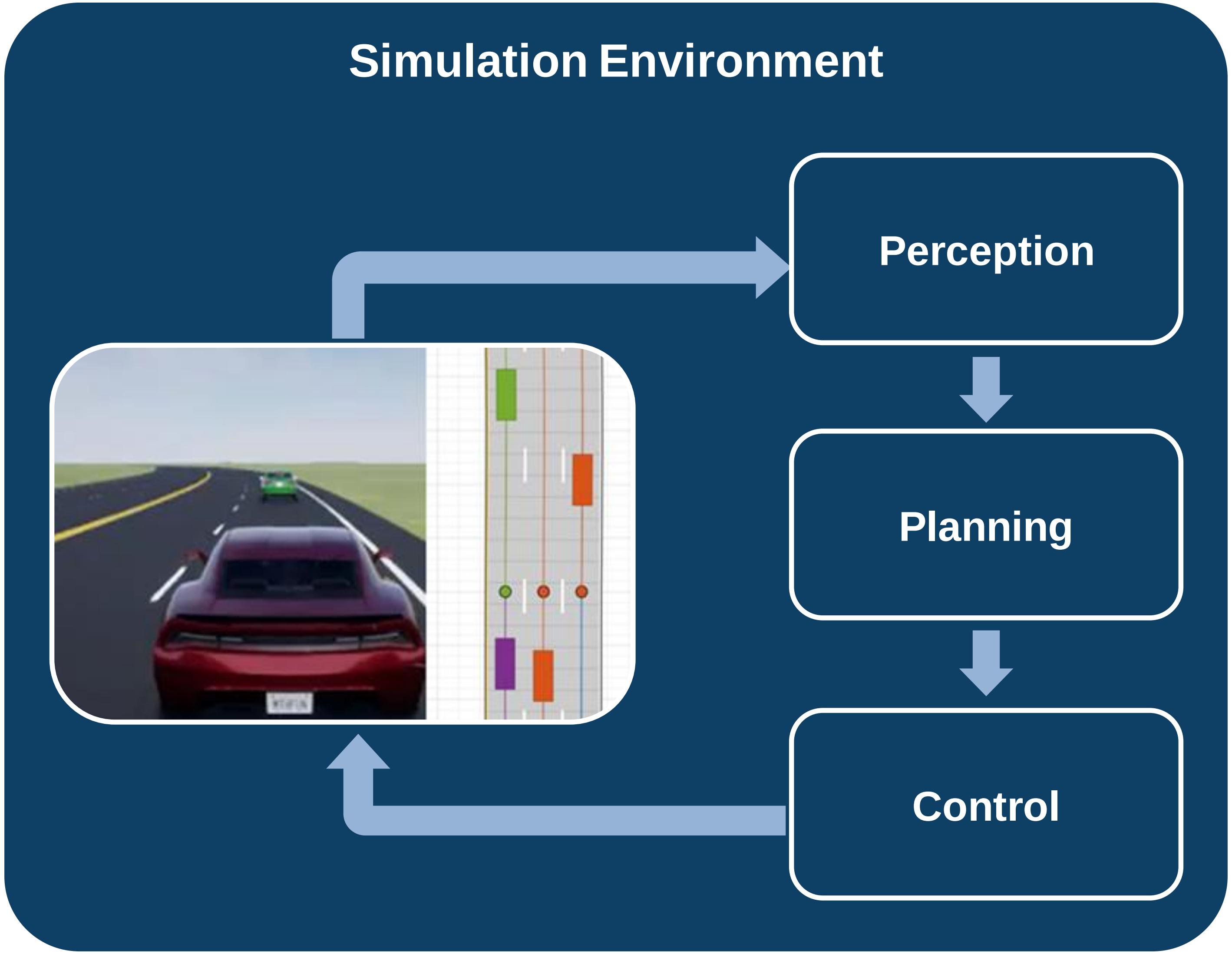


Visualize Depth and Semantic Segmentation Data in 3D Environment

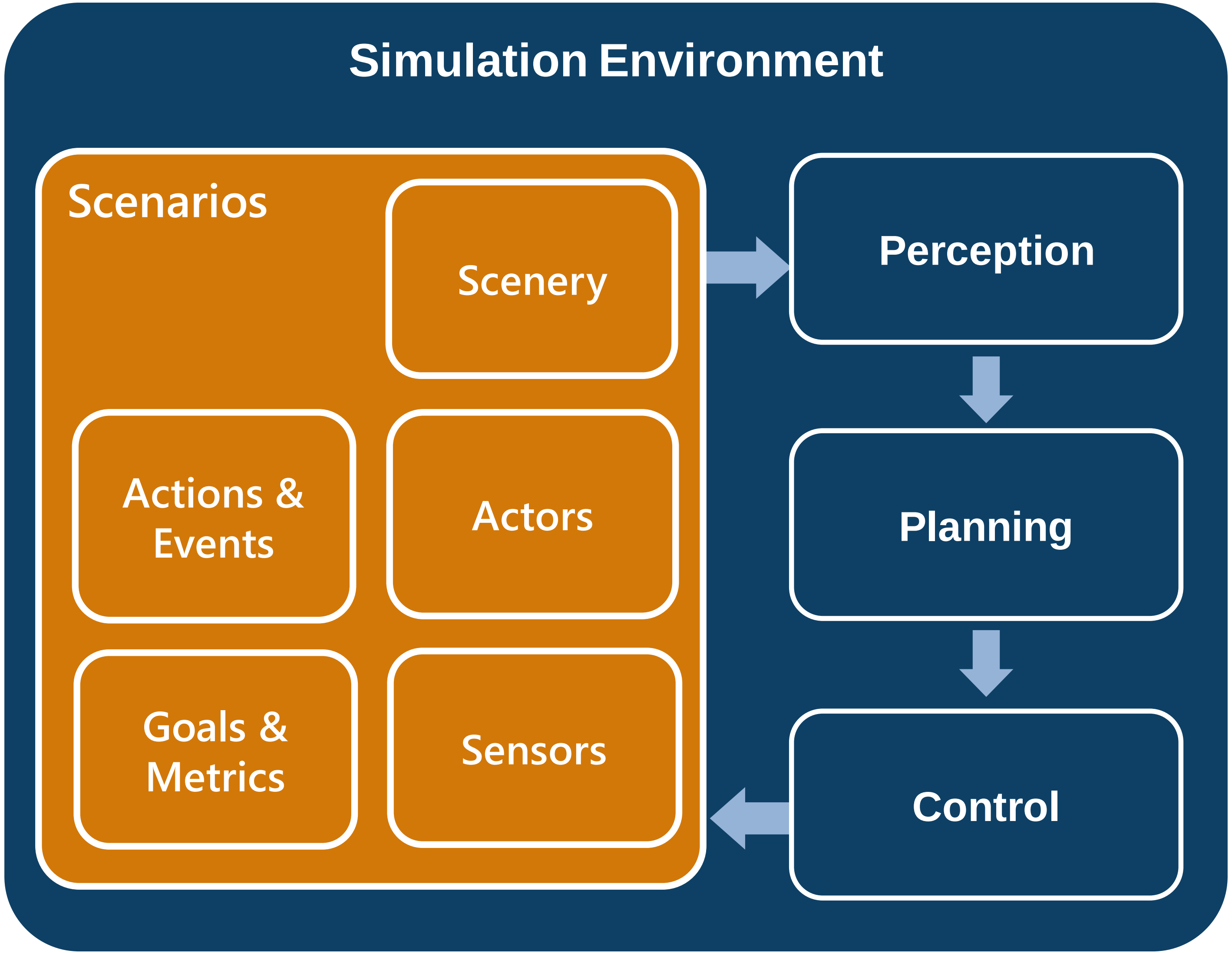
Visualize depth and semantic segmentation data captured from a camera sensor in a 3D simulation environment.

Open Model

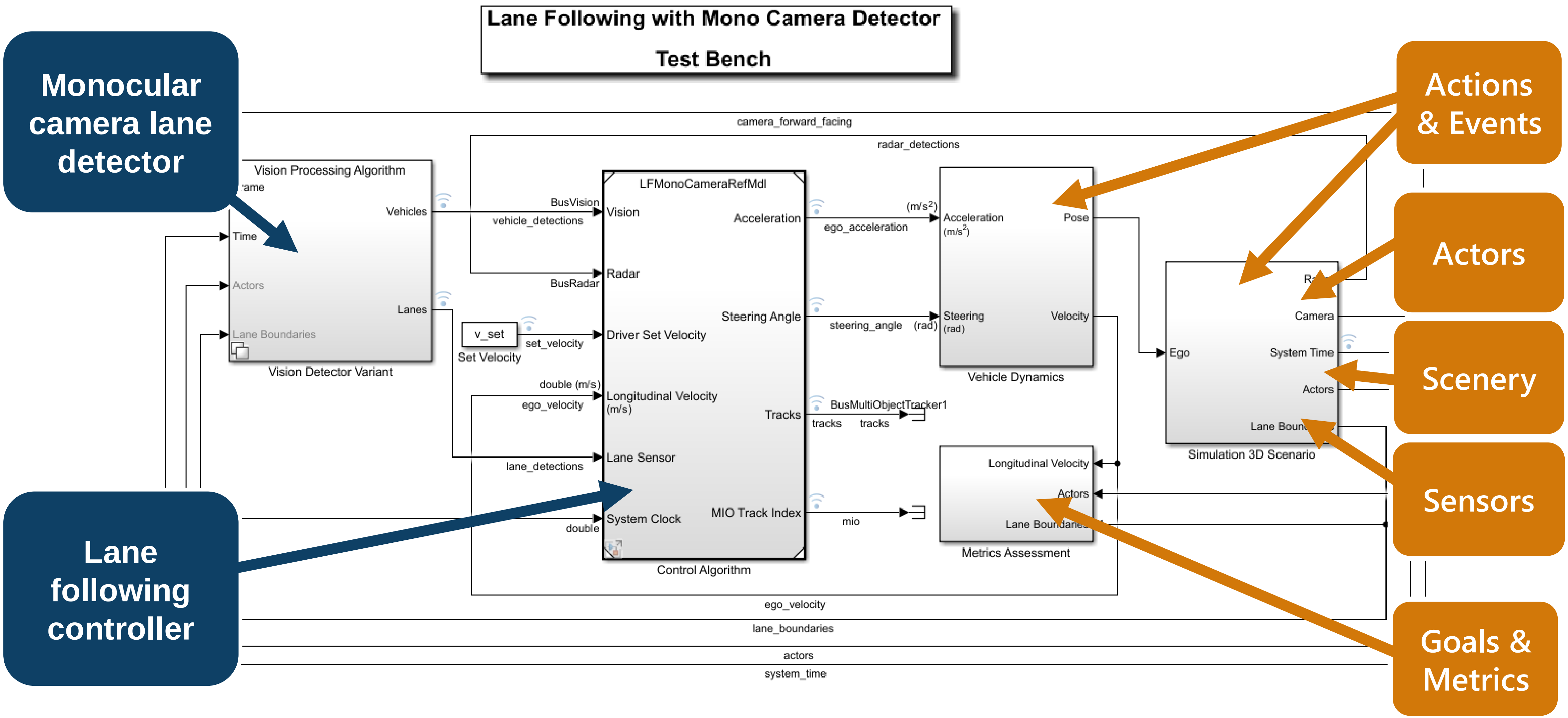
Simulating automated driving systems with MATLAB and Simulink



Simulating automated driving systems with MATLAB and Simulink



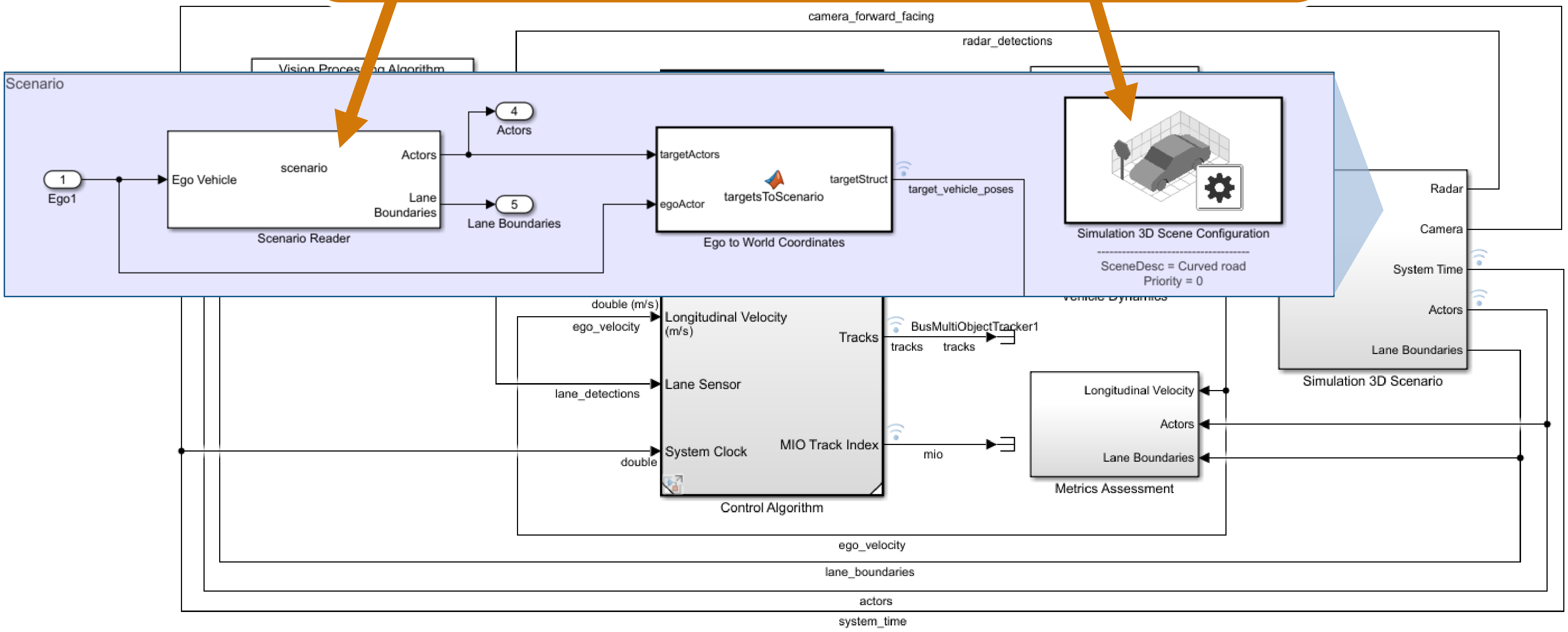
Integrate components and model scenarios



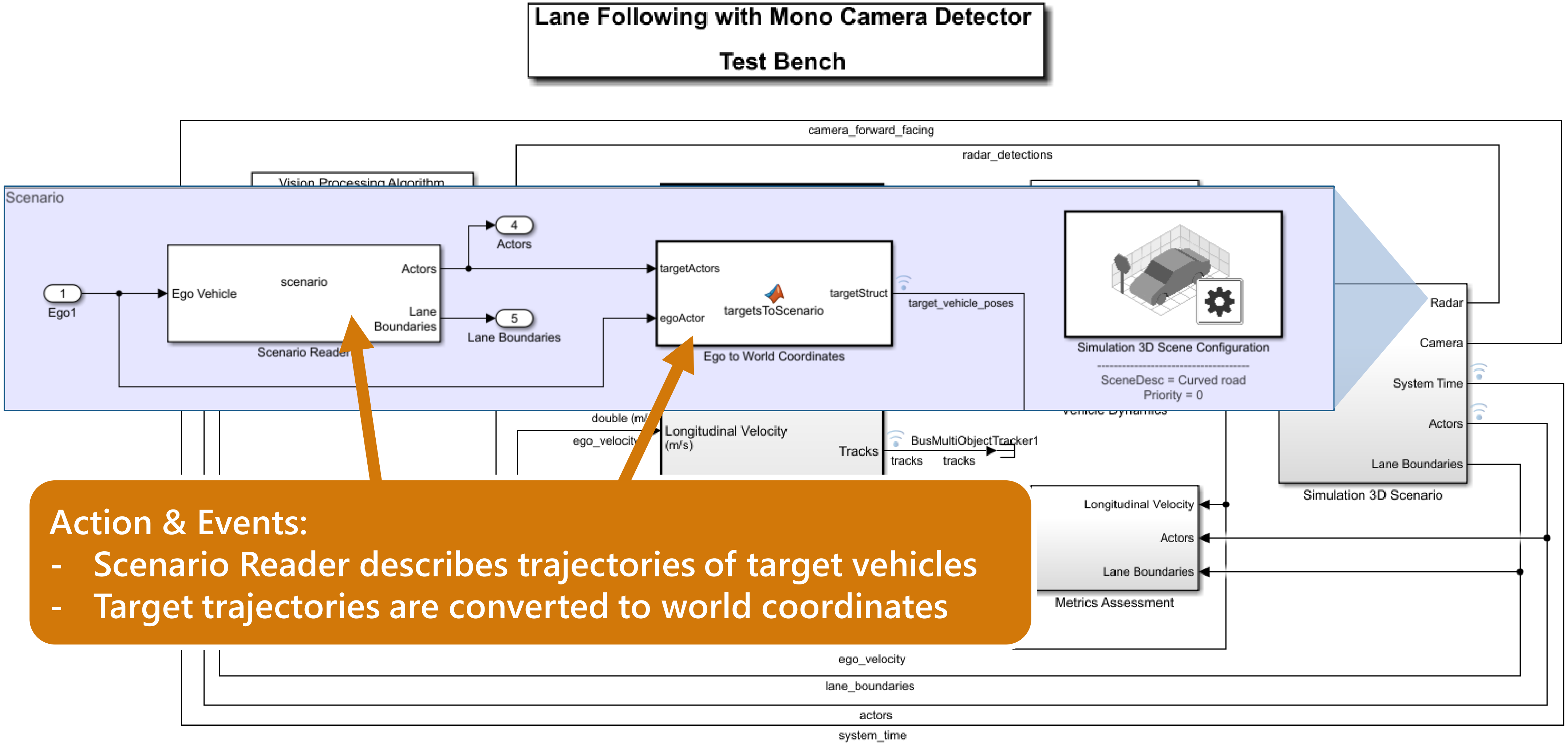
Specify equivalent 3D Simulation scenery

Scenery:
- Equivalent straight and curved roads in Simulation 3D Scene Configuration and Scenario Reader

Supported Scenery:
- ✓ Straight road
- ✓ Curved road segment
- ✓ Curved road (not exposed in example, but available)



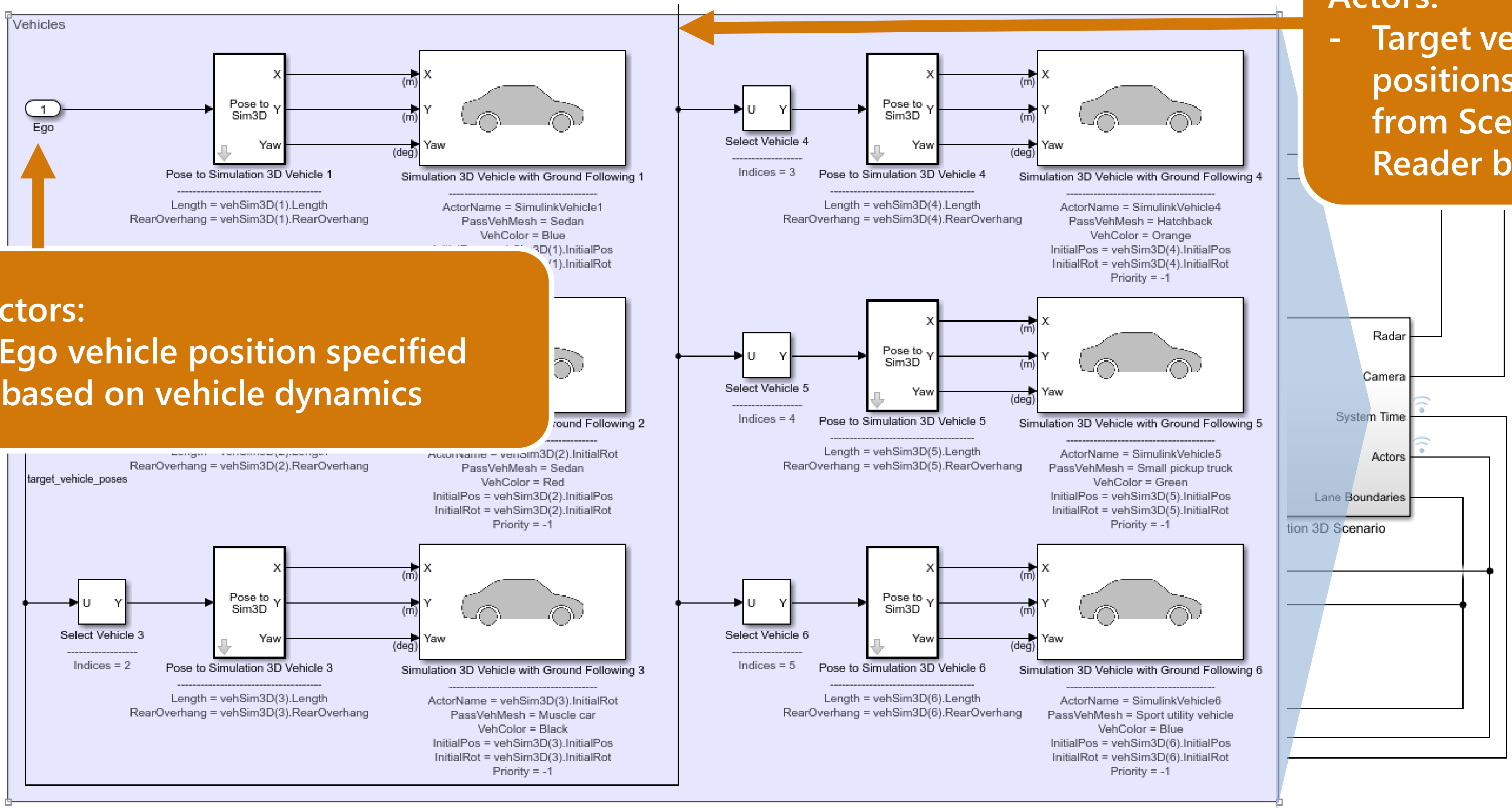
Specify 3D Simulation actor trajectories



Action & Events:

- Scenario Reader describes trajectories of target vehicles
- Target trajectories are converted to world coordinates

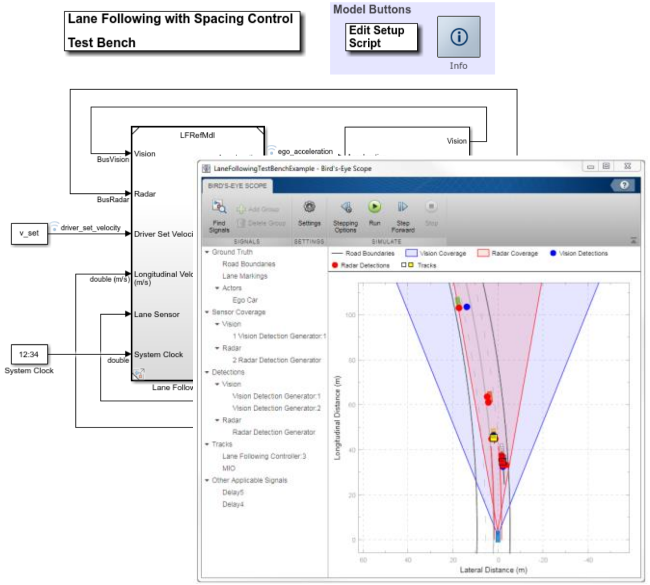
Specify 3D Simulation vehicles

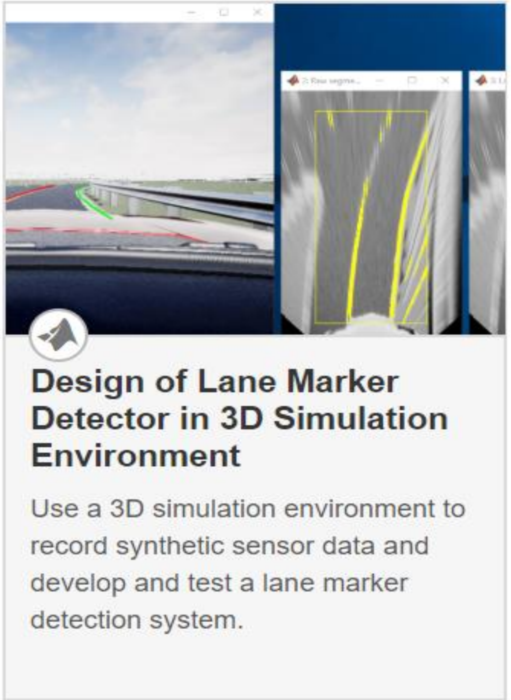


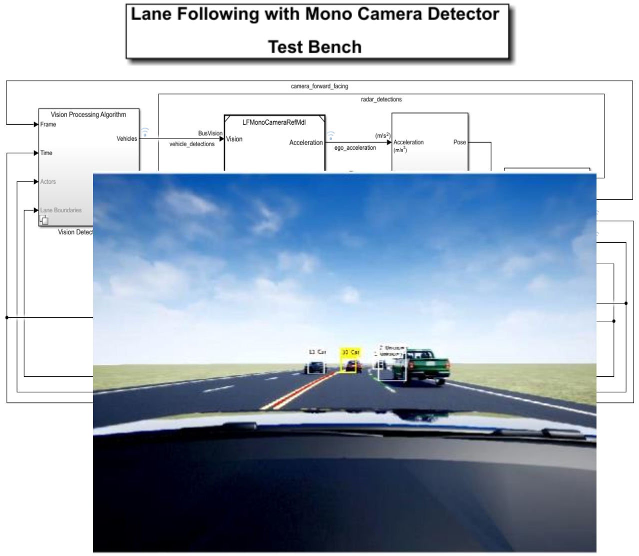
Actors:
- Target vehicle positions specified from Scenario Reader block

Actors:
- Ego vehicle position specified based on vehicle dynamics

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

Some common questions from automated driving engineers

Perception

Planning

Control

Simulation Integration

ROS | CAN | C/C++ | Python | Cross Release | Third Party

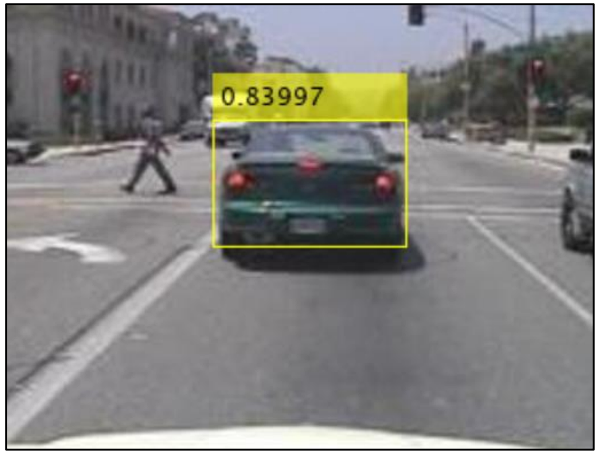
How can I **synthesize scenarios** to test my designs?

How can I **discover and design** in multiple domains?

How can I **integrate** with other environments?

Design camera, lidar, and radar perception algorithms

Detect vehicle with camera



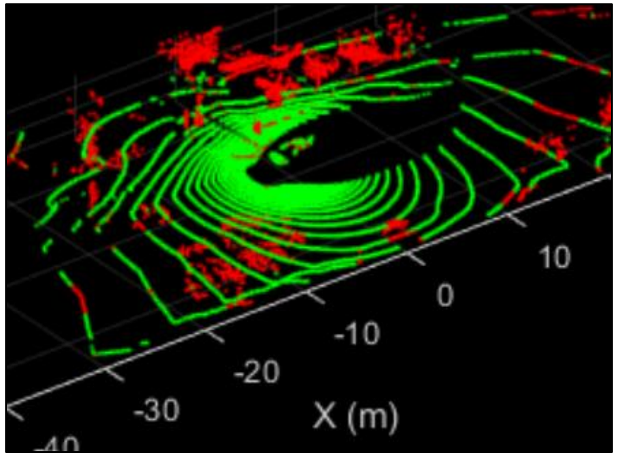
Object Detection Using YOLO v2 Deep Learning

Computer Vision Toolbox™

Deep Learning Toolbox™

R2019a

Detect ground with lidar

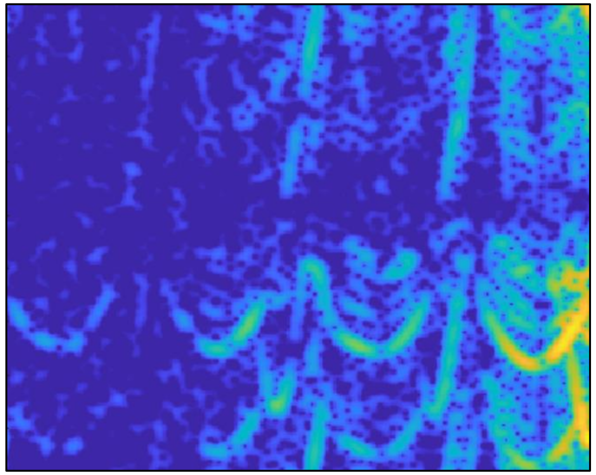


Segment Ground Points from Organized Lidar Data

Computer Vision Toolbox™

R2018b

Detect pedestrian with radar

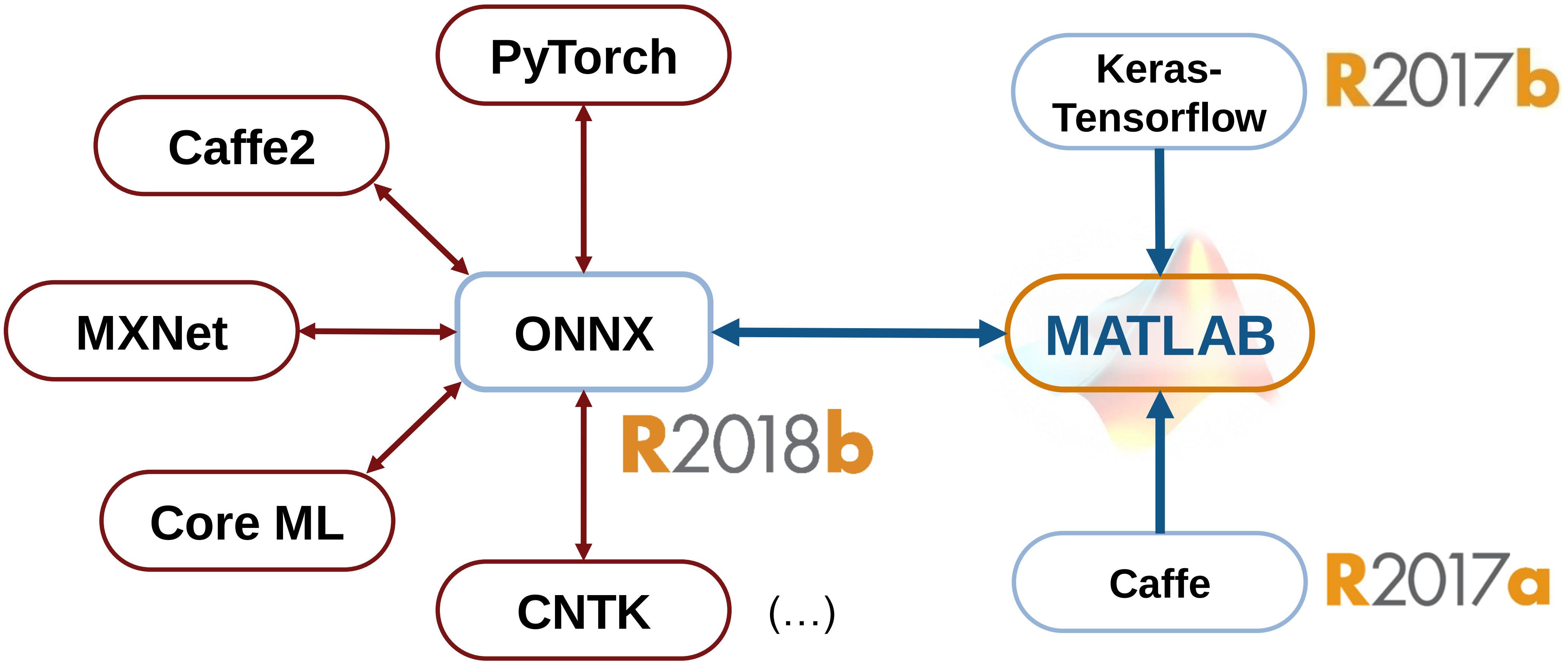


Introduction to Micro-Doppler Effects

Phased Array System Toolbox™

R2019a

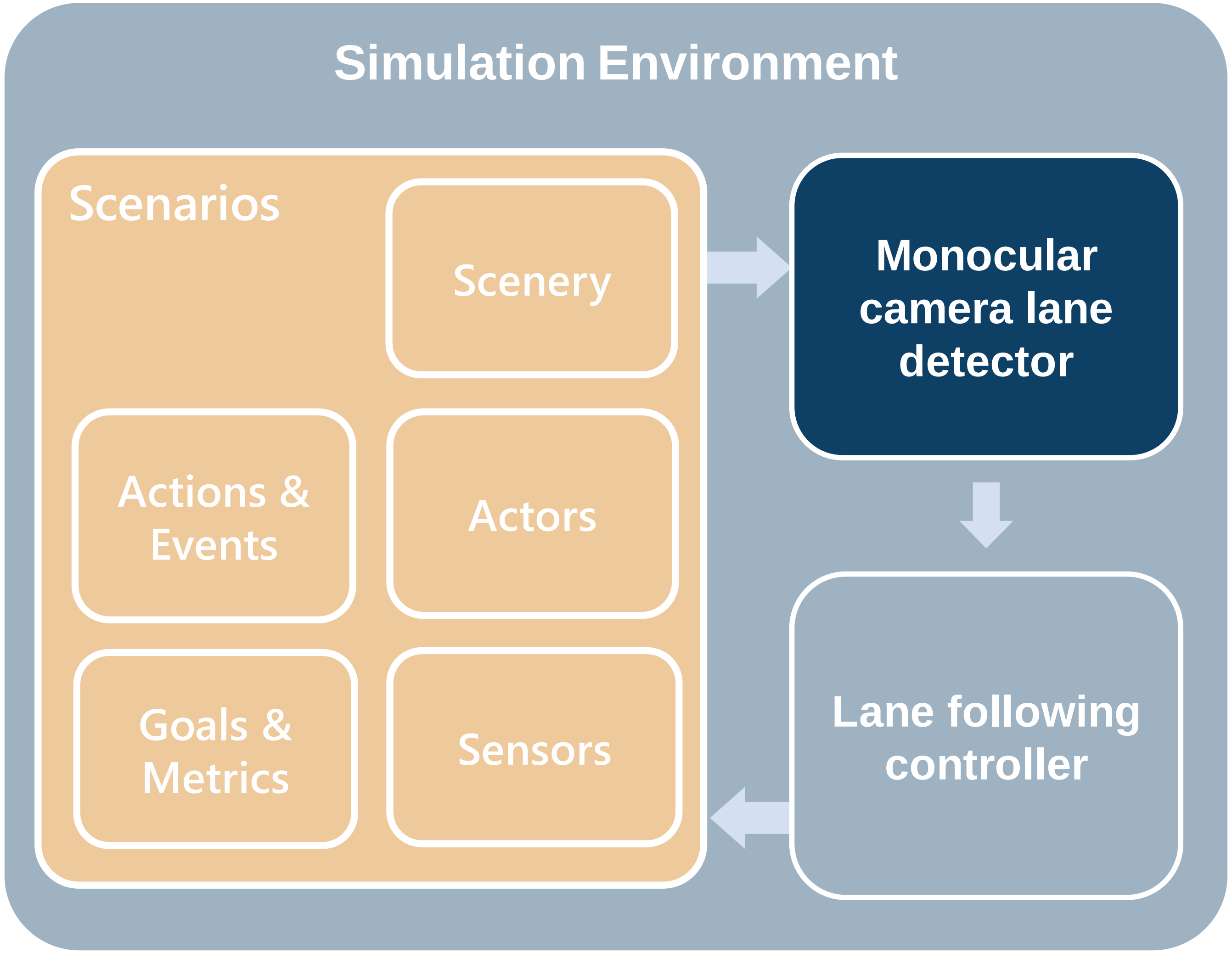
Interoperate with neural network frameworks



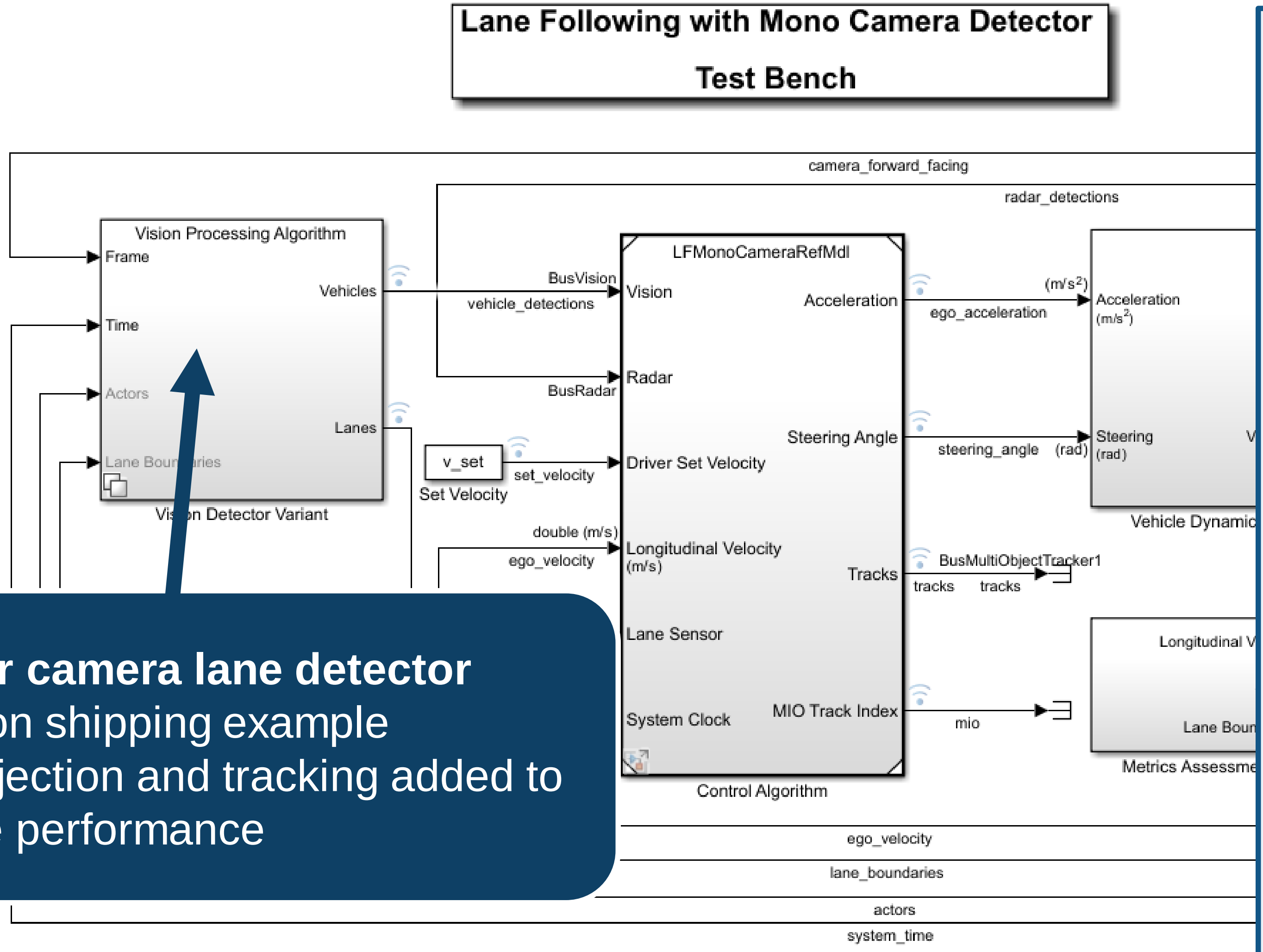
Open Neural Network Exchange

Visit the Demo Stations to see more…

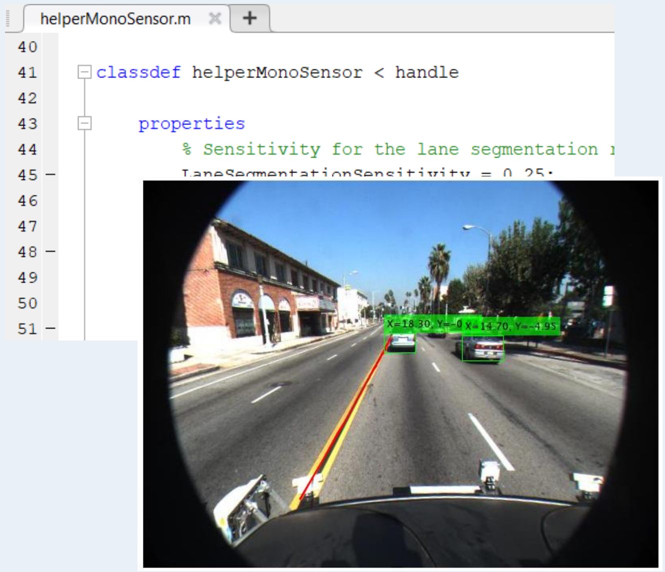
Simulate lane detection and lane following system with MATLAB and Simulink



Monocular Camera Lane detector example



Monocular camera lane detector
- Based on shipping example
- Lane rejection and tracking added to improve performance



[Visual Perception Using Monocular Camera]

Automated Driving Toolbox™

R2017a

Design detector for lidar point cloud data

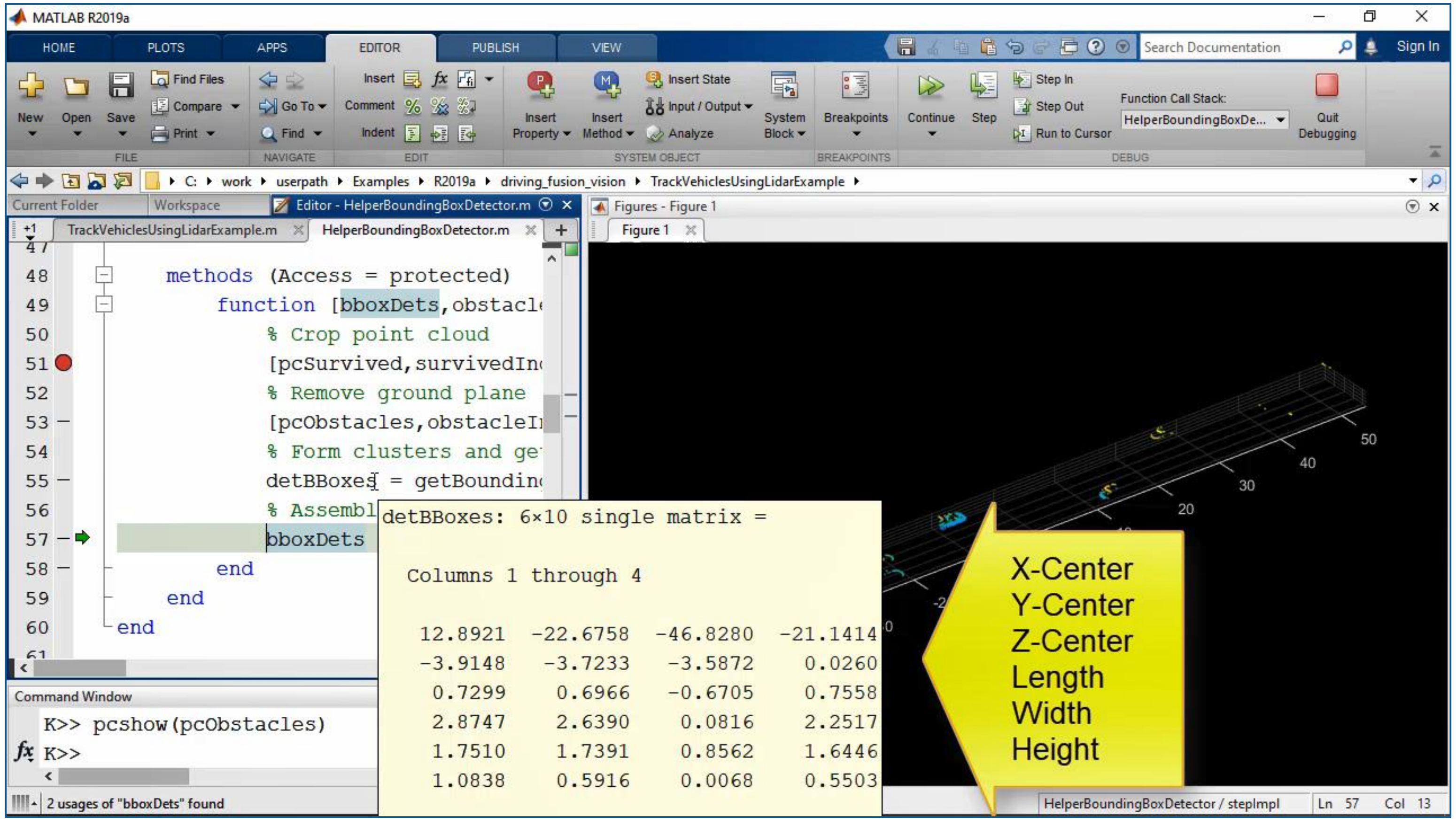
[Track Vehicles Using Lidar: From Point Cloud to Track List]

- Design 3-D bounding box detector
- Design tracker (target state and measurement models)
- Generate C/C++ code for detector and tracker

Sensor Fusion and Tracking Toolbox™

Computer Vision Toolbox™

R2019a



Design tracker for lidar point cloud data

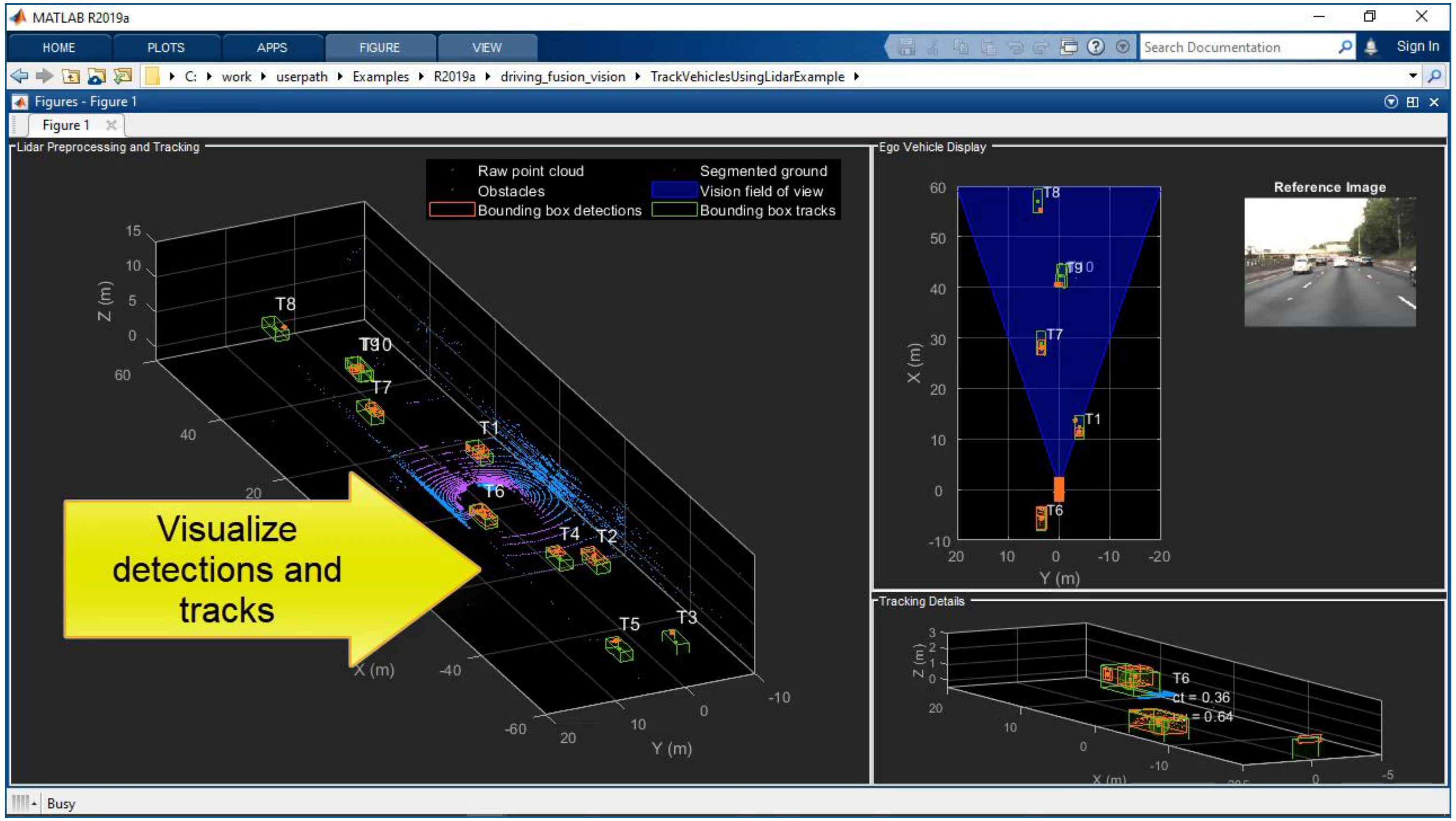
[Track Vehicles Using Lidar: From Point Cloud to Track List]()

- Design 3-D bounding box detector
- Design tracker (target state and measurement models)
- Generate C/C++ code for detector and tracker

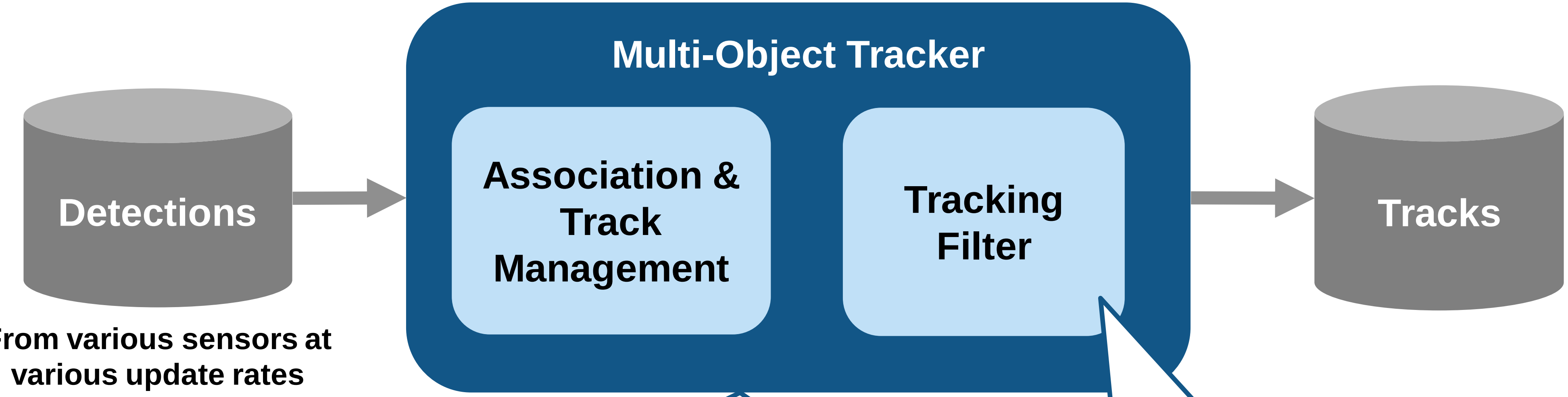
Sensor Fusion and Tracking Toolbox™

Computer Vision Toolbox™

R2019a



Design trackers



From various sensors at various update rates

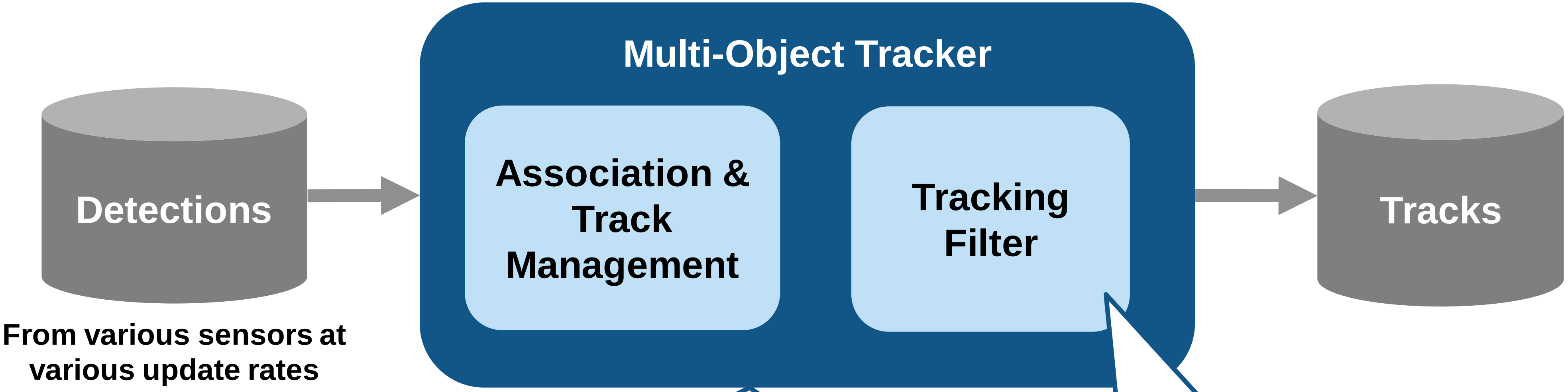
- Multi-object tracker

- Linear, extended, and unscented Kalman filters

Automated Driving Toolbox™

R2017a

Design trackers



- Multi-object tracker
- Global Nearest Neighbor (GNN) tracker
- Joint Probabilistic Data Association (JPDA) tracker
- Track-Oriented Multi-Hypothesis Tracker (TOMHT)
- Probability Hypothesis Density (PHD) tracker

- Linear, extended, and unscented Kalman filters
- Particle, Gaussian-sum, IMM filters

Automated Driving Toolbox™

Sensor Fusion and Tracking Toolbox™

R2019a

Evaluate error metrics

[Extended Object Tracking]

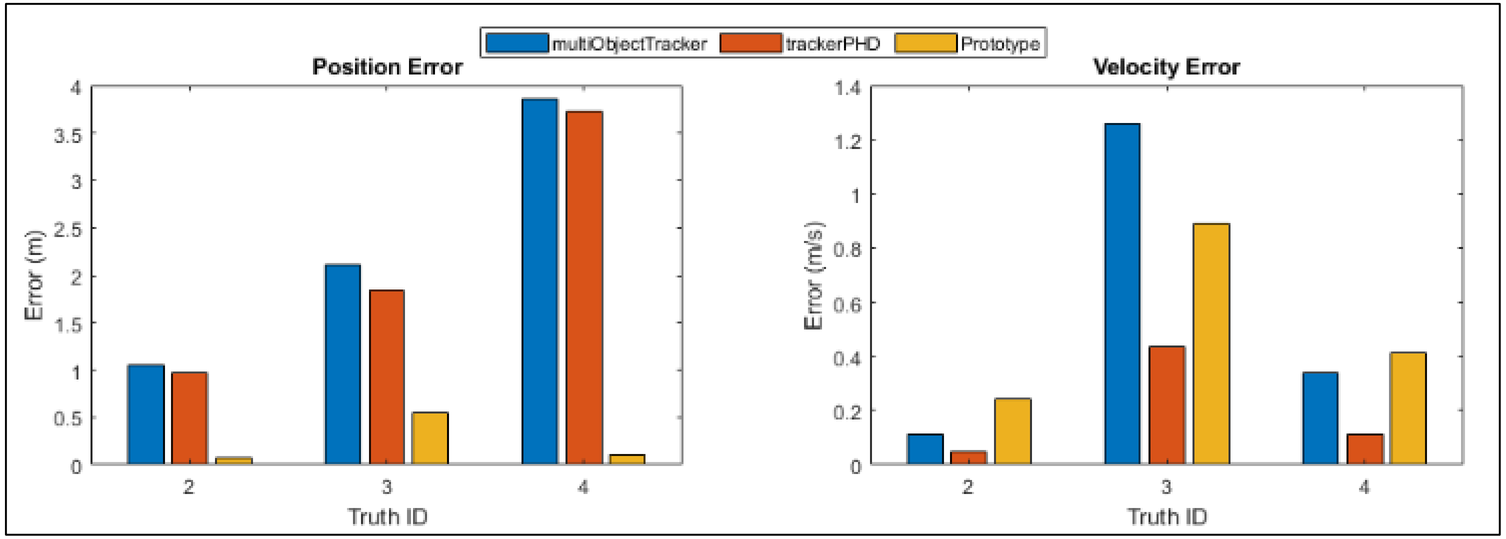
- Design multi-object tracker
- Design extended object trackers
- Evaluate tracking metrics
- Evaluate error metrics
- Evaluate desktop execution time

Sensor Fusion and

Tracking Toolbox™

Automated Driving Toolbox™

Updated R2019a



Multi-object tracker
Probability Hypothesis Density tracker
Extended object (size and orientation) tracker

Compare relative execution times of object trackers

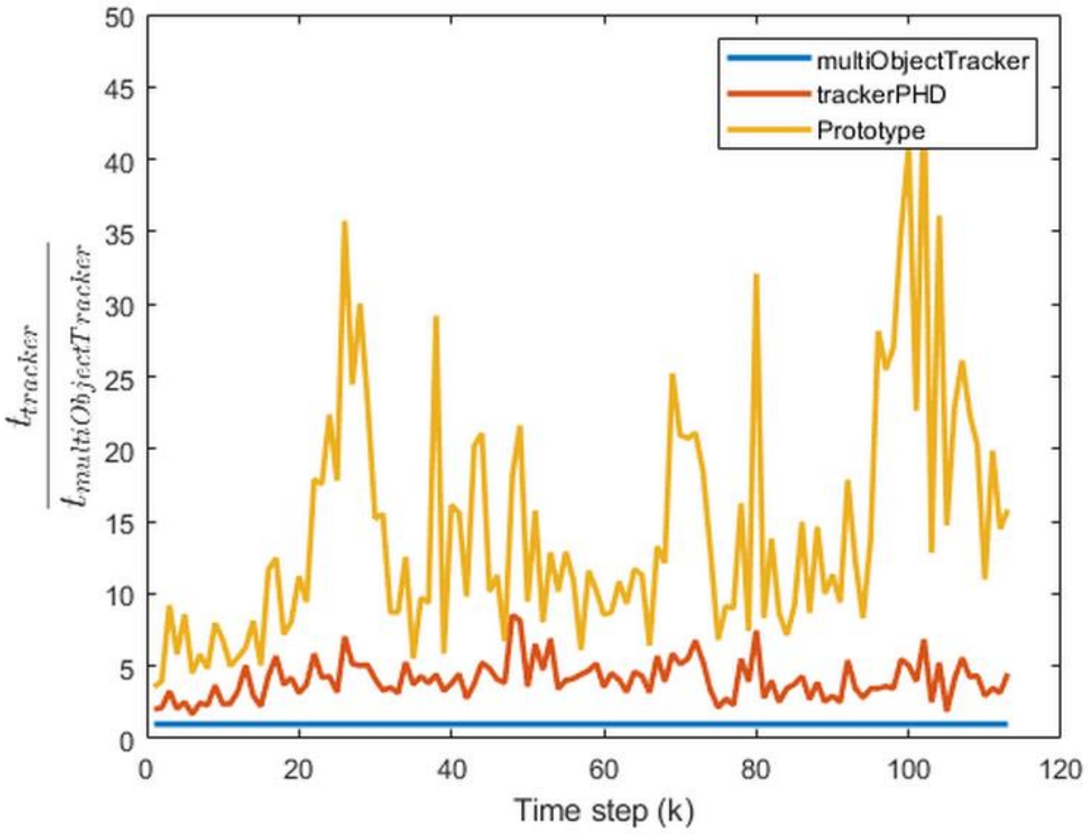
Extended Object Tracking

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

Sensor Fusion and Tracking Toolbox™

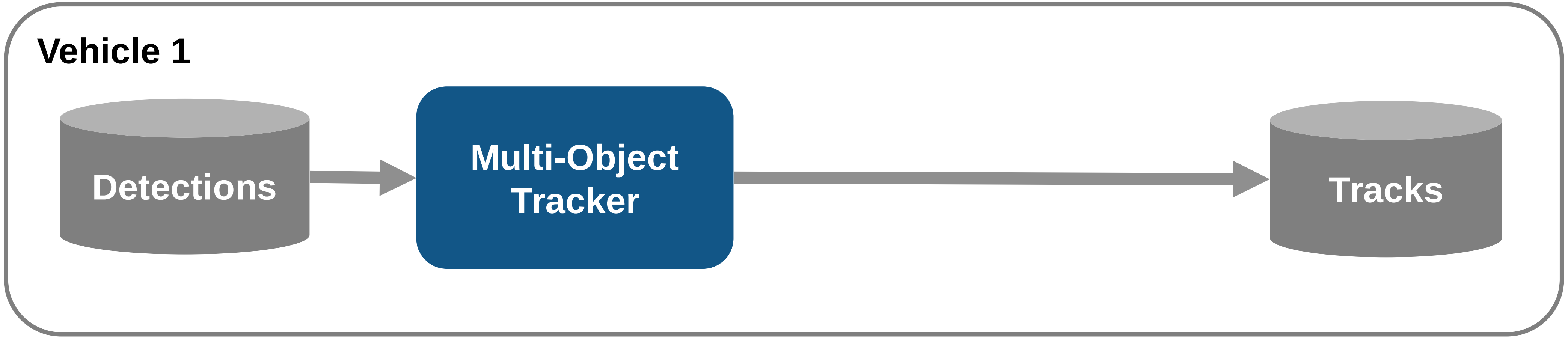
Automated Driving Toolbox™

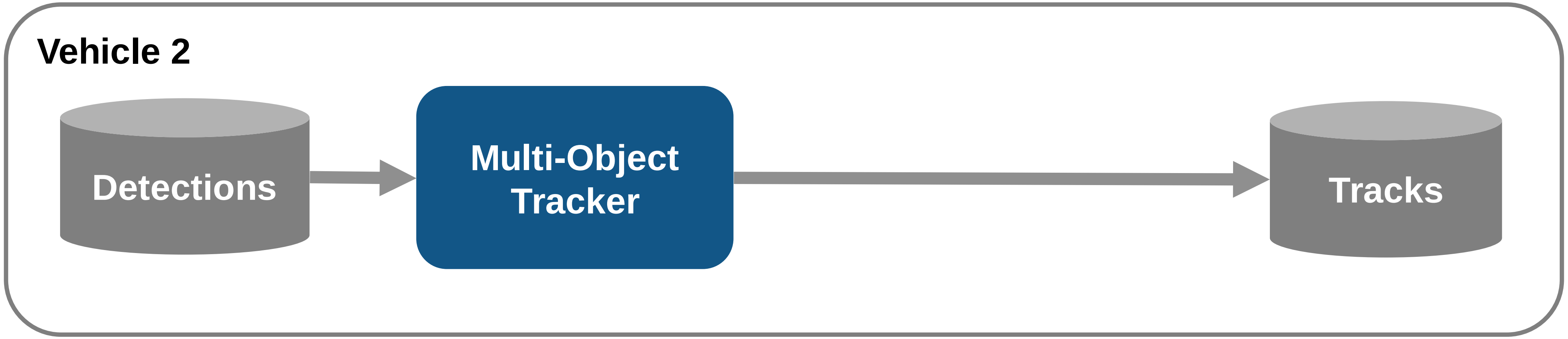
Updated R2019a



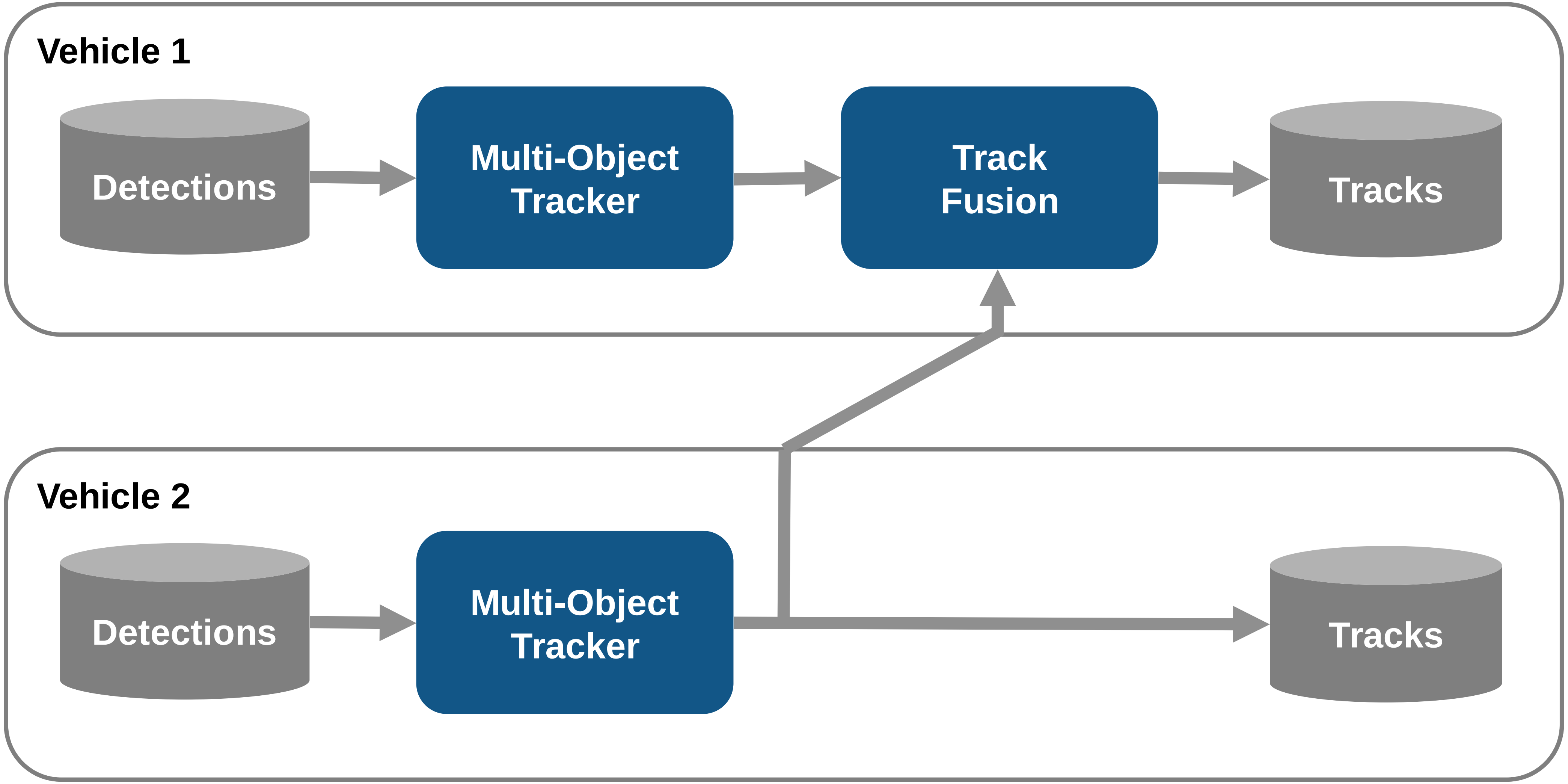
- Multi-object tracker
- Probability Hypothesis Density tracker
- Extended object (size and orientation) tracker

Design track level fusion systems

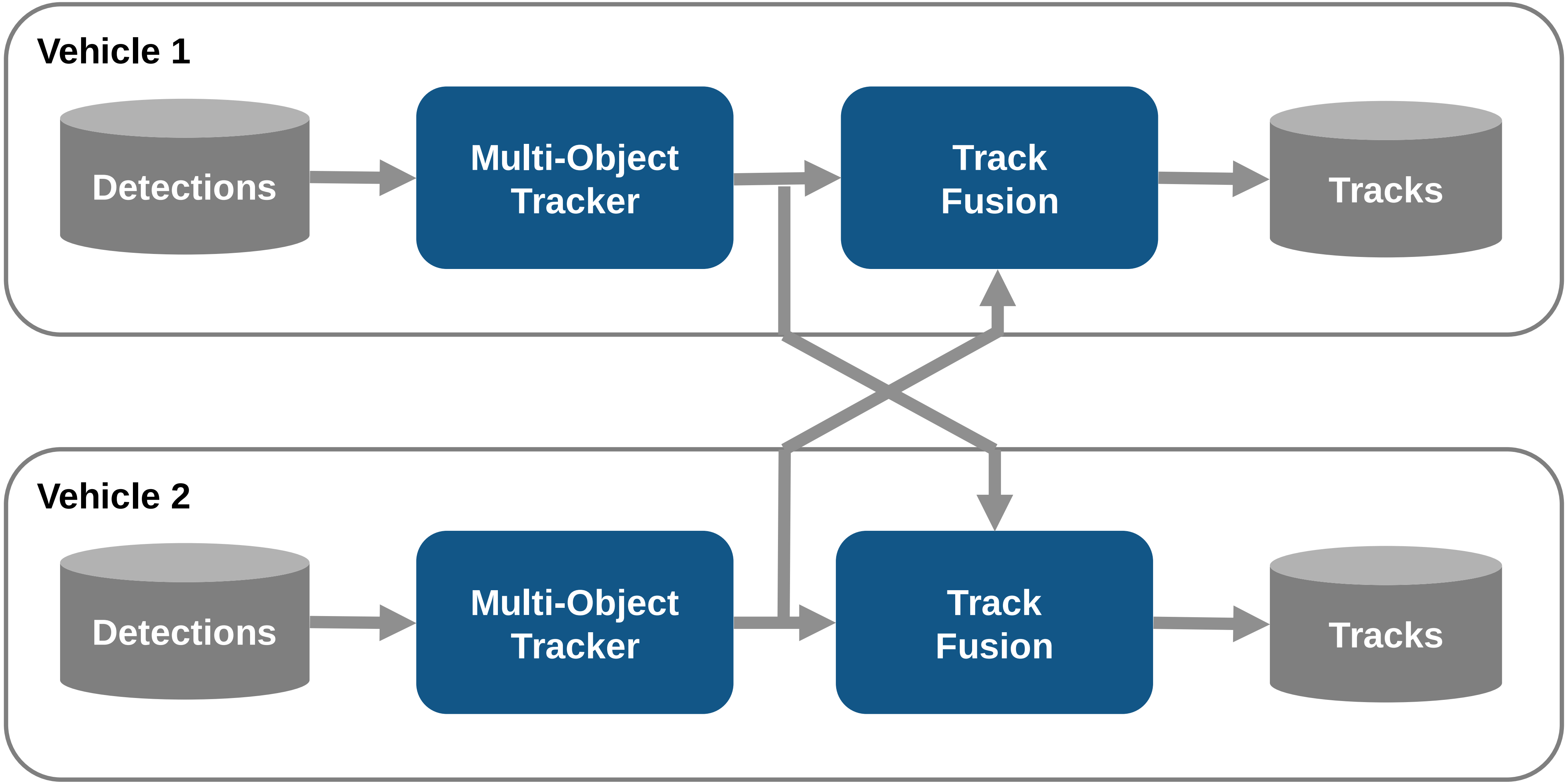




Design track level fusion systems



Design track level fusion systems

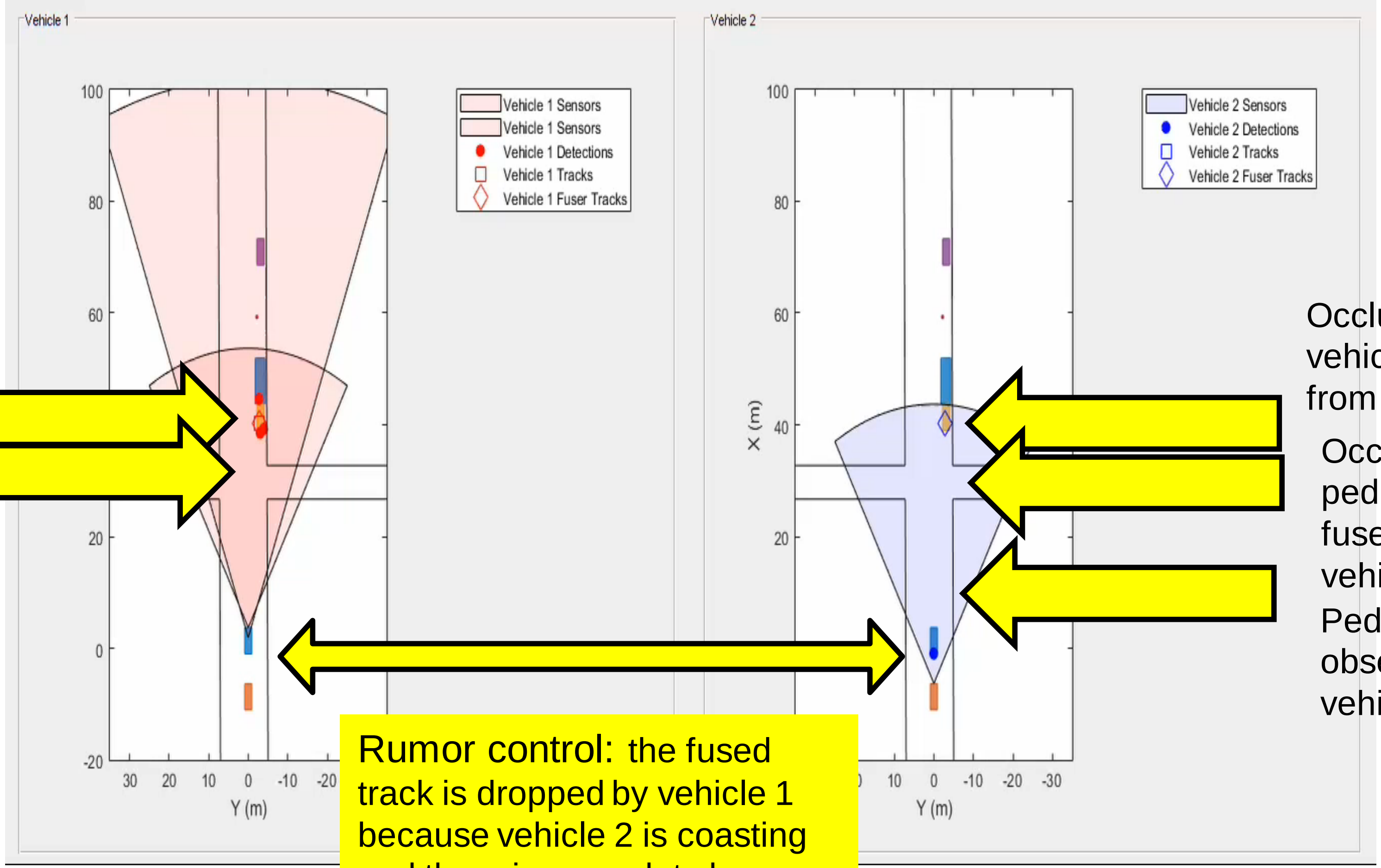


Track-level fusion

[Track-to-Track Fusion for Automotive Safety Applications]

Parked vehicles observed by vehicle 1

Pedestrian observed by vehicle 1



Occluded vehicle fused from vehicle 1

Occluded pedestrian fused from vehicle 1

Pedestrian observed by vehicle 2

Rumor control: the fused track is dropped by vehicle 1 because vehicle 2 is coasting and there is no update by vehicle 1 sensors

Sensor Fusion and Tracking Toolbox™
Automated Driving Toolbox™

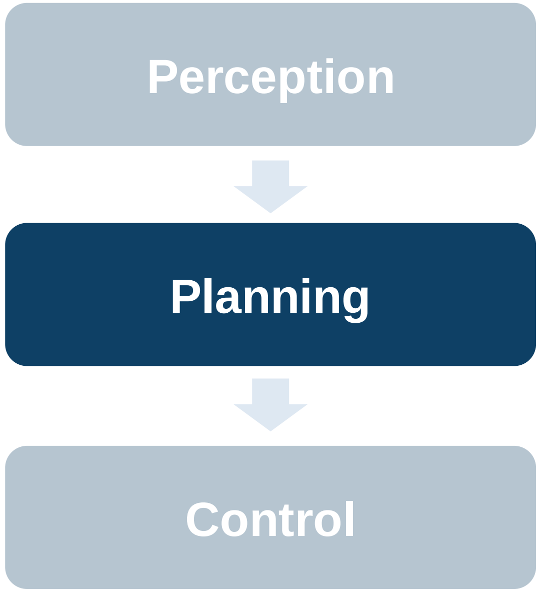
R2019b

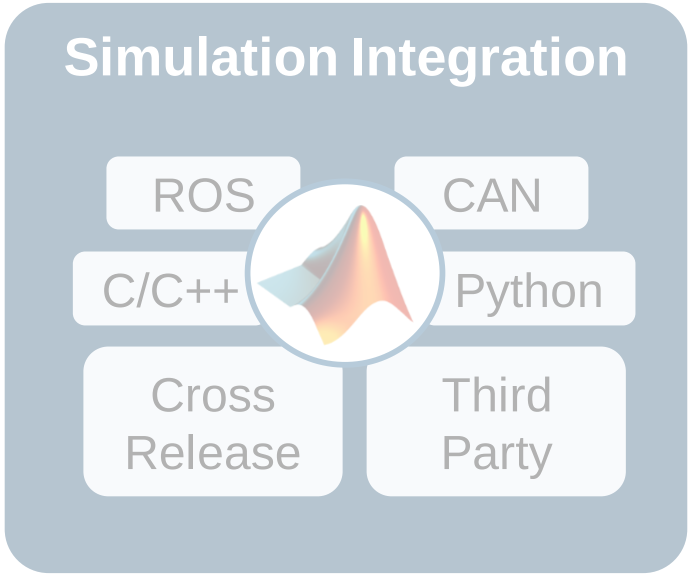
For more on Sensor Fusion and Tracking…

Visit Marc Willerton's presentation later this afternoon

	Technical Computing	Model-Based Design	Getting Started with MATLAB and Simulink	Master Classes	Innovation Auditorium
15:15	Break				
15:45	Developing Smart IoT Sensors Using the MathWorks Toolchain *Samuel Bailey, Skyrad Consulting*	Synchronous Machine Modelling Using Simscape *Peenki Rani, Cummins Generator Technologies*	Sensor Fusion and Tracking for Autonomous Systems *Marc Willerton, MathWorks*	Simplifying Requirements-Based Verification with Model-Based Design *Fraser Macmillen, MathWorks*	Predictive Maintenance with MATLAB *Phil Rottier, MathWorks*
16:15	Industrial IoT and Digital Twins *Coorous Mohtadi, MathWorks*	Developing Fit-For-Purpose Simscape Models to Support System and Control Design *Rick Hyde, MathWorks*			
17:00	End of Day				

Some common questions from automated driving engineers





How can I **synthesize scenarios** to test my designs?

How can I **discover and design** in multiple domains?

How can I **integrate** with other environments?

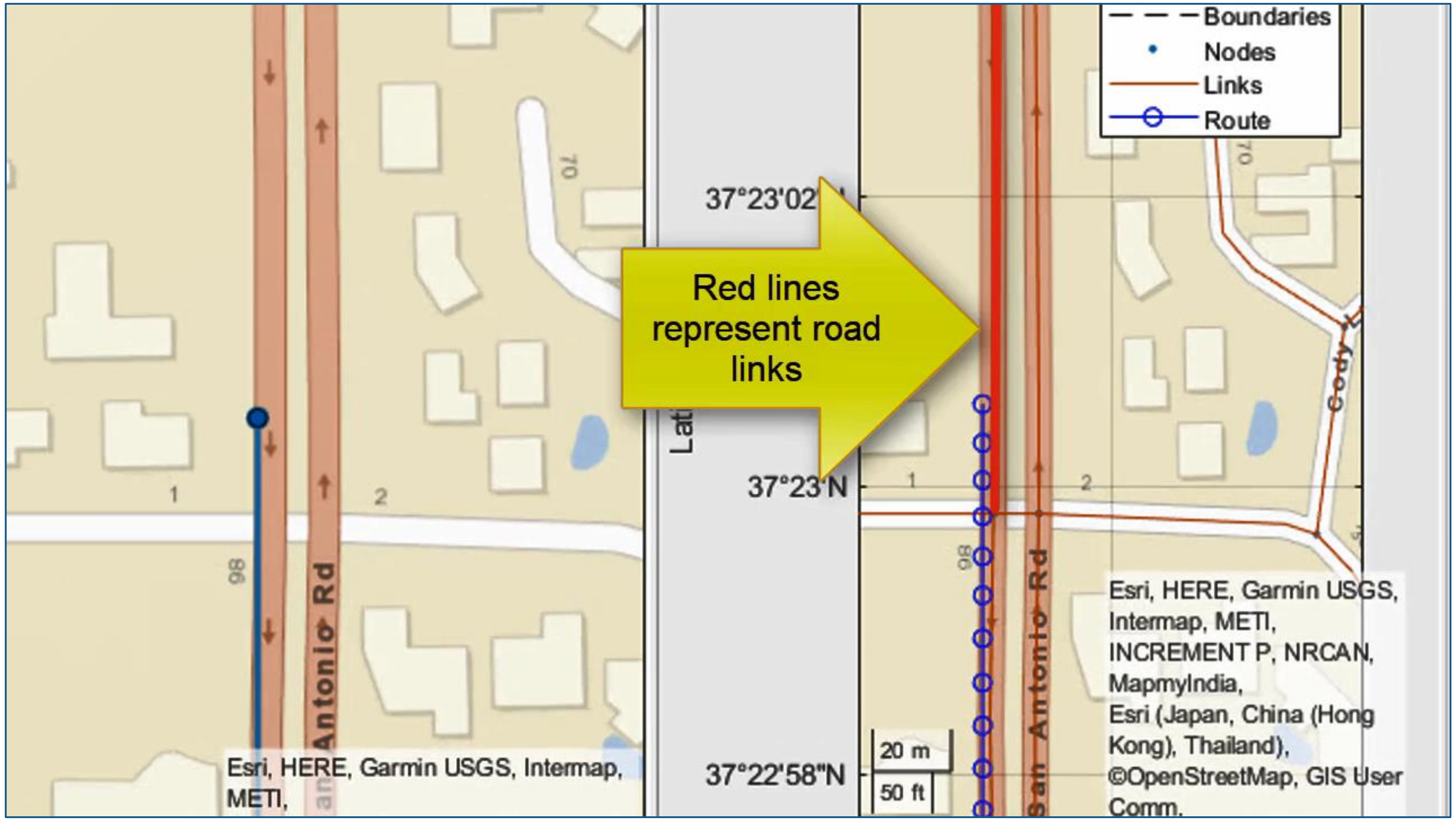
Read road and speed attributes from HERE HD Live Map data

[Use HERE HD Live Map Data to Verify Lane Configurations]

- Load camera and GPS data
- Retrieve speed limit
- Retrieve lane configurations
- Visualize composite data

Automated Driving Toolbox™

R2019a



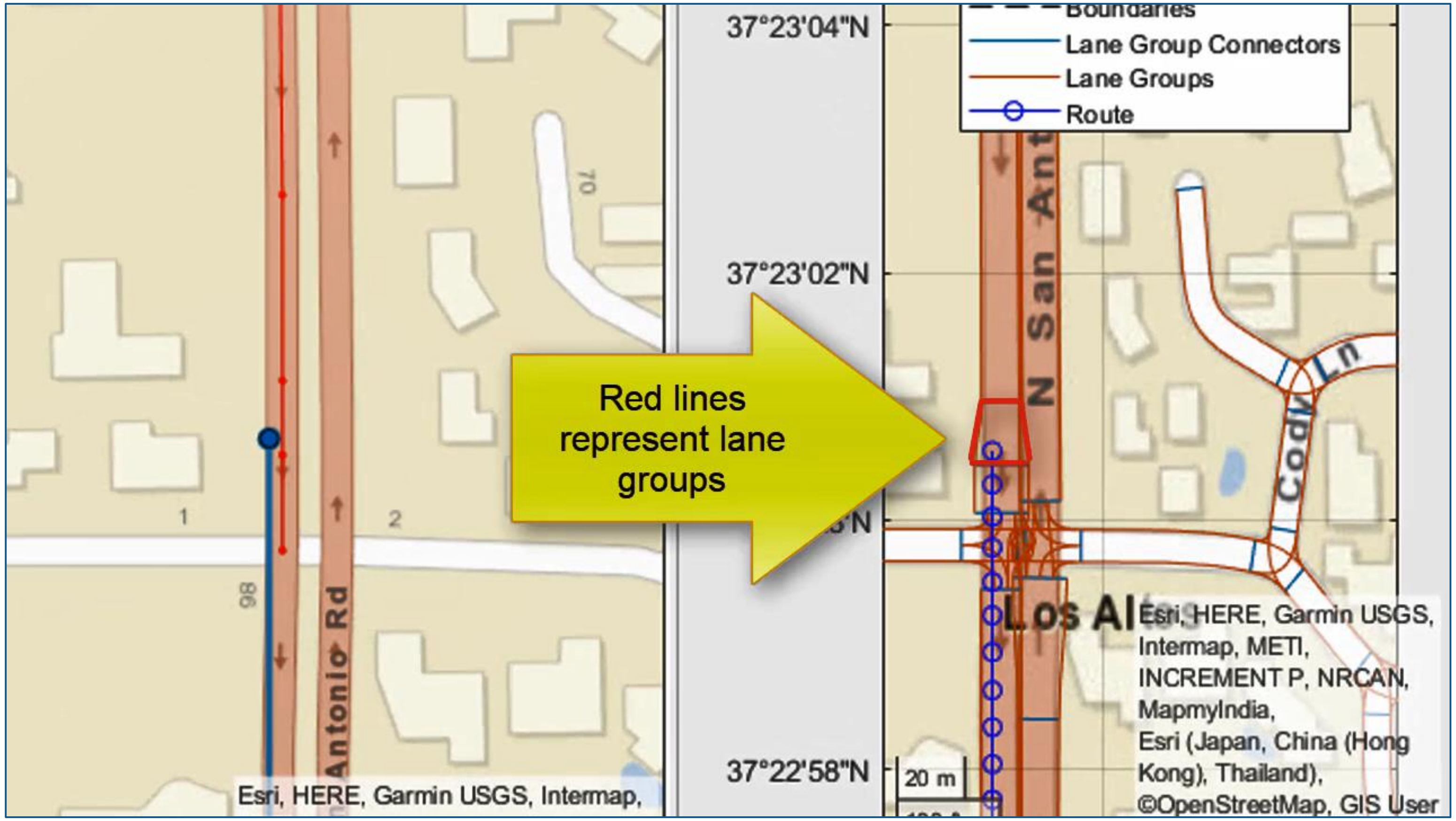
Read lane attributes from HERE HD Live Map data

[Use HERE HD Live Map Data to Verify Lane Configurations]

- Load camera and GPS data
- Retrieve speed limit
- Retrieve lane configurations
- Visualize composite data

Automated Driving Toolbox™

R2019a



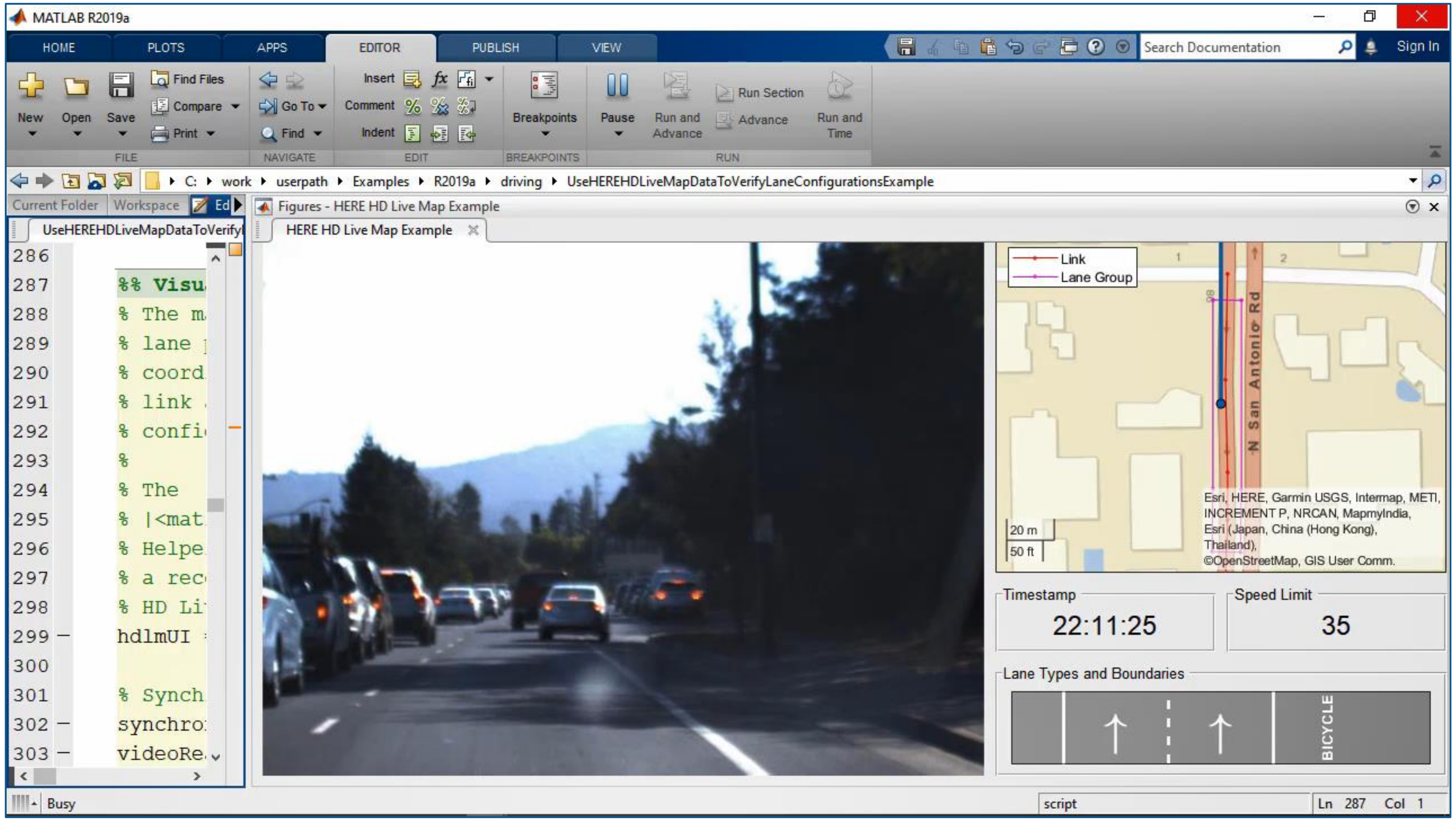
Visualize HERE HD Live Map recorded data

[Use HERE HD Live Map Data to Verify Lane Configurations]

- Load camera and GPS data
- Retrieve speed limit
- Retrieve lane configurations
- Visualize composite data

Automated Driving Toolbox™

R2019a



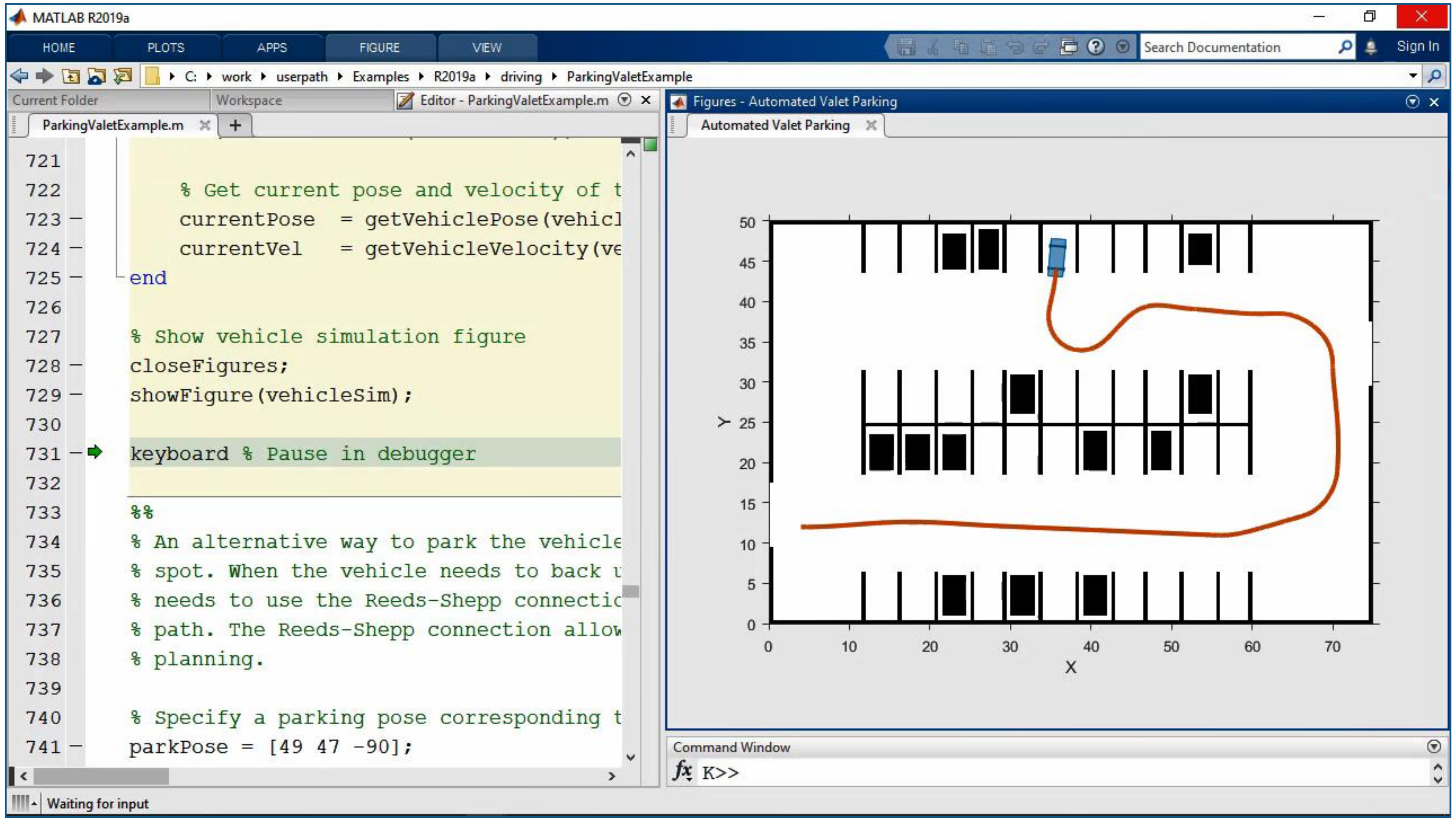
Design path planner

[Automated Parking Valet](#)

- Create cost map of environment
- Inflate cost map for collision checking
- Specify goal poses
- Plan path using rapidly exploring random tree (RRT*)

Automated Driving Toolbox™

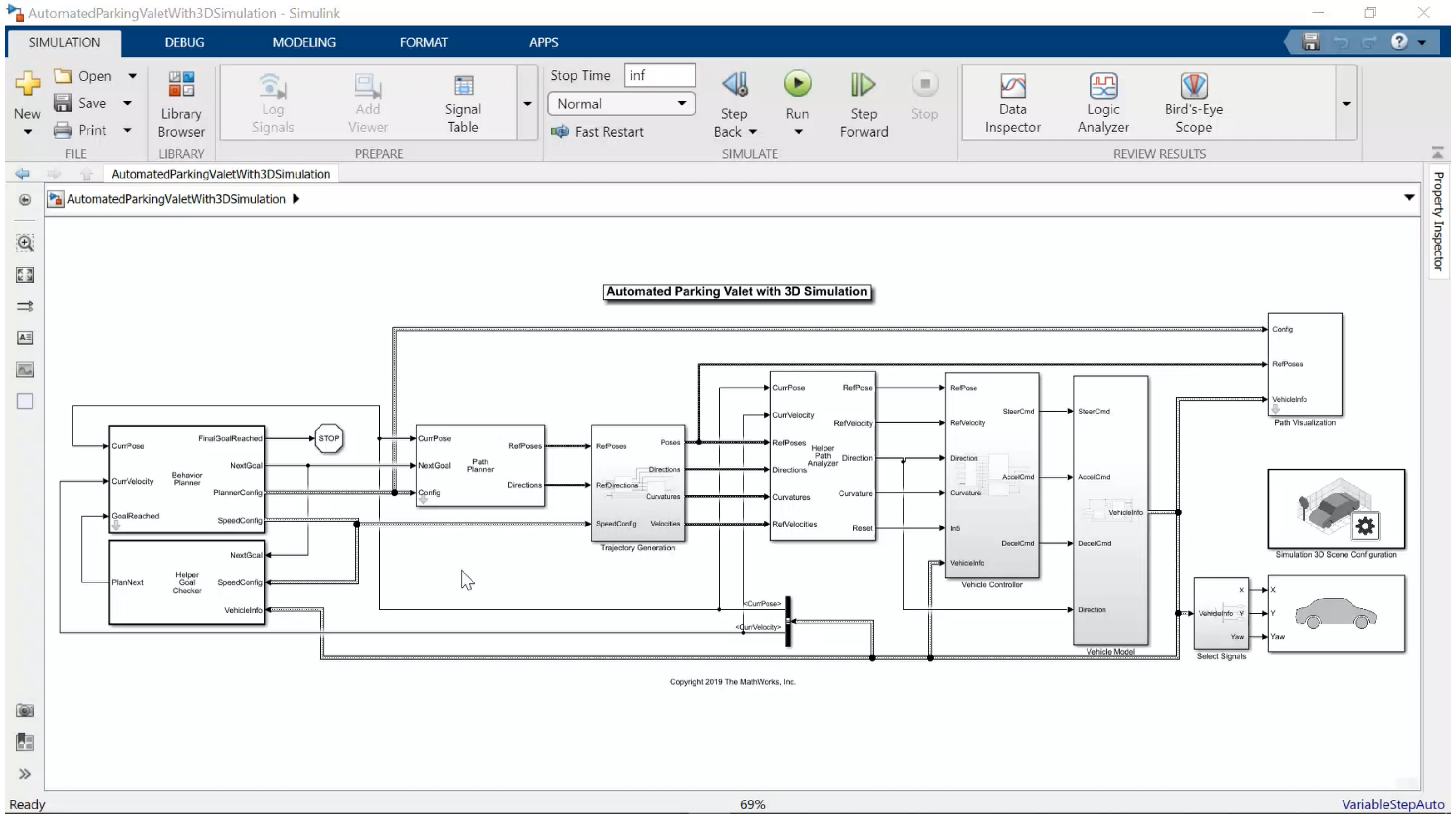
R2018a



Design path planner and controller

[Automated Parking Valet with Simulink]

- Integrate path planner
- Design lateral controller (based on vehicle kinematics)
- Design longitudinal controller (PID)
- Simulate closed loop with vehicle dynamics



[Visualize Automated Parking Valet Using 3D Simulation]

Automated Driving Toolbox™

R2018b R2019b

Some common questions from automated driving engineers

Perception

Planning

Control

Simulation Integration

ROS

CAN

C/C++

Python

Cross Release

Third Party

How can I
synthesize scenarios
to test my designs?

How can I
discover and design
in multiple domains?

How can I
integrate
with other environments?

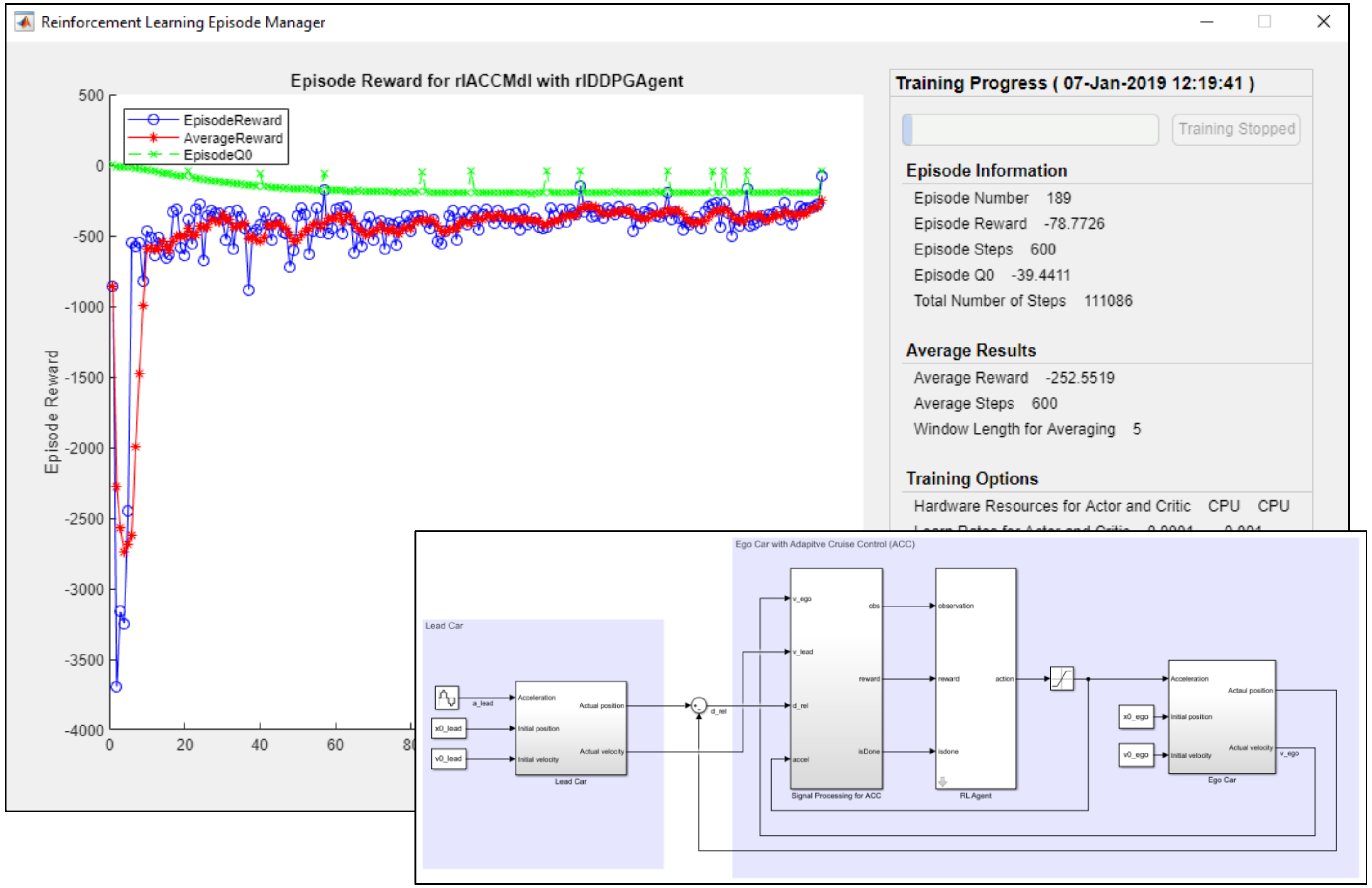
Train reinforcement learning networks for ADAS controllers

[Train Deep Deterministic Policy Gradient (DDPG) Agent for Adaptive Cruise Control]

- Create environment interface
- Create agent
- Train agent
- Simulate trained agent

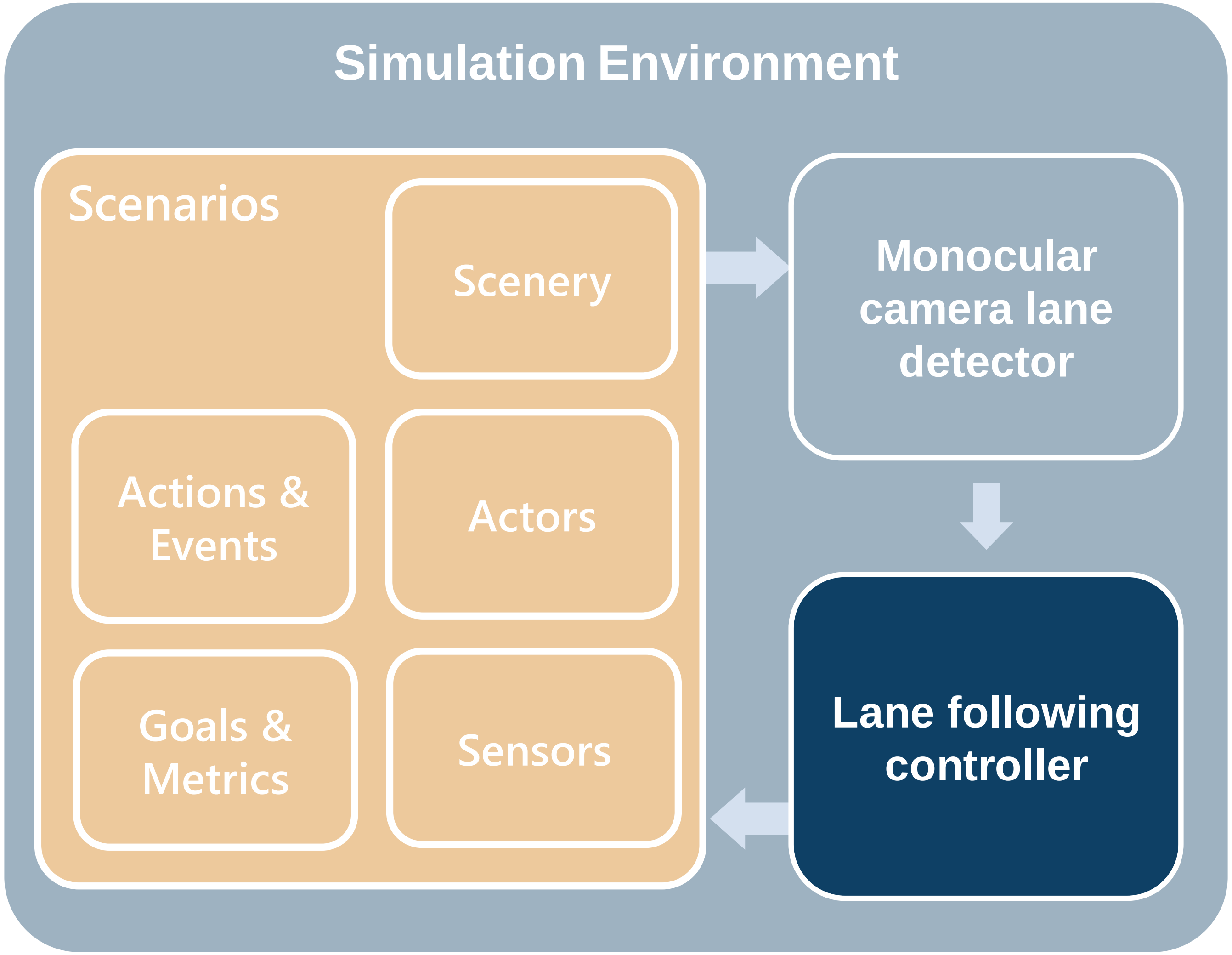
Reinforcement Learning Toolbox™

R2019a

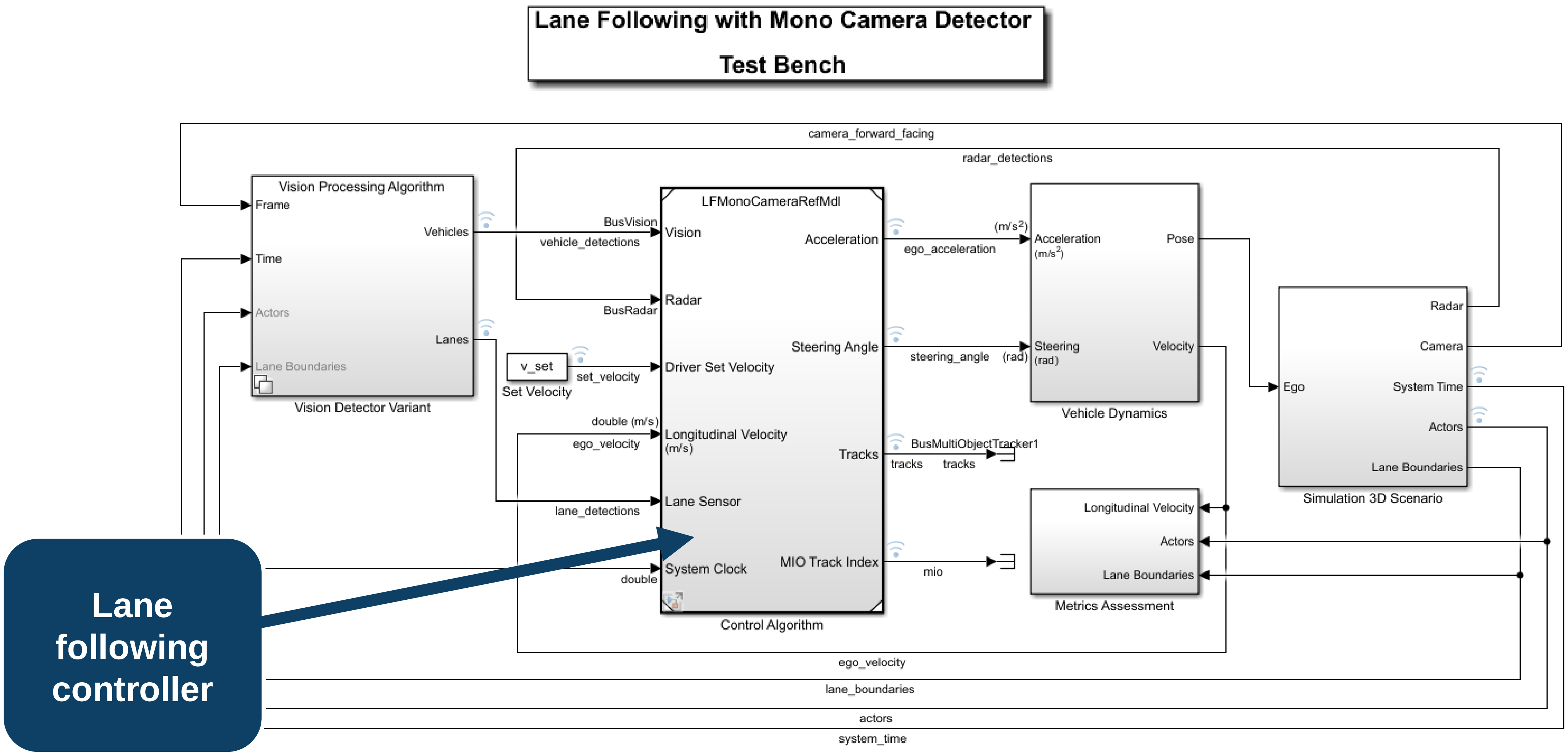


Visit the Demo Stations to see more…

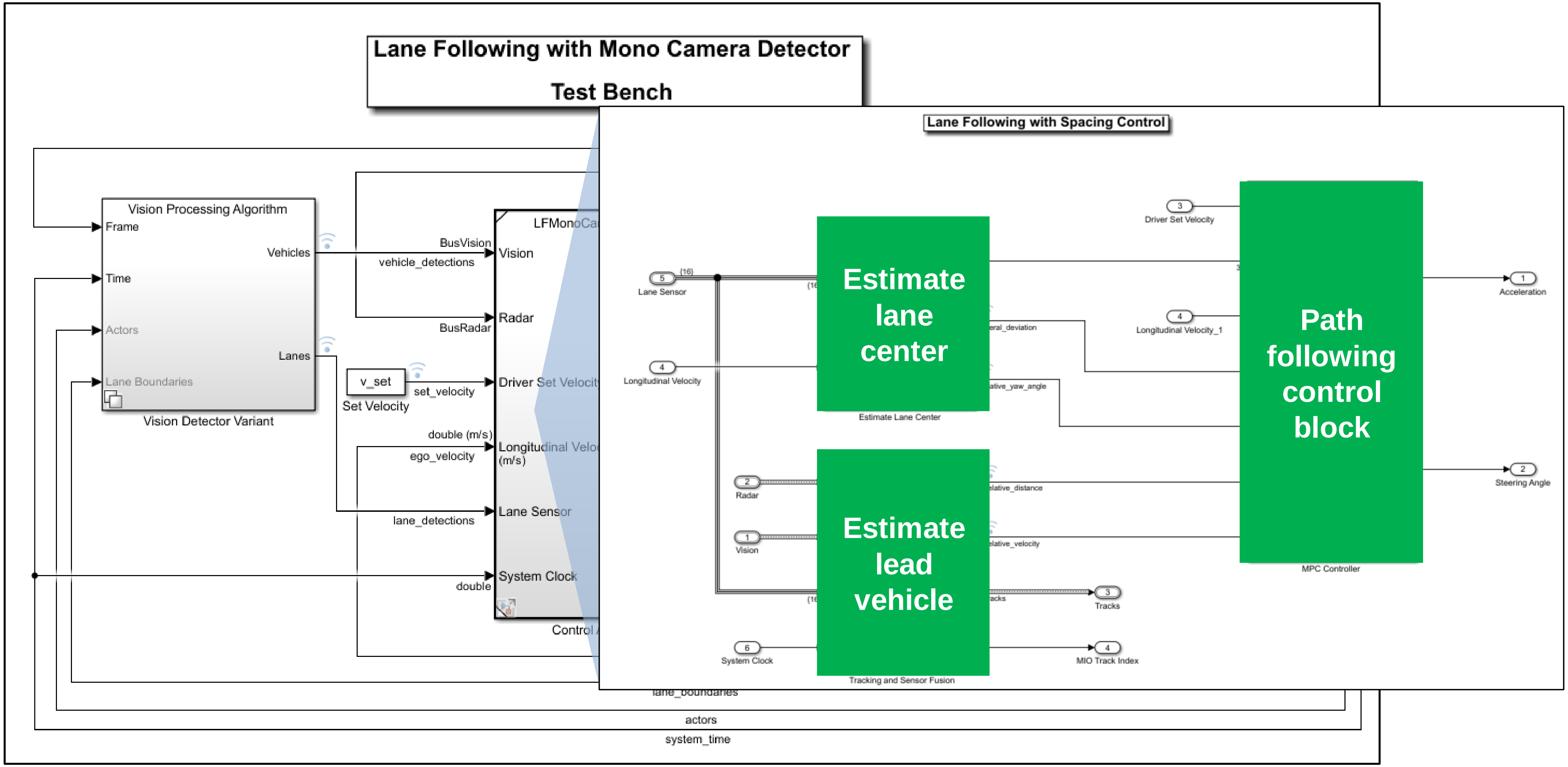
Simulate lane detection and lane following system with MATLAB and Simulink



Lane Following Controller Algorithm



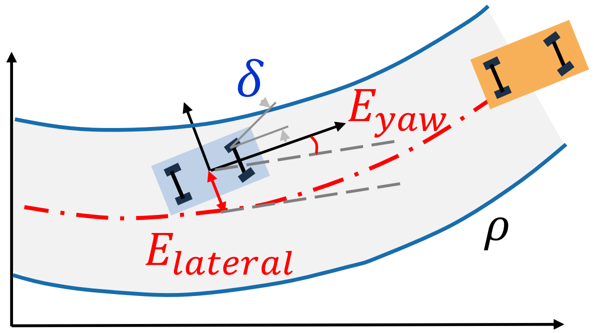
Components of lane following with spacing control algorithm



Goal

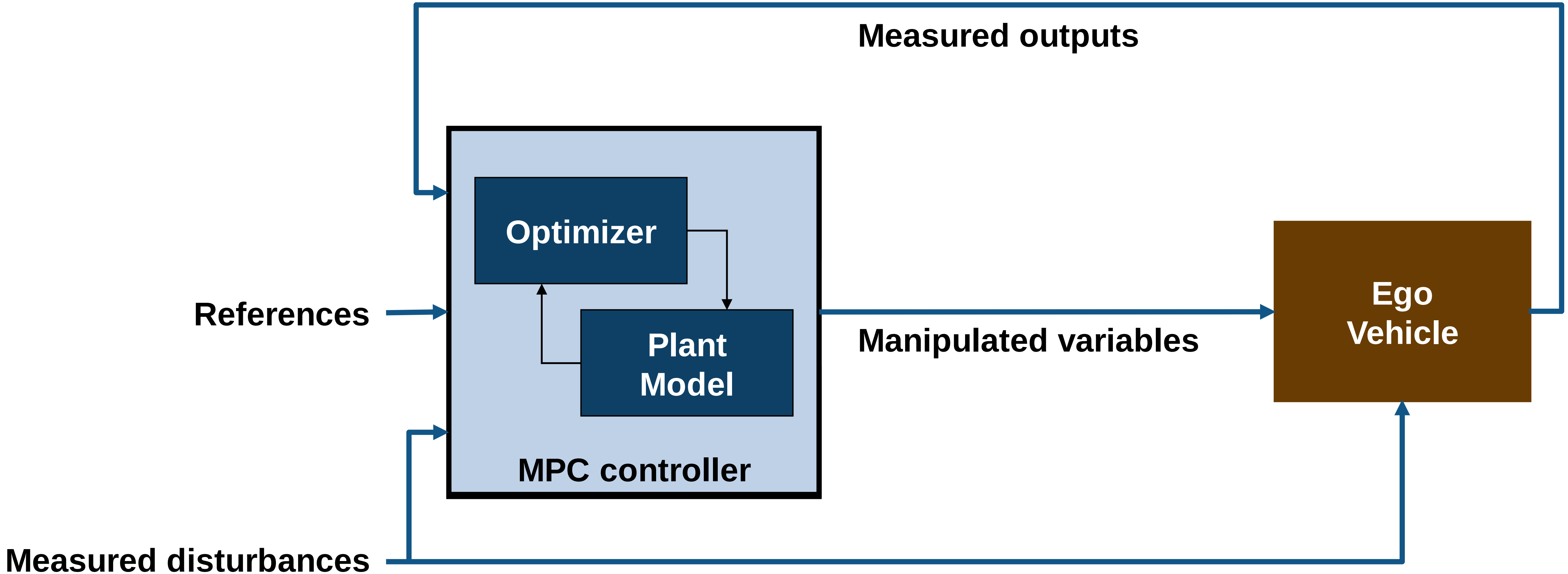
- Maintain the driver-set velocity and keep a safe distance from lead vehicle.

- Keep the ego vehicle in the middle of the lane.



- Slow down the ego vehicle when road is curvy.

Model predictive control (MPC)



MPC for Lane Following Control

minimize:

$$w_1|V_{ego} - V_{set}|^2 + w_2|E_{lateral}|^2$$

References
- Ego velocity set point (V_{set})
- Target lateral deviation (=0)

Measured disturbances
- MIO velocity (V_{mio})
- Previewed road curvature (ρ)

subject to:

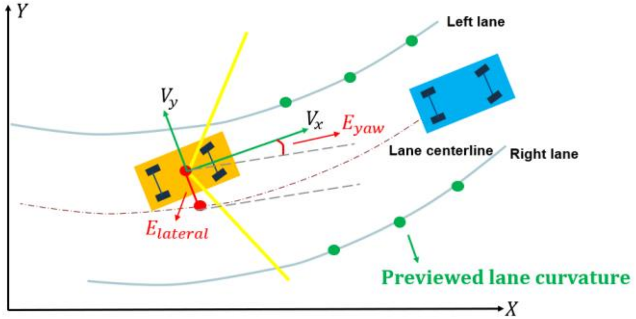
$$D_{relative} \geq D_{safe}$$
$$a_{min} \leq a \leq a_{max}$$
$$\delta_{min} \leq \delta \leq \delta_{max}$$

MPC controller
- Optimizer
- Ego Vehicle Model

Measured outputs
- Relative distance ($D_{relative}$)
- Ego velocity (V_{ego})
- Lateral deviation ($E_{lateral}$)
- Relative yaw angle (E_{yaw})

Manipulated variables
- Acceleration (a)
- Steering angle (δ)

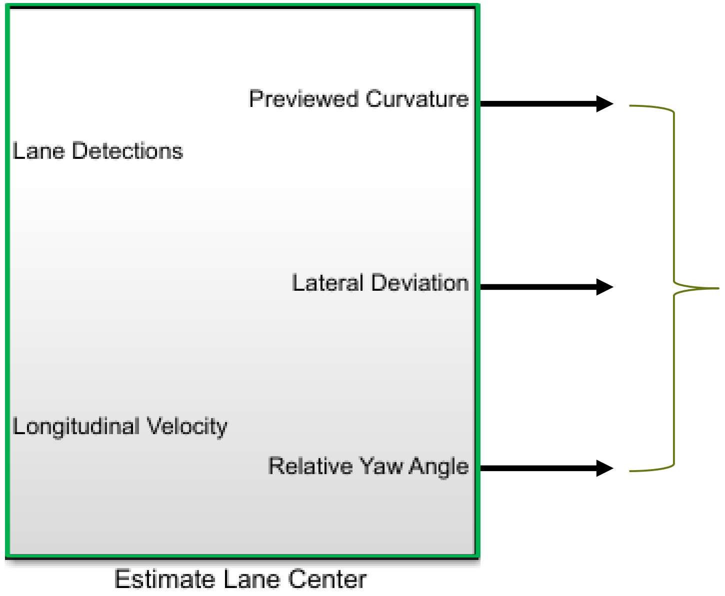
Ego Vehicle



Look ahead → Act early!

Components

- Estimate lane center

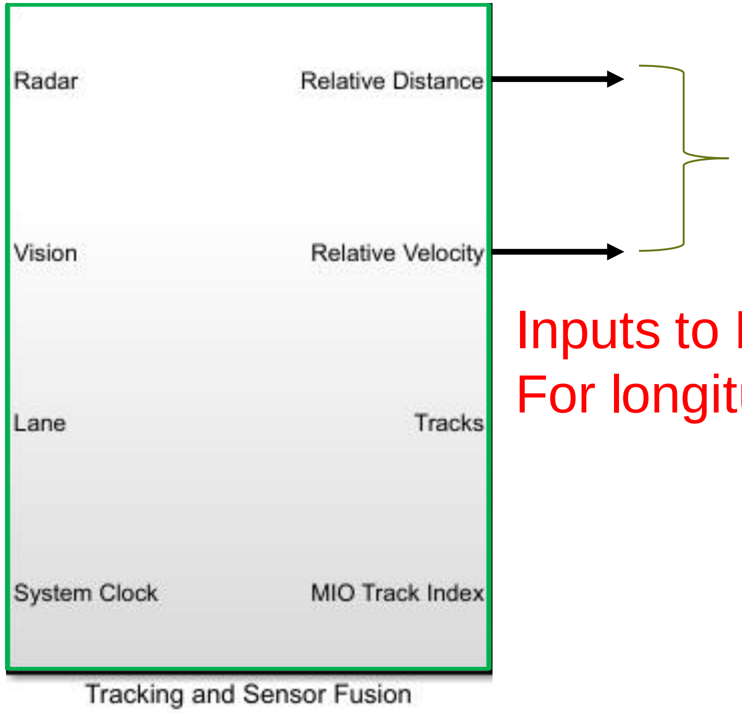


Inputs to MPC:
For lateral control

Four cases are considered:

1) Both left and right lanes are detected
2) Left lane is detected
3) Right lane is detected
4) No lane is detected

- Estimate MIO (lead vehicle)



Inputs to MPC:
For longitudinal control

- MPC: Path following controller

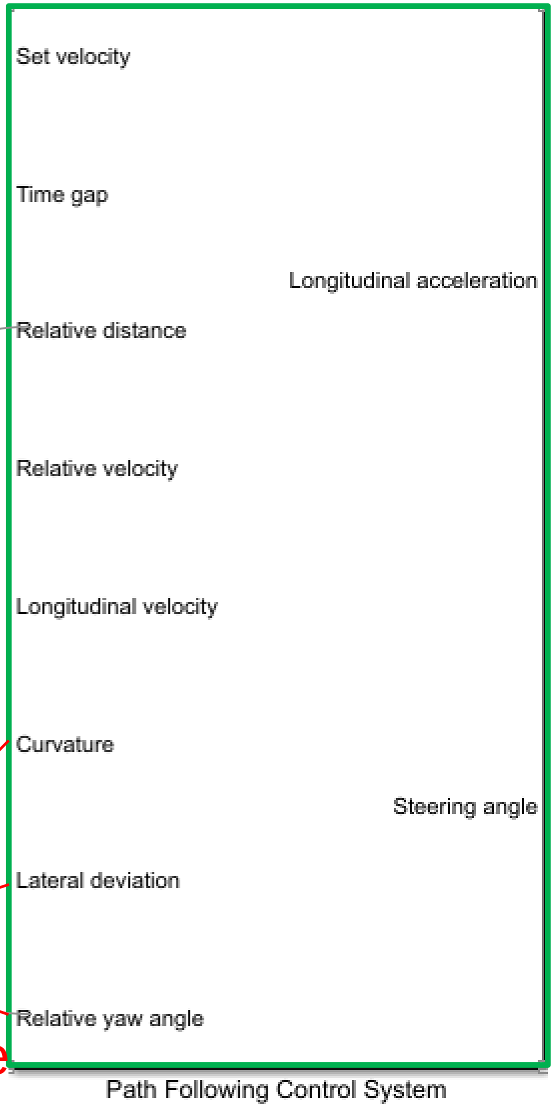
Path Following Control Block

Model Predictive Control Toolbox™ R2019b

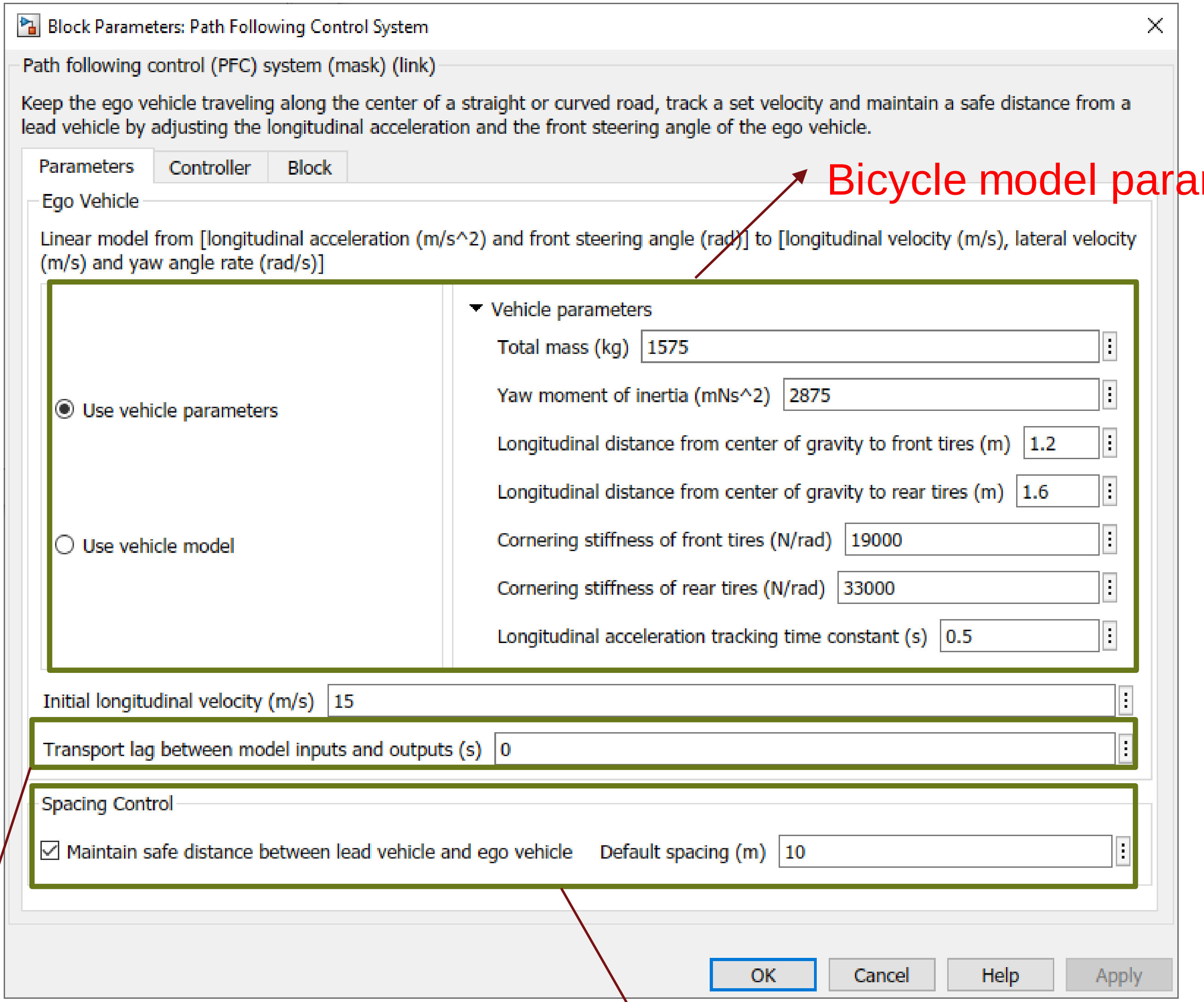
Driver setting

Measurement

Virtual lane
For lane change



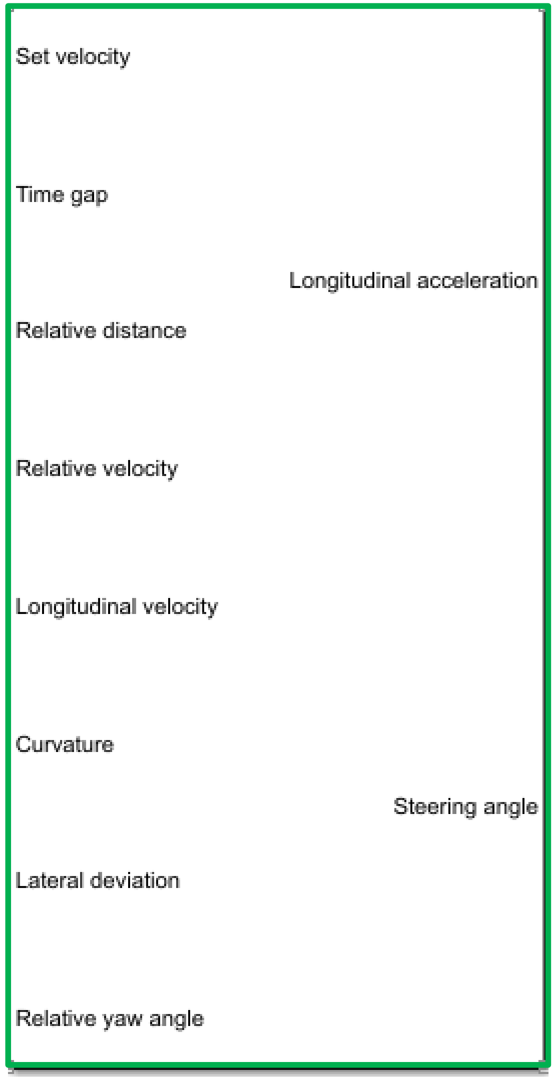
Bicycle model parameters

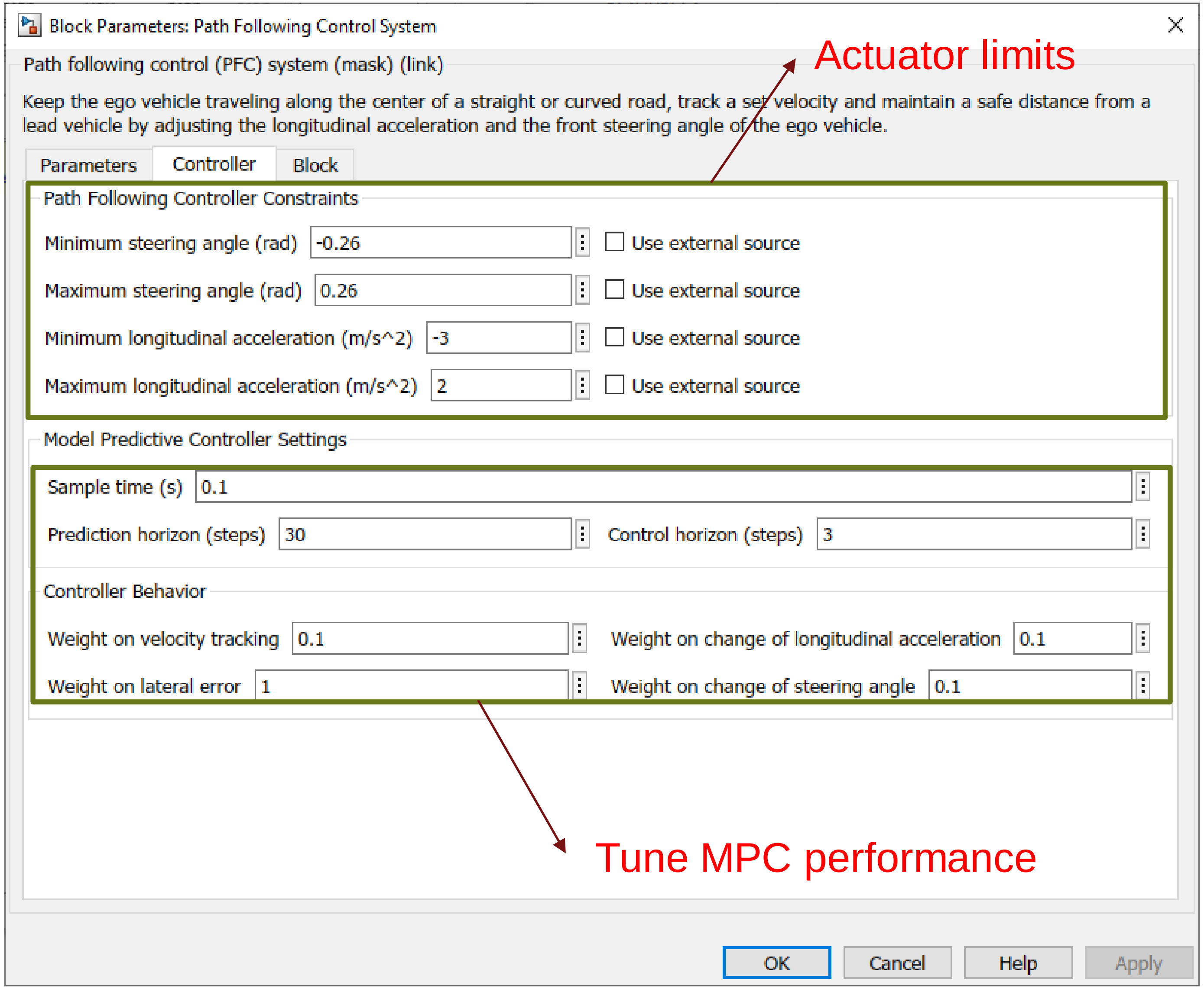


Delay or latency in the system

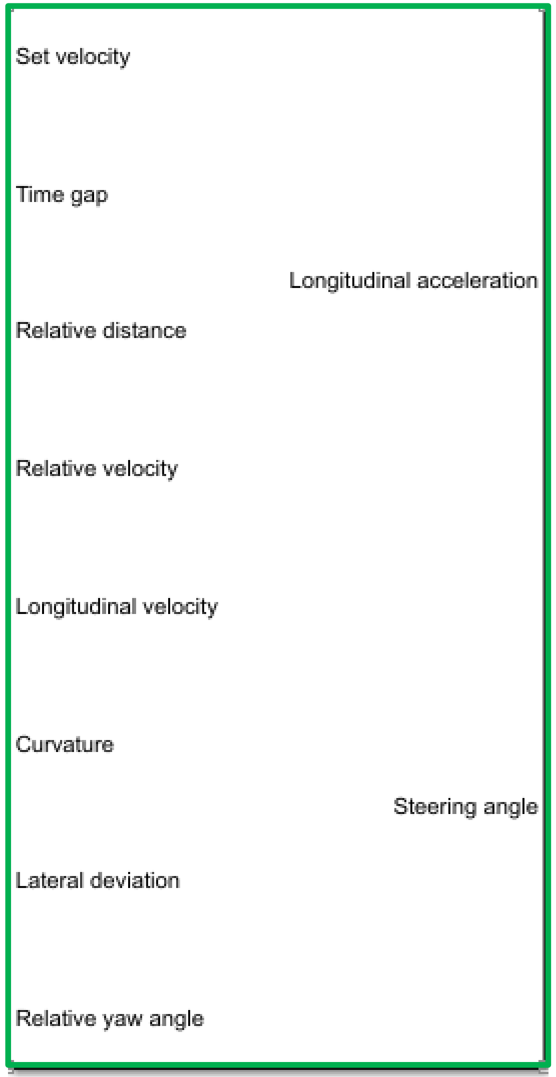
Disable distance keeping

Path Following Control Block

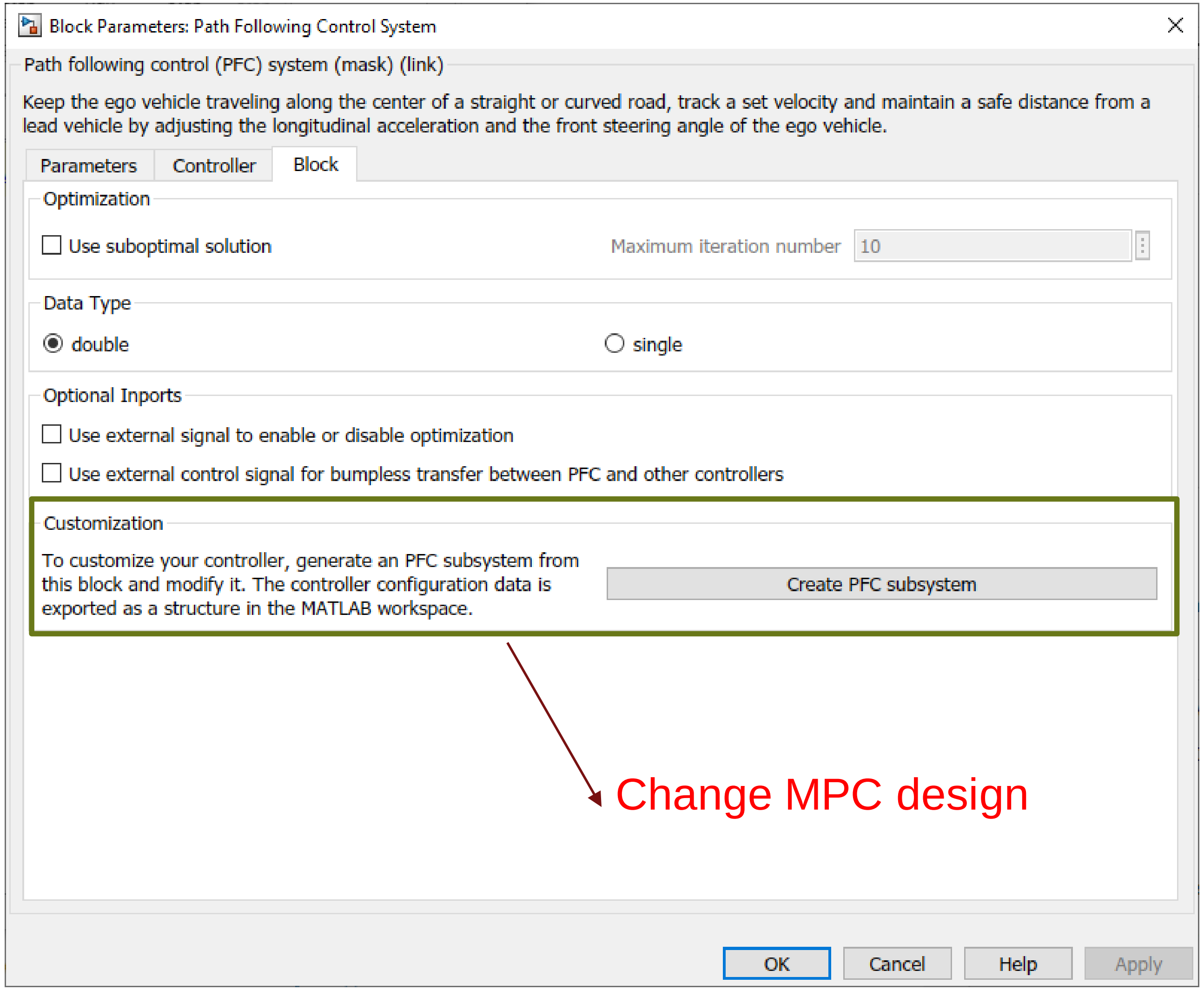




Path Following Control Block



Path Following Control System

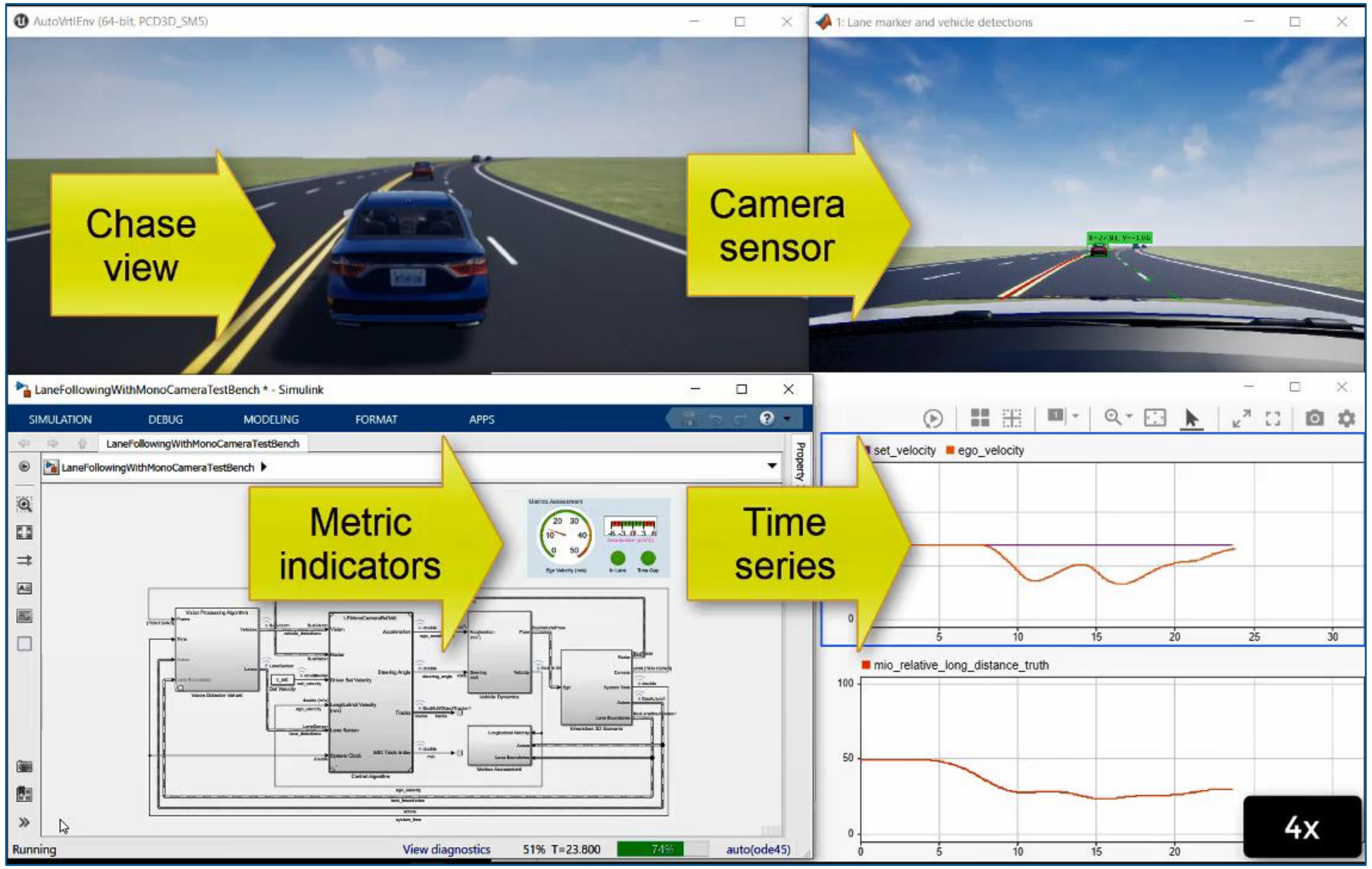


Change MPC design

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

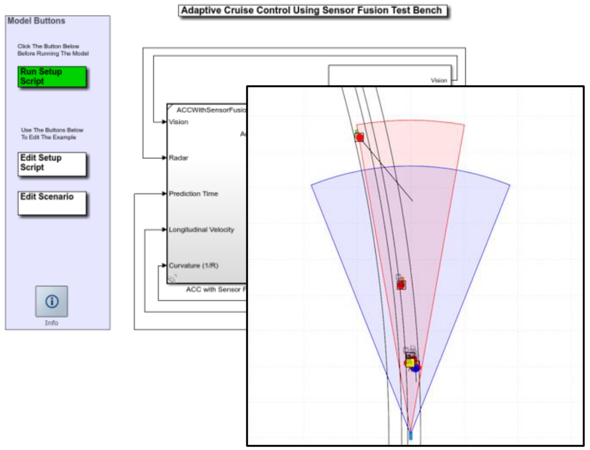


Model Predictive Control Toolbox™
Automated Driving Toolbox™
Vehicle Dynamics Blockset™

Updated R2019b

Design lateral and longitudinal Model Predictive Controllers

Longitudinal Control



[Adaptive Cruise Control with Sensor Fusion]()

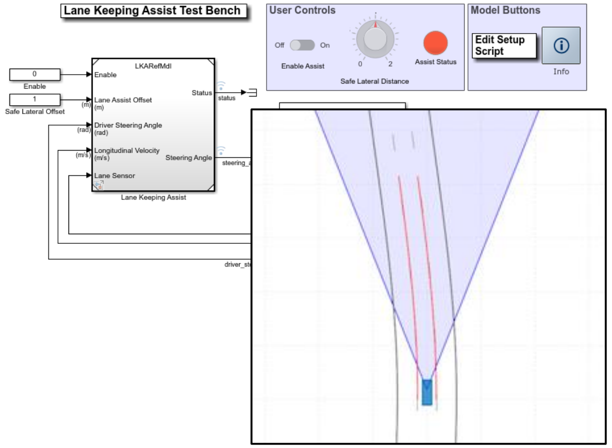
Automated Driving Toolbox™

Model Predictive Control Toolbox™

Embedded Coder®

R2017b

Lateral Control



[Lane Keeping Assist with Lane Detection]()

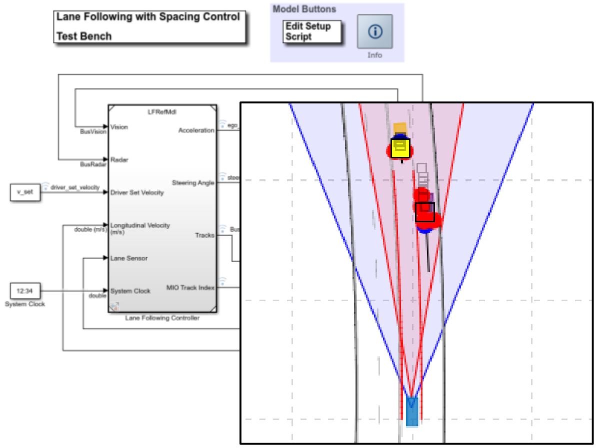
Automated Driving Toolbox™

Model Predictive Control Toolbox™

Embedded Coder®

R2018a

Longitudinal + Lateral



[Lane Following Control with Sensor Fusion and Lane Detection]()

Automated Driving Toolbox™

Model Predictive Control Toolbox™

Embedded Coder®

R2018b

Some common questions from automated driving engineers

How can I
synthesize scenarios
to test my designs?

Perception

Planning

Control

How can I
discover and design
in new domains?

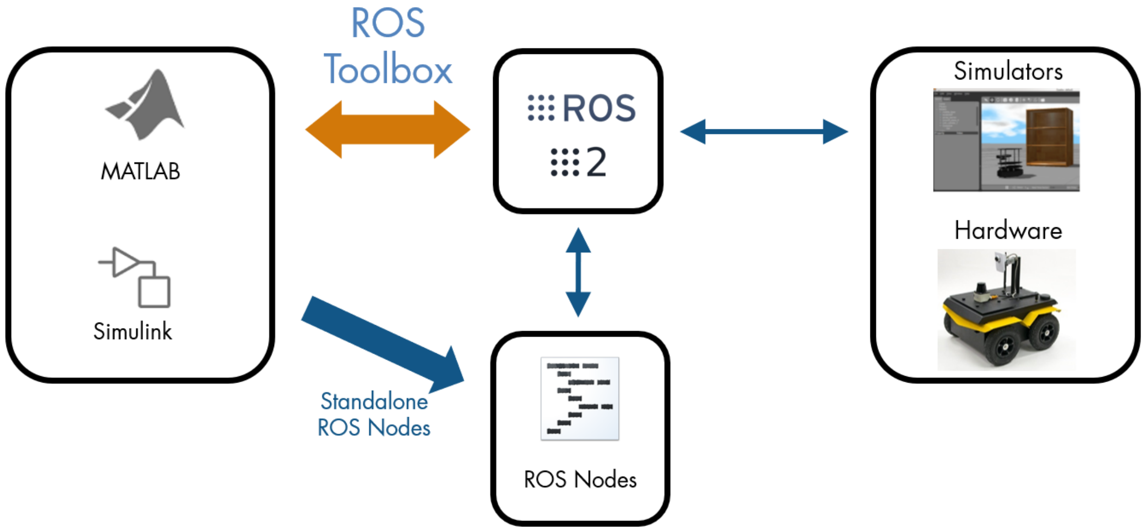
Simulation Integration

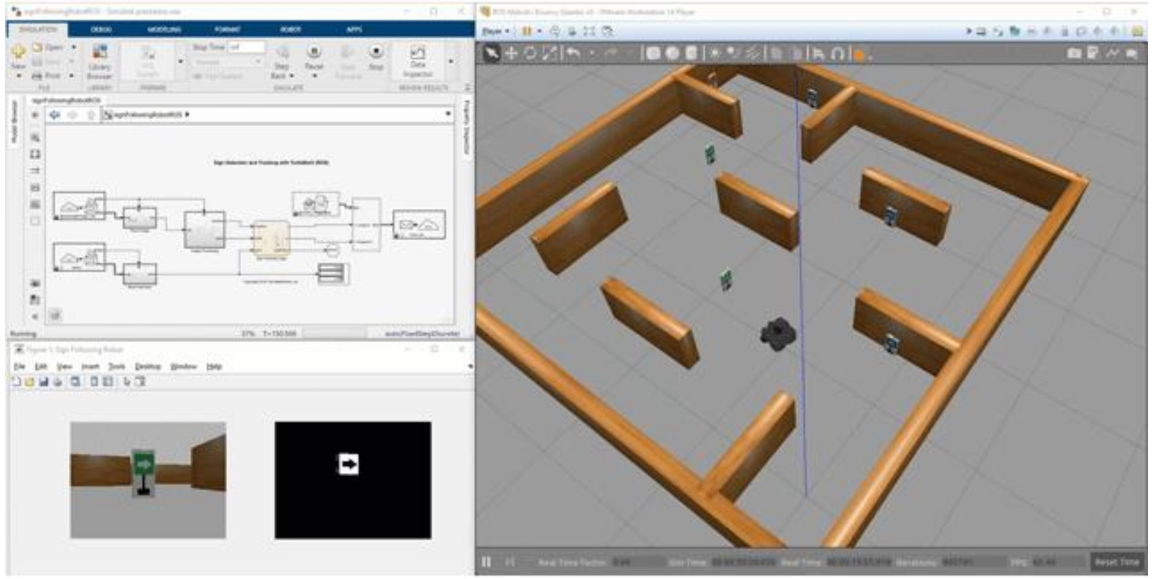
ROS
CAN
C/C++
Python
Cross Release
Third Party

How can I
integrate
with other environments?

ROS Toolbox - NEW!

- Communicate with ROS and ROS 2 nodes
- Multiplatform support
- Connect to live ROS data
- Replay logged data
- Generate standalone ROS nodes through code generation

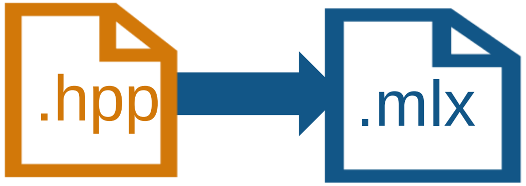




R2019b

Call C++, Python, and OpenCV from MATLAB

Call C++



[Import C++ Library Functionality into MATLAB]

MATLAB®

R2019a

Call Python

```
tw = ...
py.textwrap.TextWrapper(...
  pyargs(...
   'initial_indent',    '% ',...
   'subsequent_indent','% ',...
   'width', int32(30)))
```

[Call Python from MATLAB]

MATLAB®

R2014a

Call OpenCV & OpenCV GPU

```
cv::Rect
cv::KeyPoint
cv::Size
cv::Mat
cv::Ptr
...
```

[Install and Use Computer Vision Toolbox OpenCV Interface]

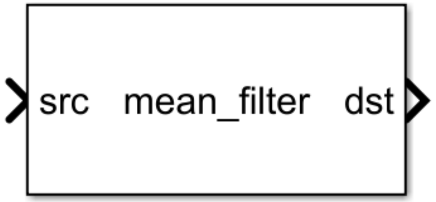
Computer Vision System Toolbox™

OpenCV Interface Support Package

Updated R2018b

Call C code from Simulink

Call C code



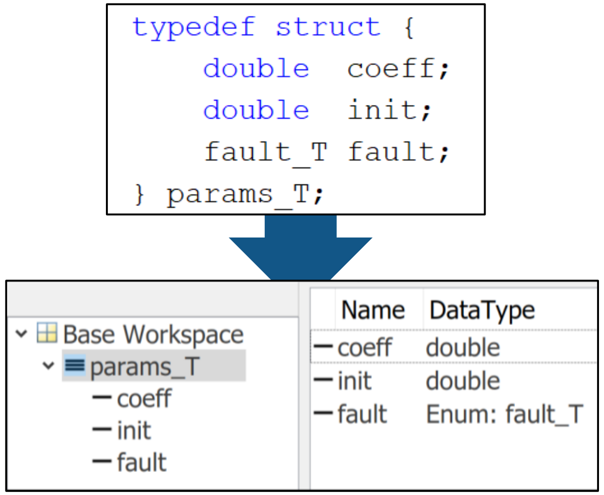
C Caller

[Bring Custom Image Filter Algorithms as Reusable Blocks in Simulink]

Simulink®

R2017b

Create buses from C structs

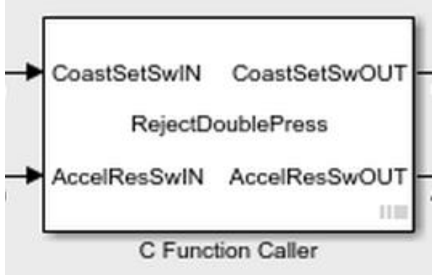


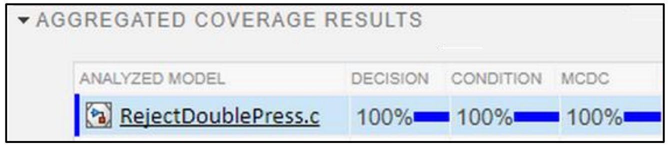
[Import Structure and Enumerated Types]

Simulink®

R2017a

Test and verify C code





[Custom C Code Verification with Simulink Test]

Simulink Test™

Simulink Coverage™

R2019a

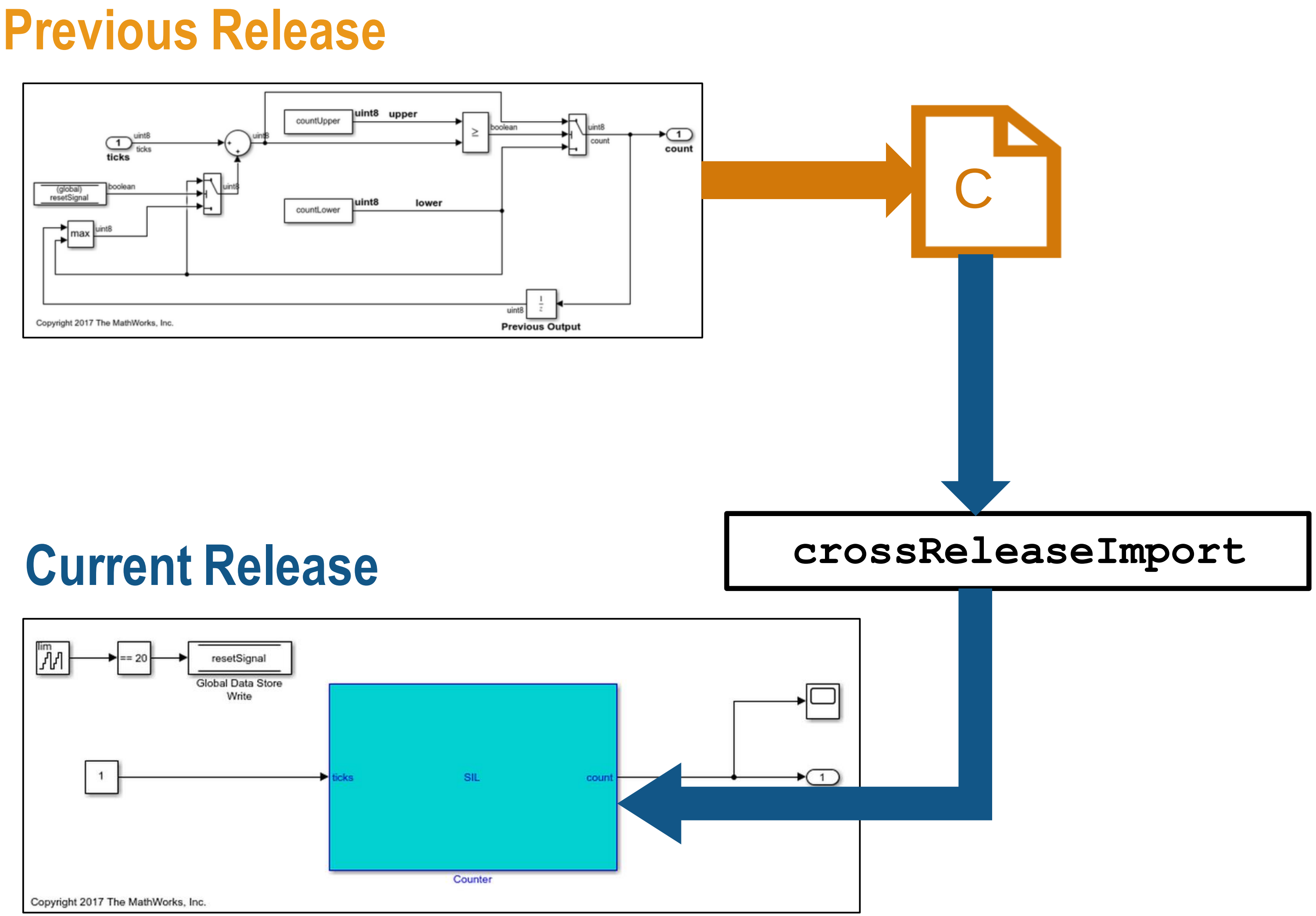
Cross-release simulation through code generation

[Integrate Generated Code by Using Cross-Release Workflow]

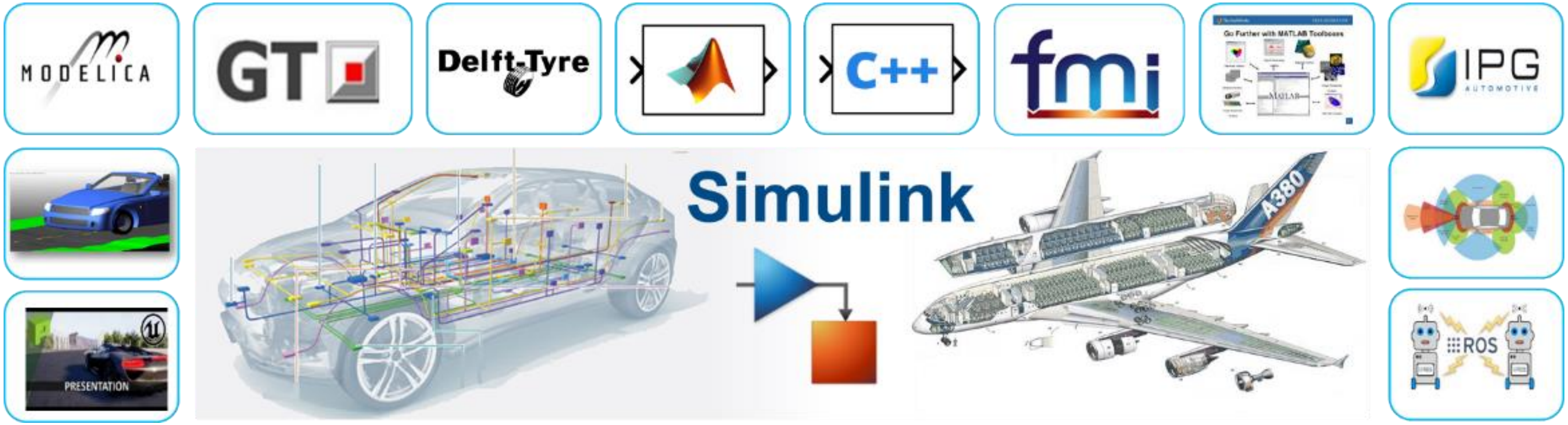
- Generate code from previous release (R2010a or later)
- Import generated code as a block in current release
- Tune parameters
- Access internal signals

Embedded Coder

R2016a



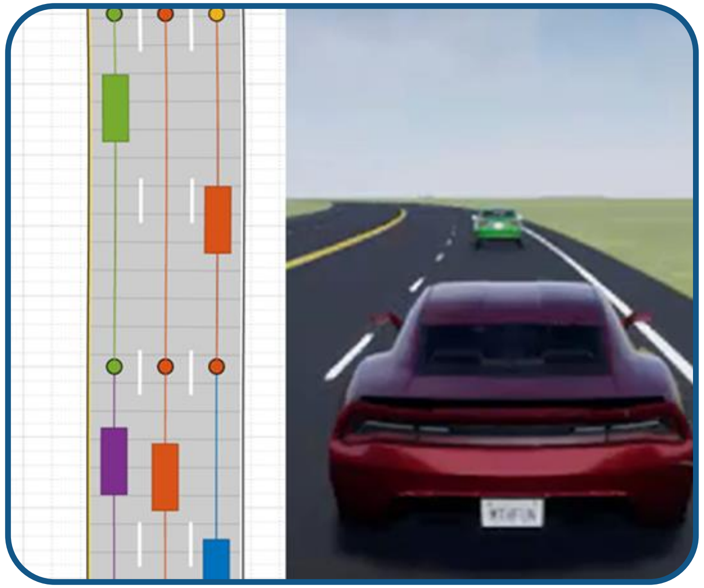
Connect to third party tools



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

MathWorks
Partner

Some common questions from automated driving engineers



Perception

Planning

Control

Simulation Integration

ROS

CAN

C/C++

Python

Cross Release

Third Party

Synthesize scenarios
to test my designs

Discover and design
in multiple domains

Integrate
with other environments

Thank You!