Masterclass

Scene and Scenario design for ADAS Simulation

October 20, 2022 | Stuttgart

Maxime François
Peter Fryscak

MathWorks

Agenda

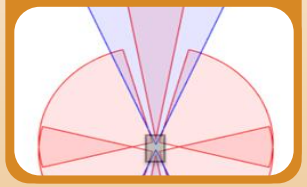
1. Introduction
2. Testing ADAS Systems
3. Cuboid scene and trajectory-driven scenario design
4. Photorealistic scene and logic-driven scenario design
5. Conclusion

Industry trends drive MathWorks investments

Virtual Worlds
are required for
design and verification

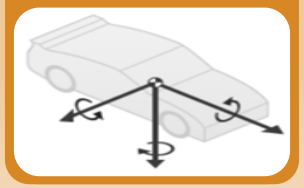
Scenes

Sensors



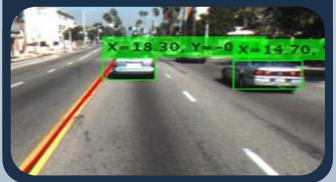
Scenarios

Dynamics

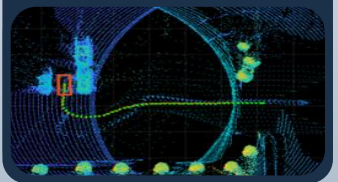


Multidisciplinary Skills
are needed as teams are expected to
learn and apply new disciplines

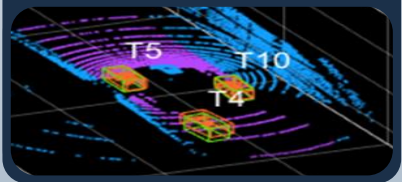
Detection



Localization

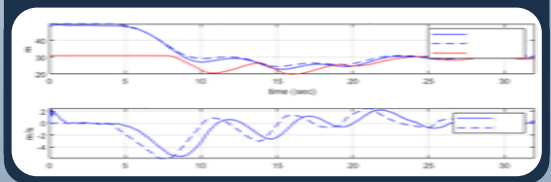


Tracking



Planning

Decision & Controls



Software
Safety, Security,
Agility at scale

Code

```
C/C++
GPU HDL
```

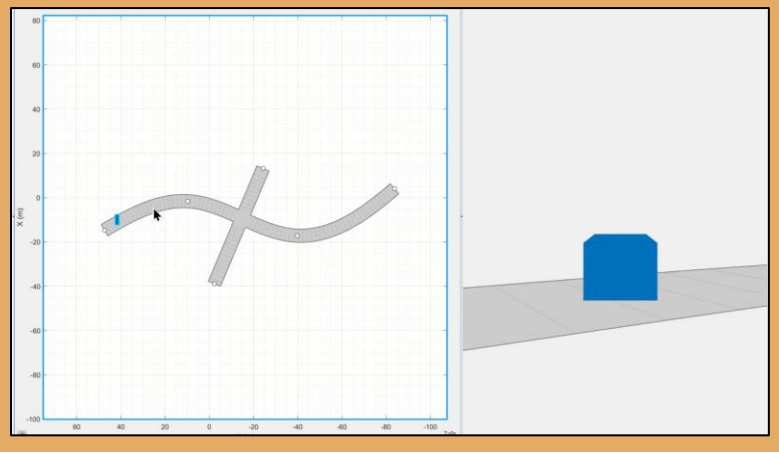
Architectures

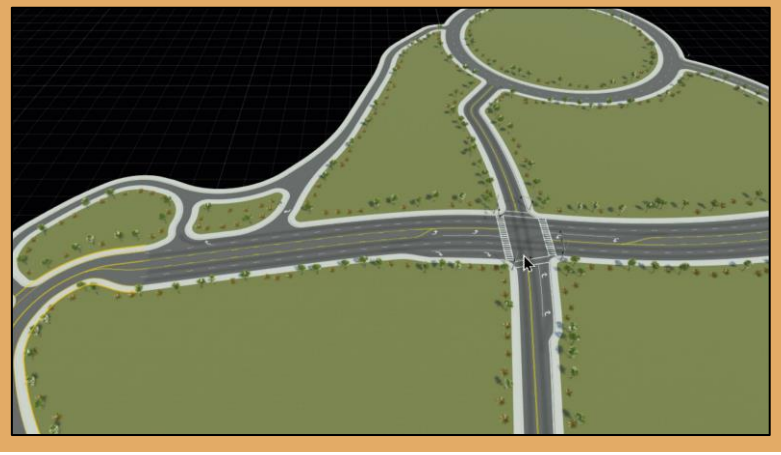
```
AUTOSAR
ROS DDS
```

RoadRunner product family	**MATLAB & Simulink** product family	**Polyspace** product family

Scene and scenario design answers a need

Scenes





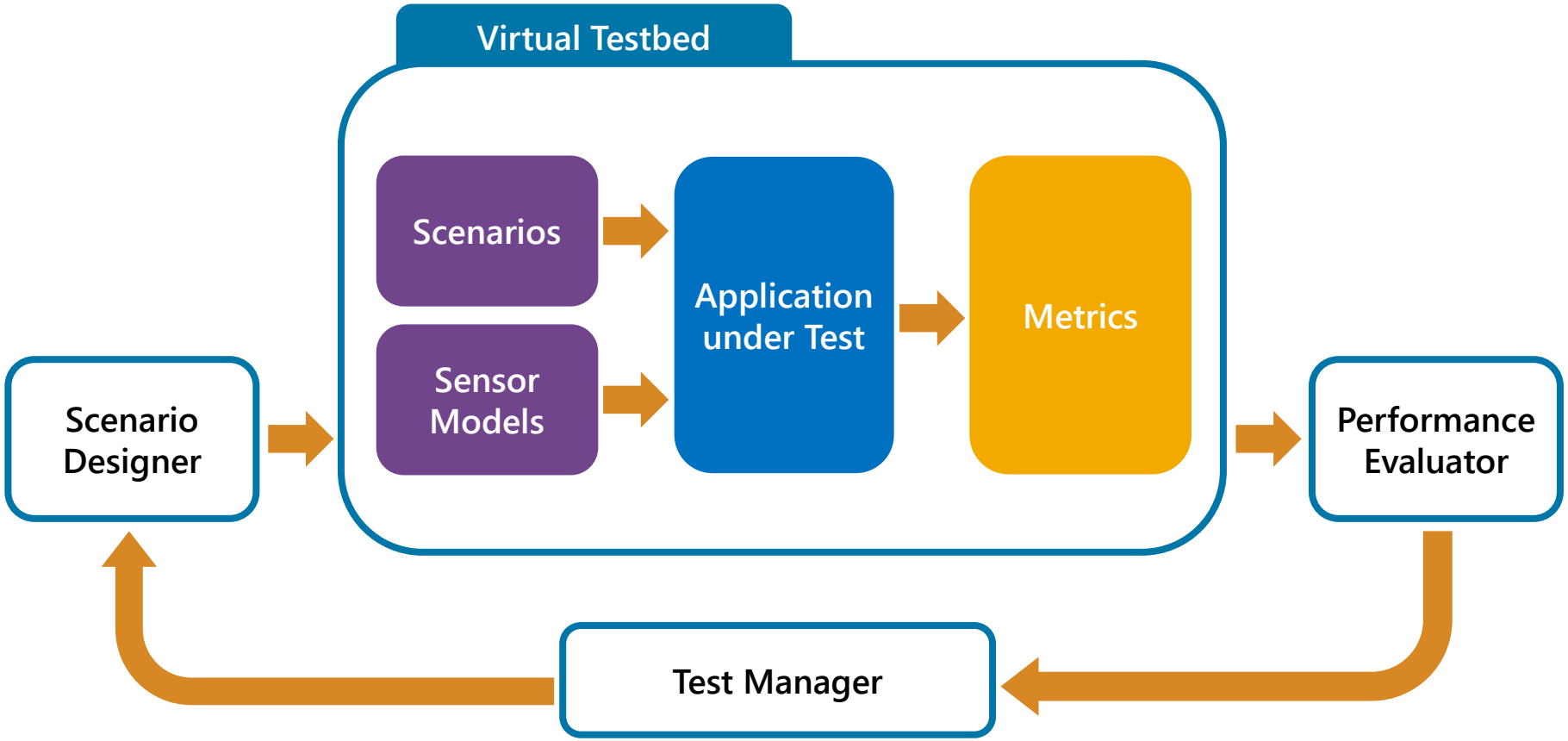
Scenarios

Complexity and realism

Agenda

1. Introduction
2. Testing ADAS Systems
3. Cuboid scene and trajectory-driven scenario design
4. Photorealistic scene and logic-driven scenario design
5. Conclusion

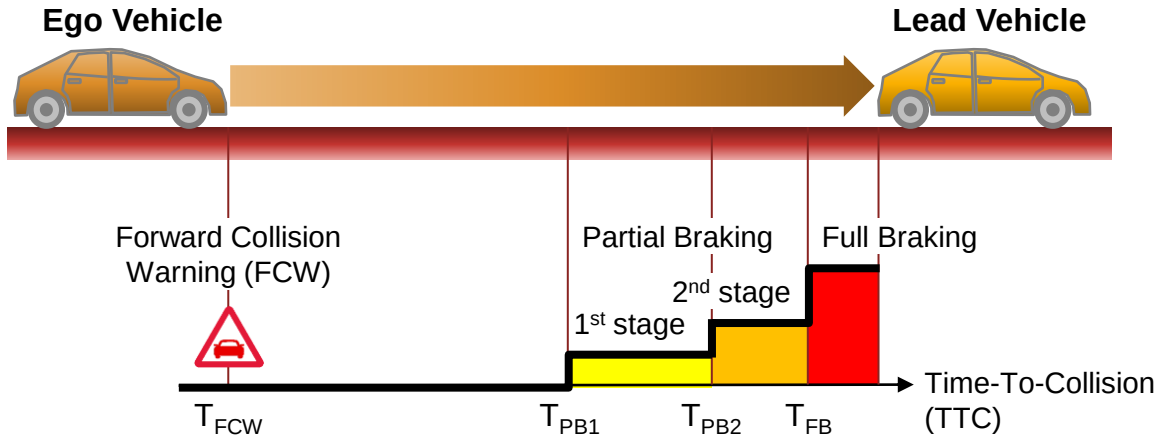
Virtual testing framework for autonomous systems

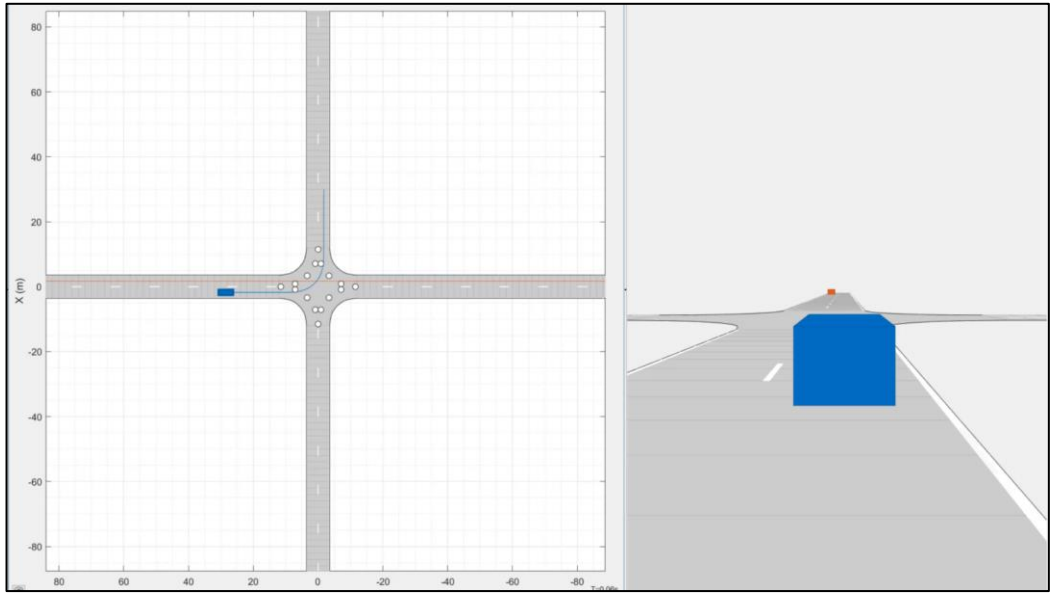


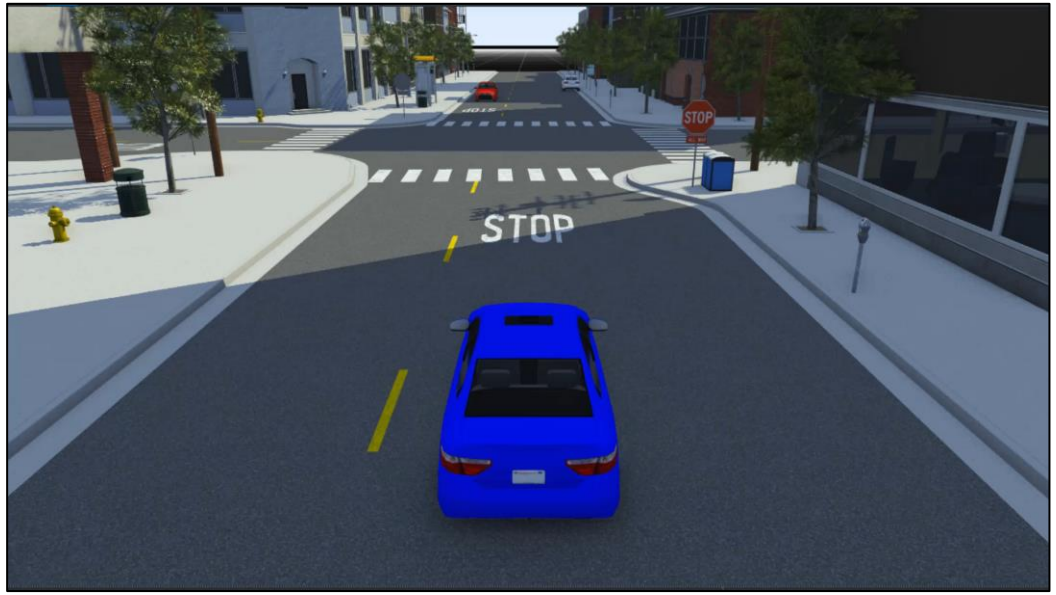
ADAS system under test

Autonomous Emergency Braking (AEB)

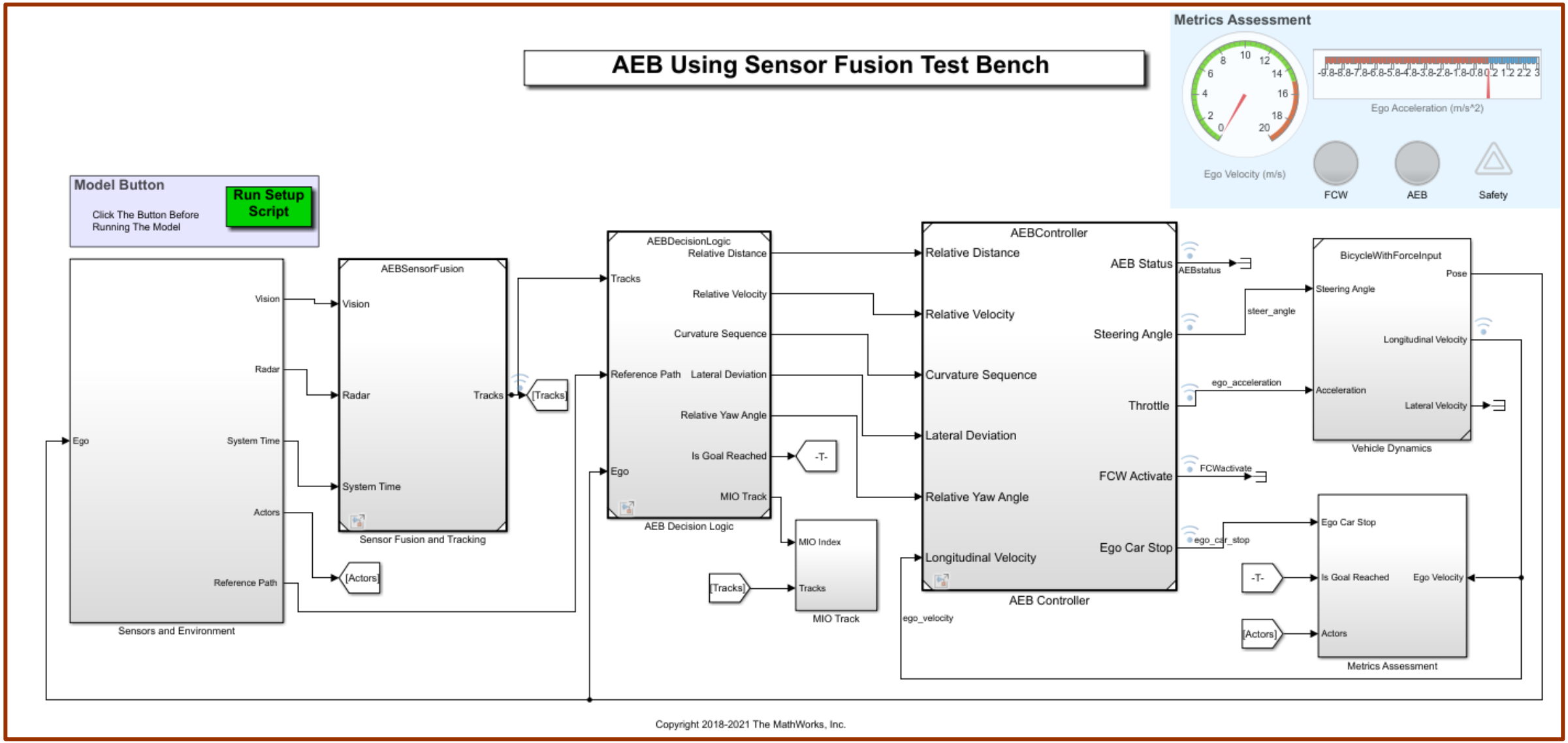
- If driver fails to apply brakes in time, AEB system engages automatically to avoid or mitigate collision



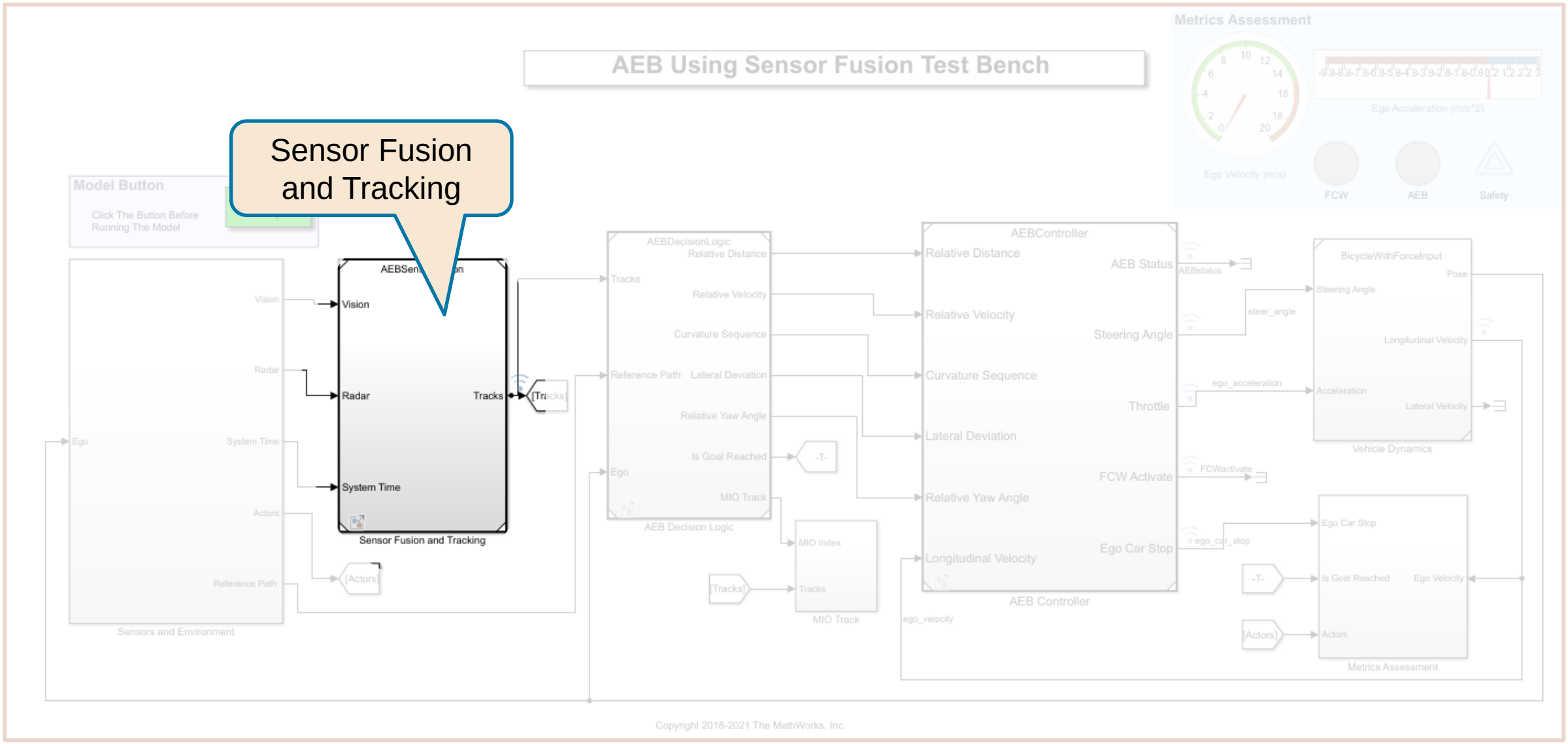




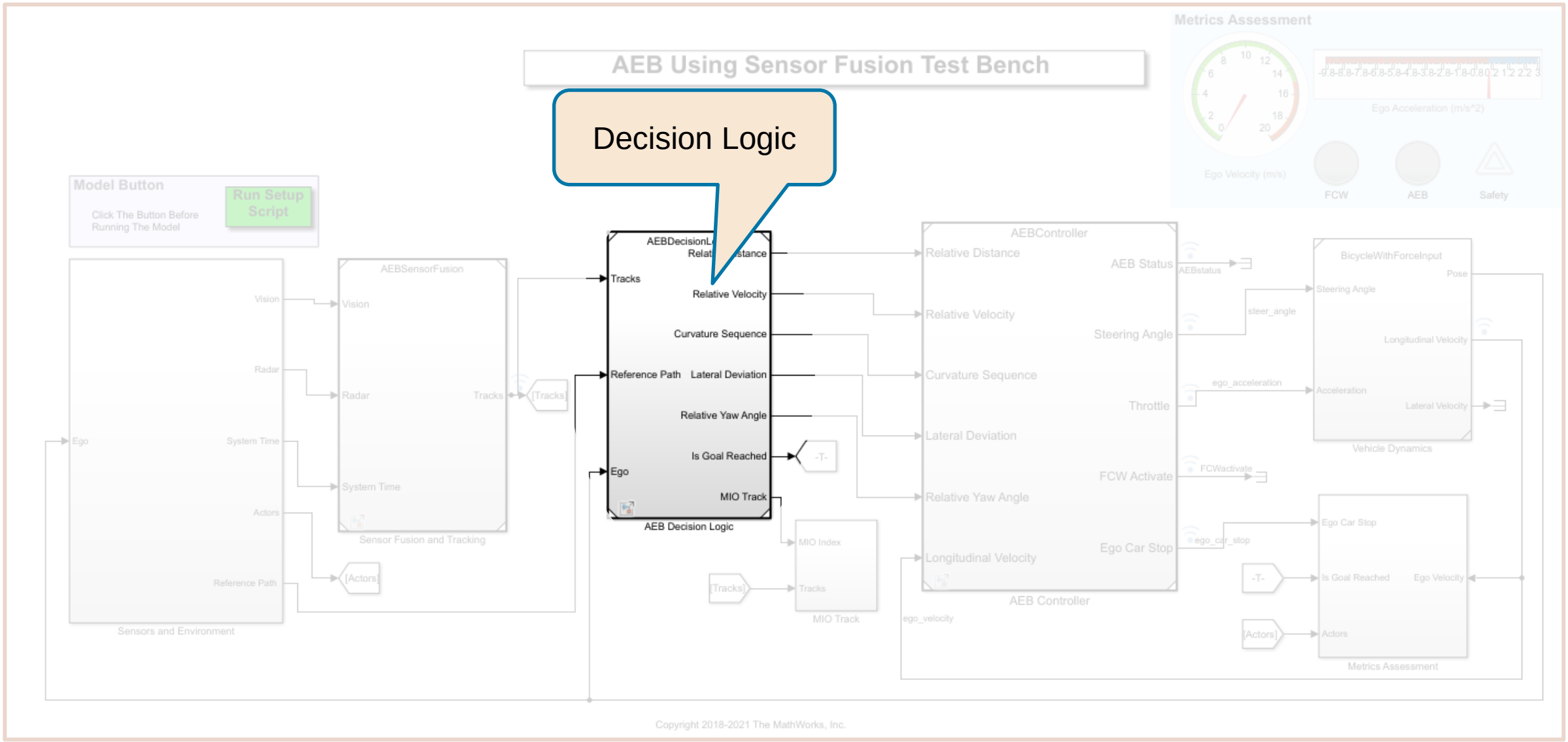
Explore test bench model



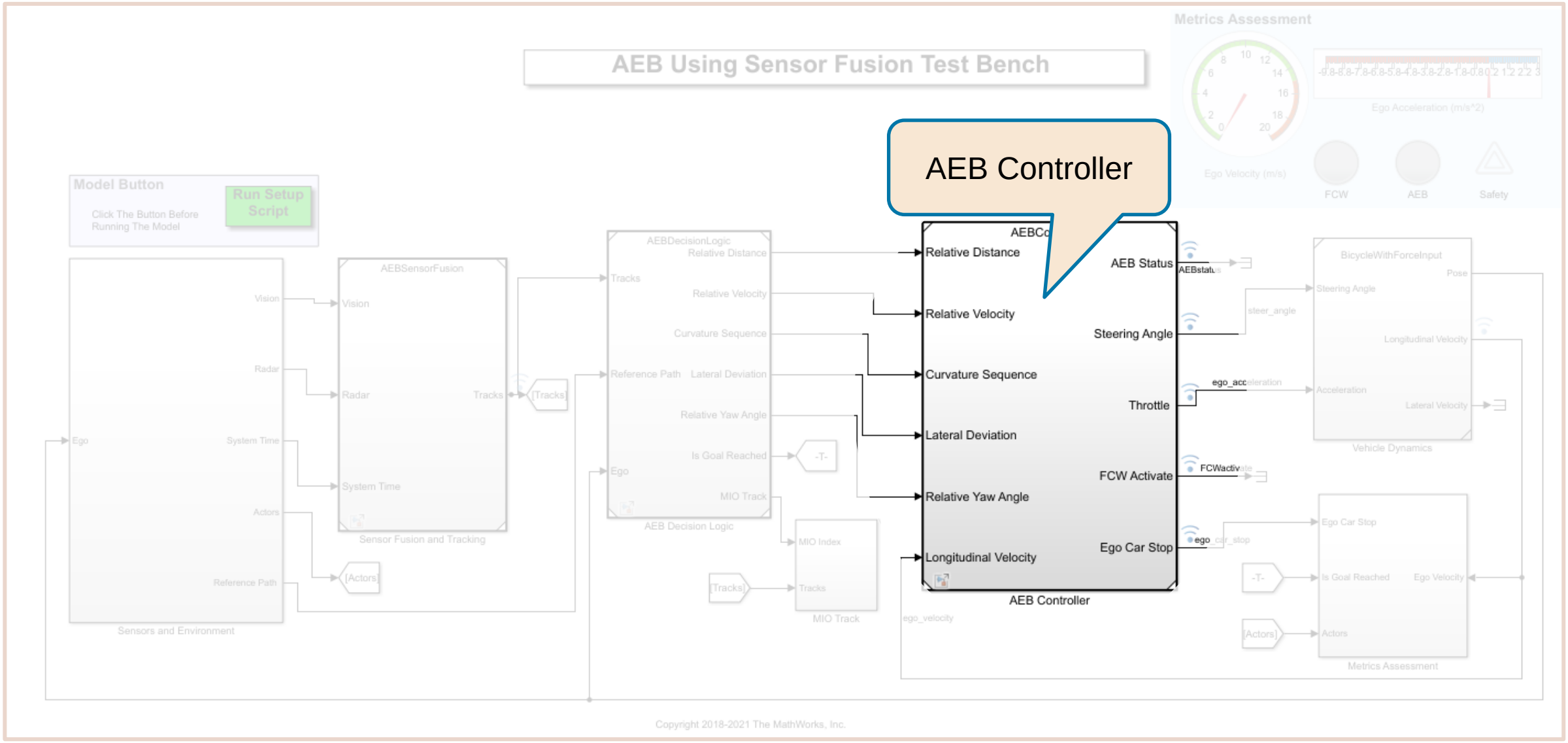
Explore test bench model



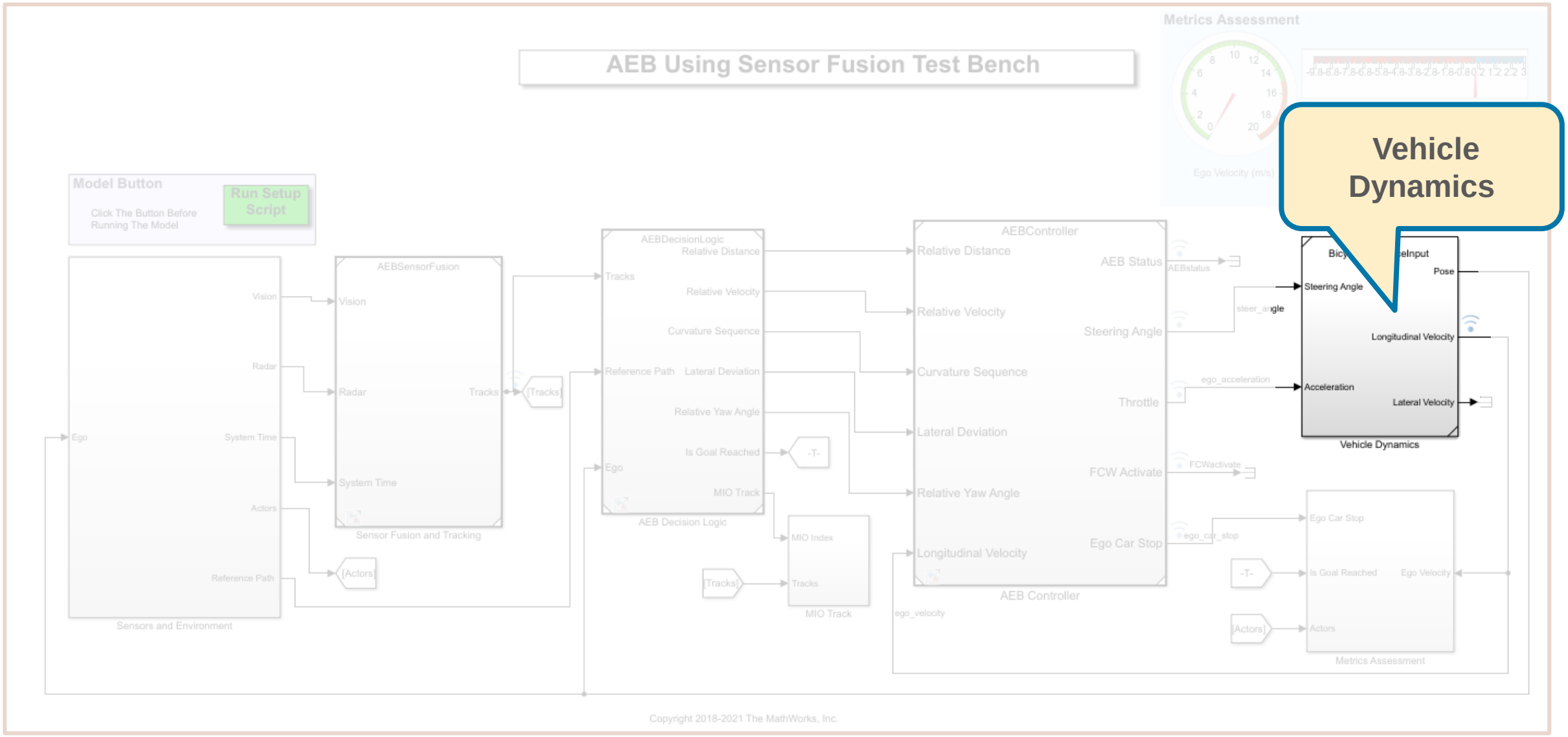
Explore test bench model



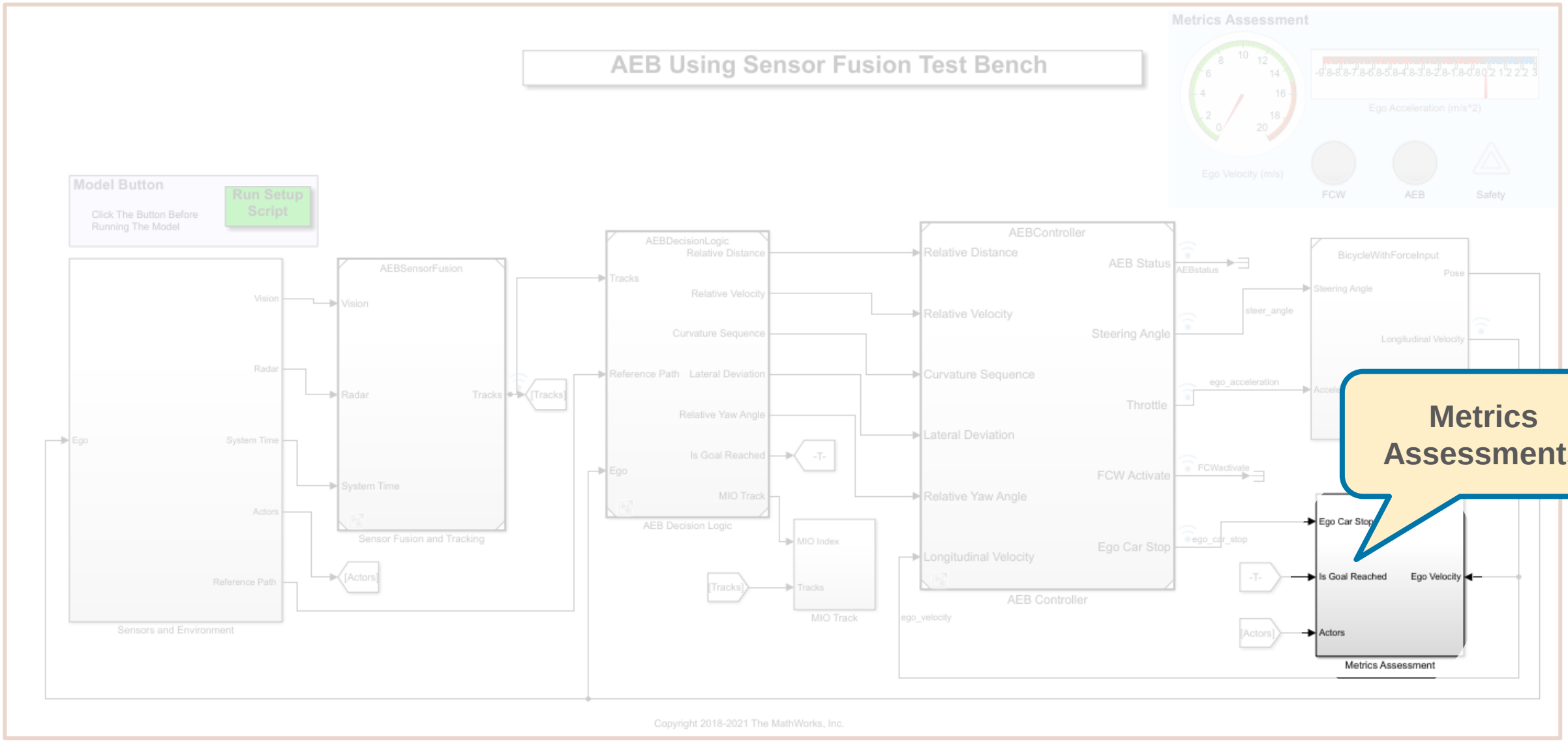
Explore test bench model



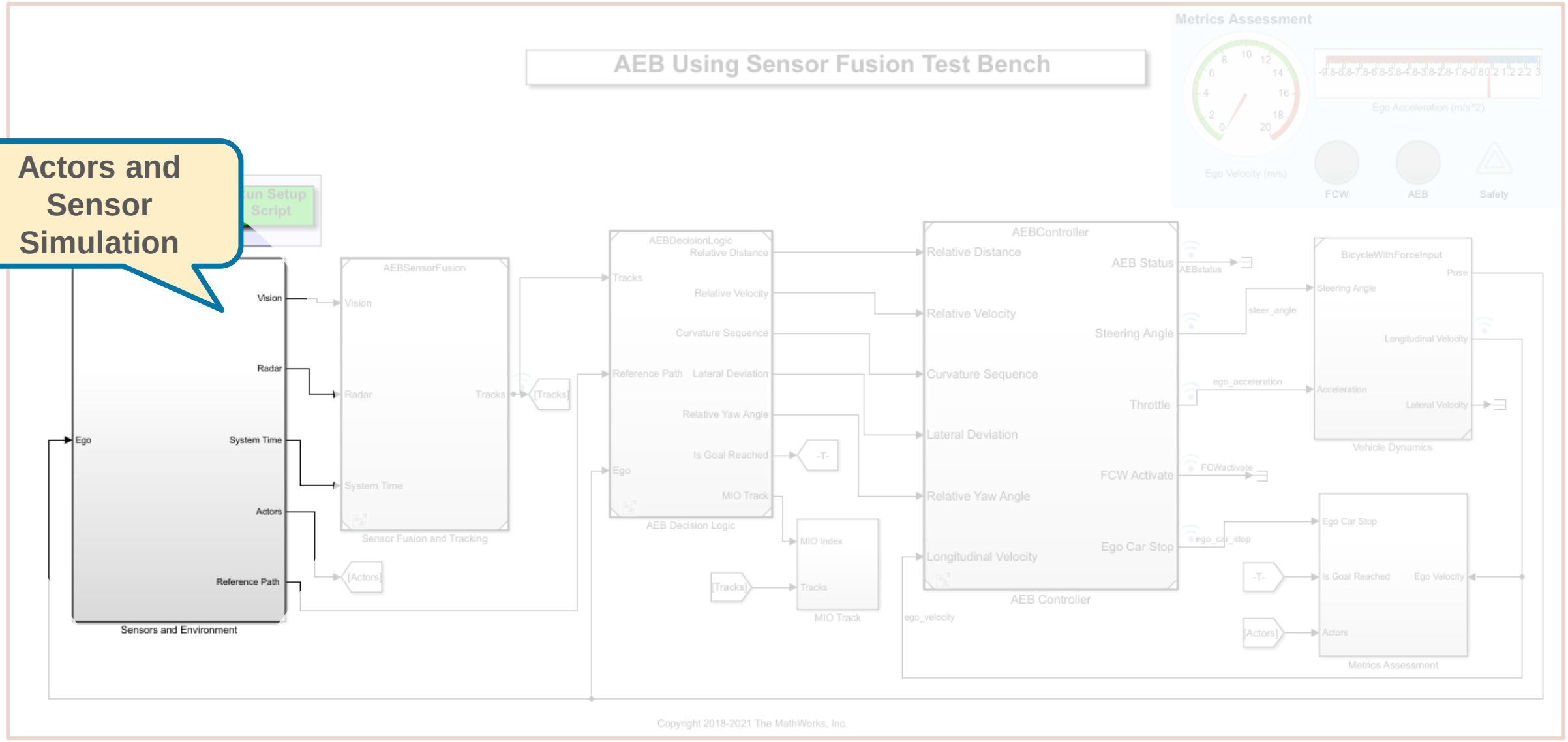
Explore test bench model



Explore test bench model



Explore test bench model



Agenda

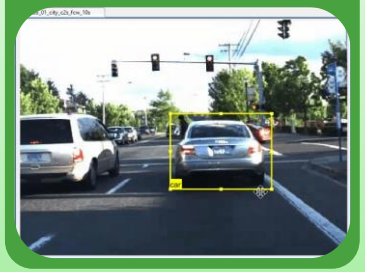
1. Introduction
2. Testing ADAS Systems
3. Cuboid scene and trajectory-driven scenario design
4. Photorealistic scene and logic-driven scenario design
5. Conclusion

Automated Driving Toolbox

Overview

Labeling

Ground Truth

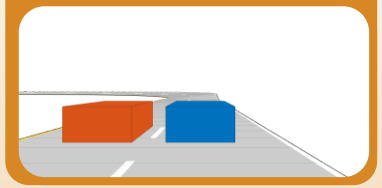


I/O

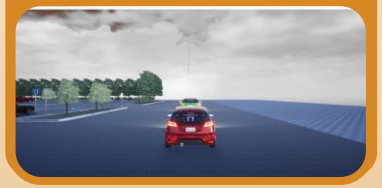
Read Data

Simulation

Cuboid



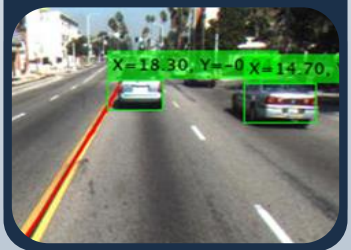
Unreal Engine



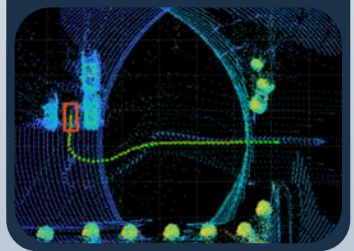
RoadRunner

Algorithms

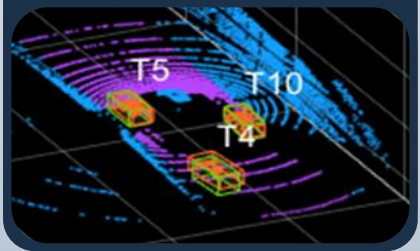
Detection



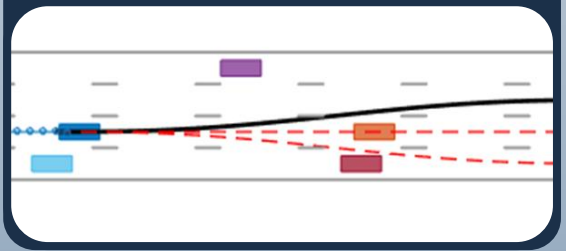
Localization



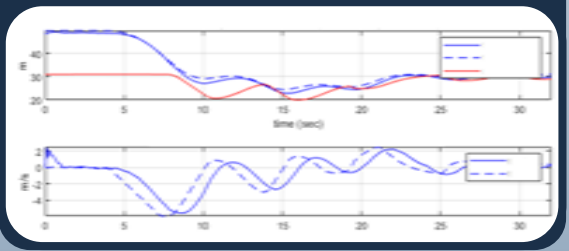
Tracking



Planning



Decision & Controls



Automated Driving Toolbox

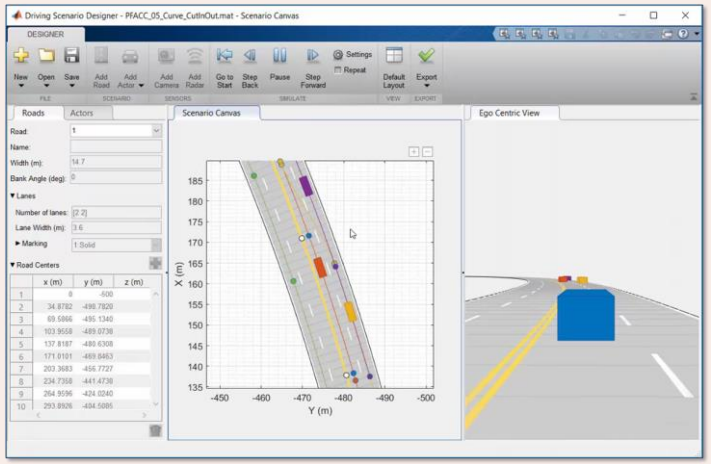
Cuboid Simulation

OpenDRIVE
OpenStreetMap
HERE HD Maps
Zenrin SD

Custom meshes

Design Driving Scenarios

Driving Scenario Designer App



Driving Scenario API

```
ds = drivingScenario();
```

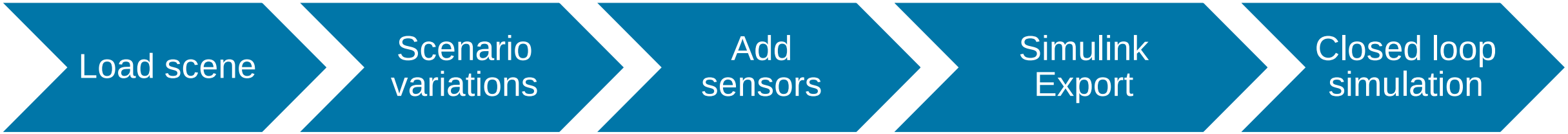
Driving scenario
- Roads
- Lane markings
- Barriers & guardrails
- Vehicle trajectories

OpenDRIVE

OpenSCENARIO

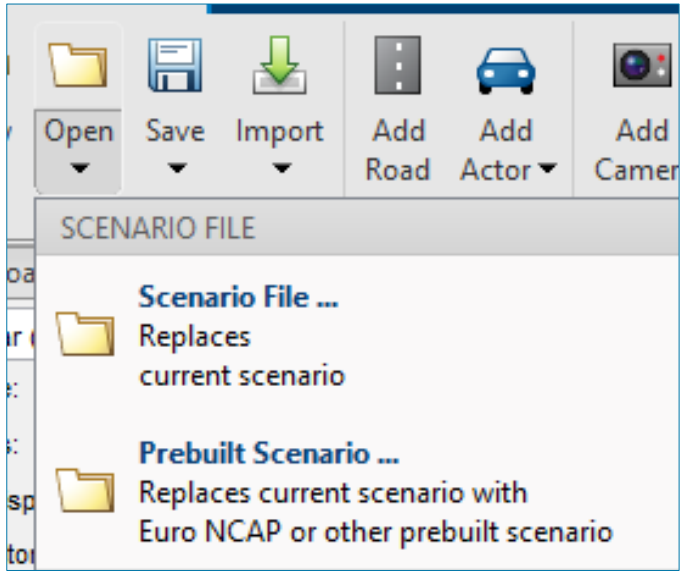
Presentation workflow

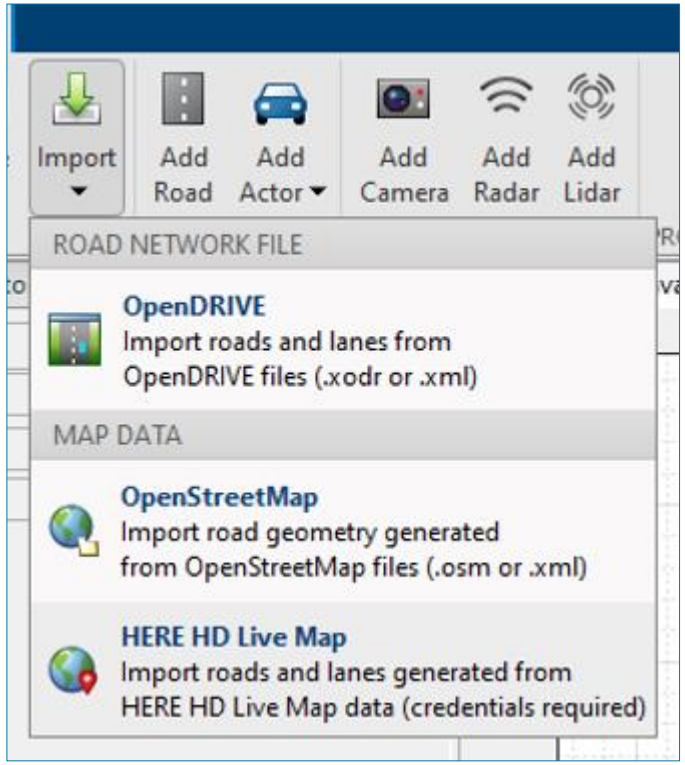
Using the Automated Driving Toolbox



Load a pre-built scene into the app

Load scene → Scenario variations → Add sensors → Simulink Export → Closed loop simulation



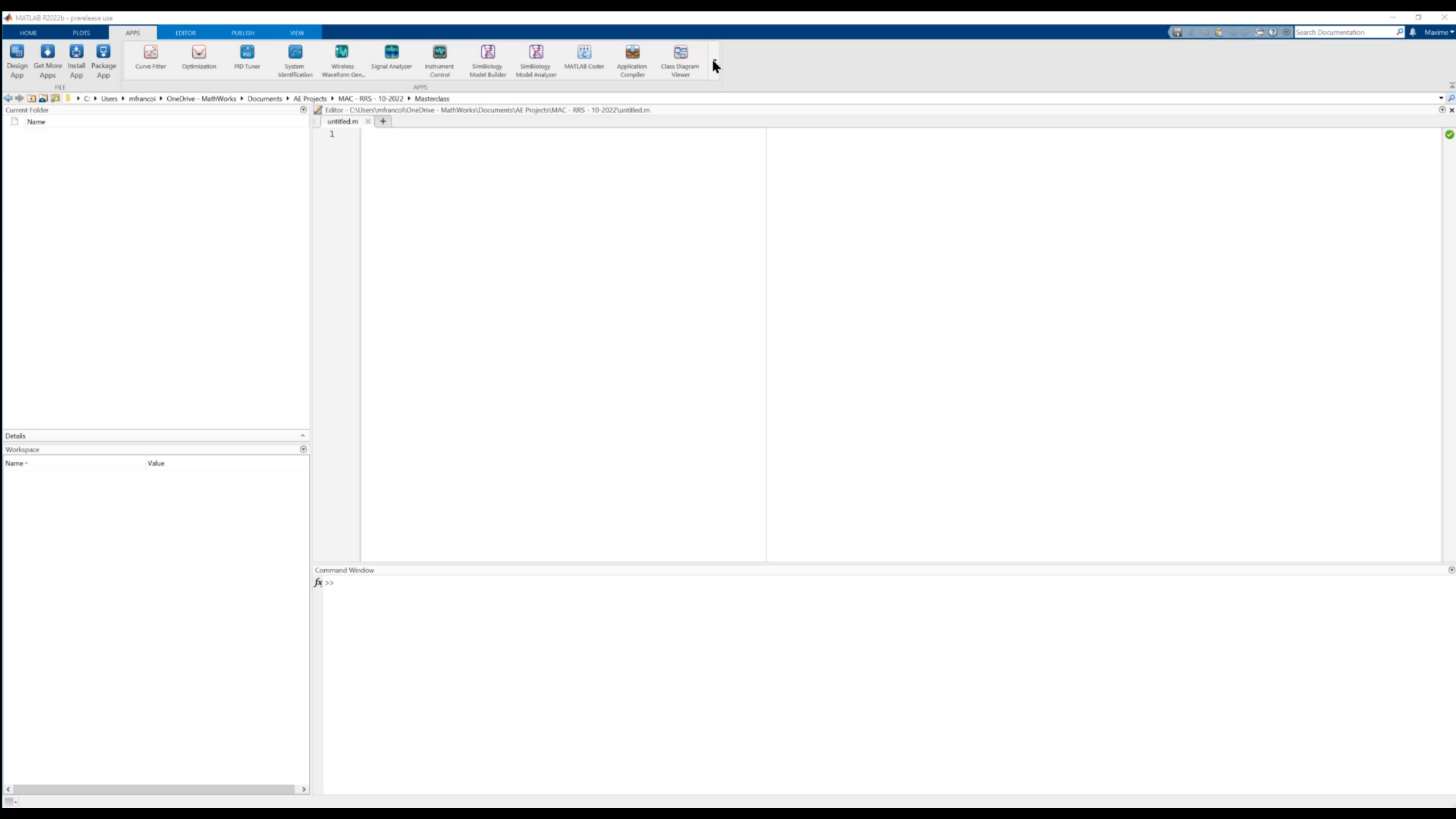


FOR SAFER CARS EURO NCAP

OpenDRIVE® managing the road ahead

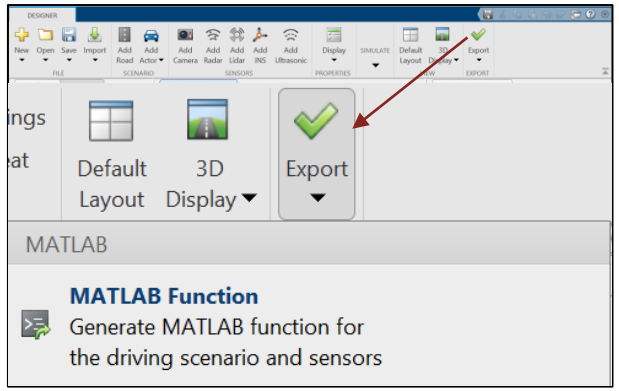
OpenStreetMap

here

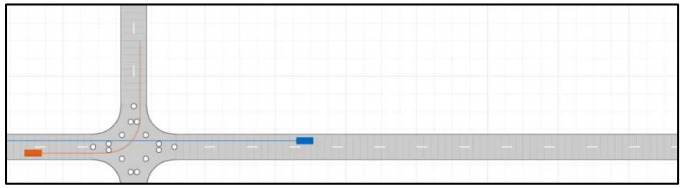

MATLAB R2022b - prerelease use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Maxime
Design App
Get More Apps
Install App
Package App
FILE
Curve Fitter
Optimization
PID Tuner
System Identification
Wireless Waveform Gen...
Signal Analyzer
Instrument Control
SimBiology Model Builder
SimBiology Model Analyzer
MATLAB Coder
Application Compiler
Class Diagram Viewer
APPS
C: ▸ Users ▸ mfrancoi ▸ OneDrive - MathWorks ▸ Documents ▸ AE Projects ▸ MAC - RRS - 10-2022 ▸ Masterclass
Current Folder
Name
Editor - C:\Users\mfrancoi\OneDrive - MathWorks\Documents\AE Projects\MAC - RRS - 10-2022\untitled.m
untitled.m
1
Details
Workspace
Name
Value
Command Window
>>

Programmatic scenario variations

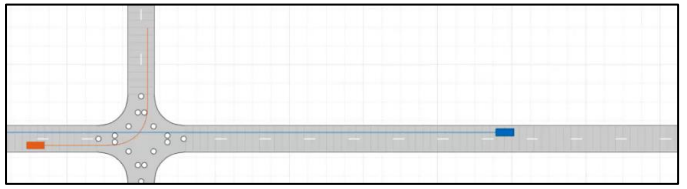
Load scene → Scenario variations → Add sensors → Simulink Export → Closed loop simulation



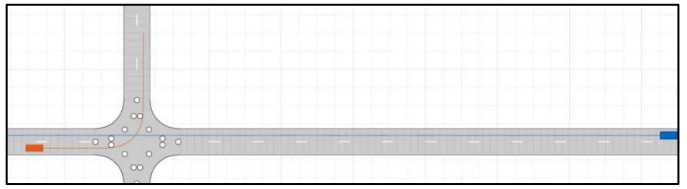

```
function [scenario, egoVehicle] = AEB_CCFtap_VUT_10kph_GVT_30kph(nonEgoSpeed)
% createDrivingScenario Returns the drivingScenario defined in the Designer

% Constant for all AEB scenarios
collisionTime = 10.0;
nonEgoInitialPosition = -nonEgoSpeed*collisionTime;

% Construct a drivingScenario object.
scenario = drivingScenario;

% Add all road segments
roadCenters = [0 160 0;
    0 -160 0];
marking = [laneMarking('Solid')
    laneMarking('Dashed')
    laneMarking('Solid')];
laneSpecification = lanespec(2, 'Width', 3.5, 'Marking', marking);
road(scenario, roadCenters, 'Lanes', laneSpecification, 'Name', 'Road');

% Add the non-ego actors
globalvehicletarget = vehicle(scenario, ...
    'ClassID', 1, ...
    'Position', [1.75 nonEgoInitialPosition 0], ...
    'FrontOverhang', 0.9, ...
    'Wheelbase', 2.8, ...
    'Mesh', driving.scenario.carMesh, ...
    'Name', 'Global Vehicle Target ');
waypoints = [1.75 nonEgoInitialPosition 0;
    1.75 100 0];
speed = nonEgoSpeed;
trajectory(globalvehicletarget, waypoints, speed);
```



nonEgoSpeed = 5m/s



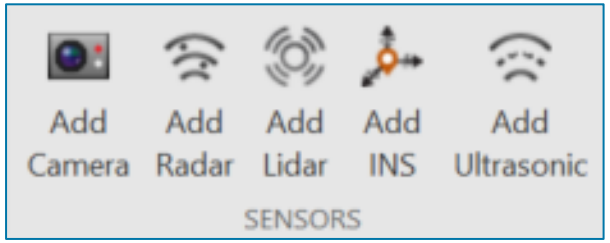
nonEgoSpeed = 10m/s



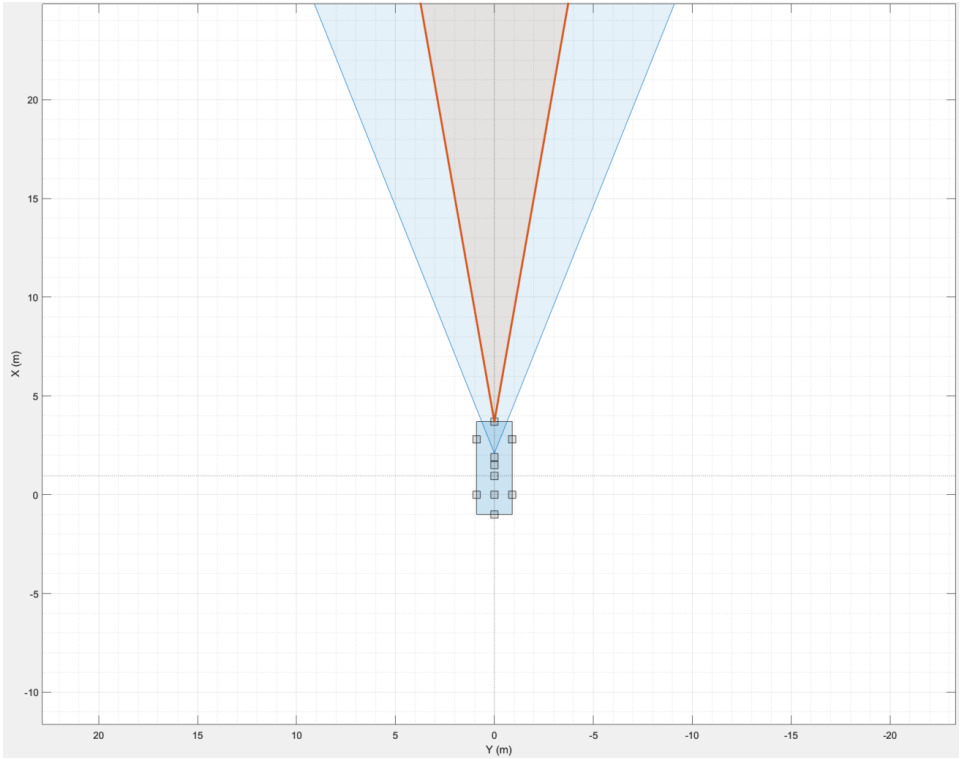
nonEgoSpeed = 15m/s

Add sensors to our ego vehicle

Load scene → Scenario variations → **Add sensors** → Simulink Export → Closed loop simulation



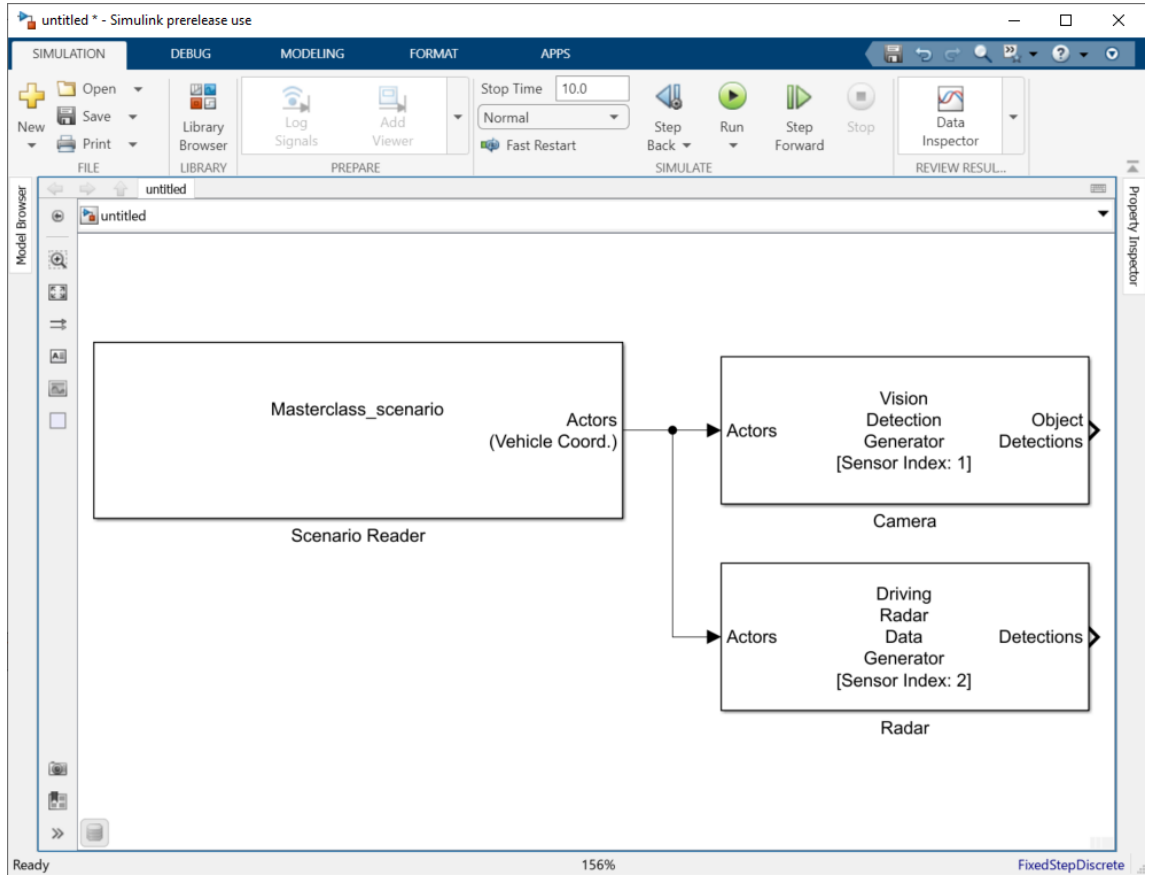
Design sensor architecture



Export scenario and sensors to Simulink

Load scene → Scenario variations → Add sensors → Simulink Export → Closed loop simulation

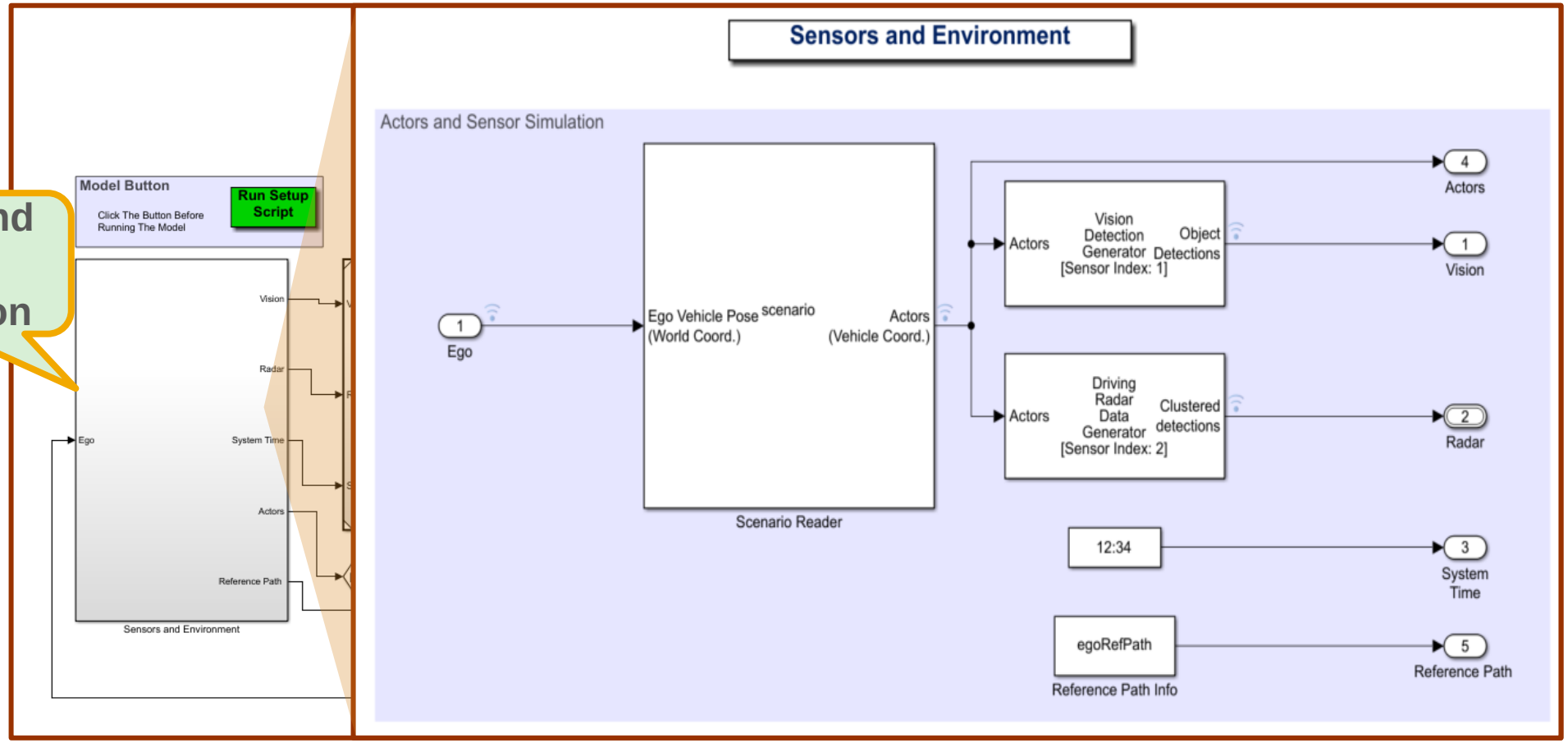
Automatic generation



Run closed-loop simulation

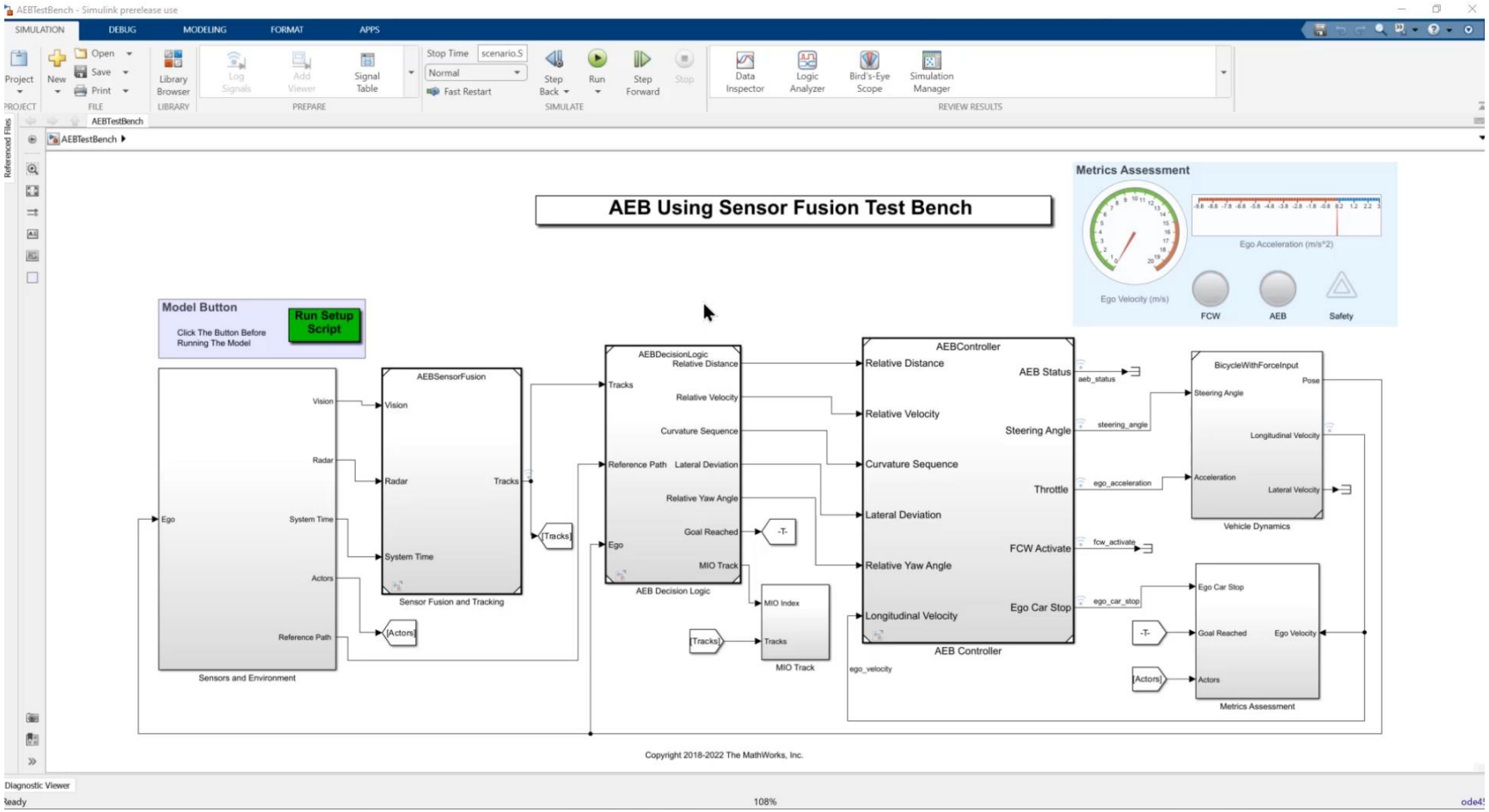
Load scene → Scenario variations → Add sensors → Simulink Export → Closed loop simulation

Actors and Sensor Simulation



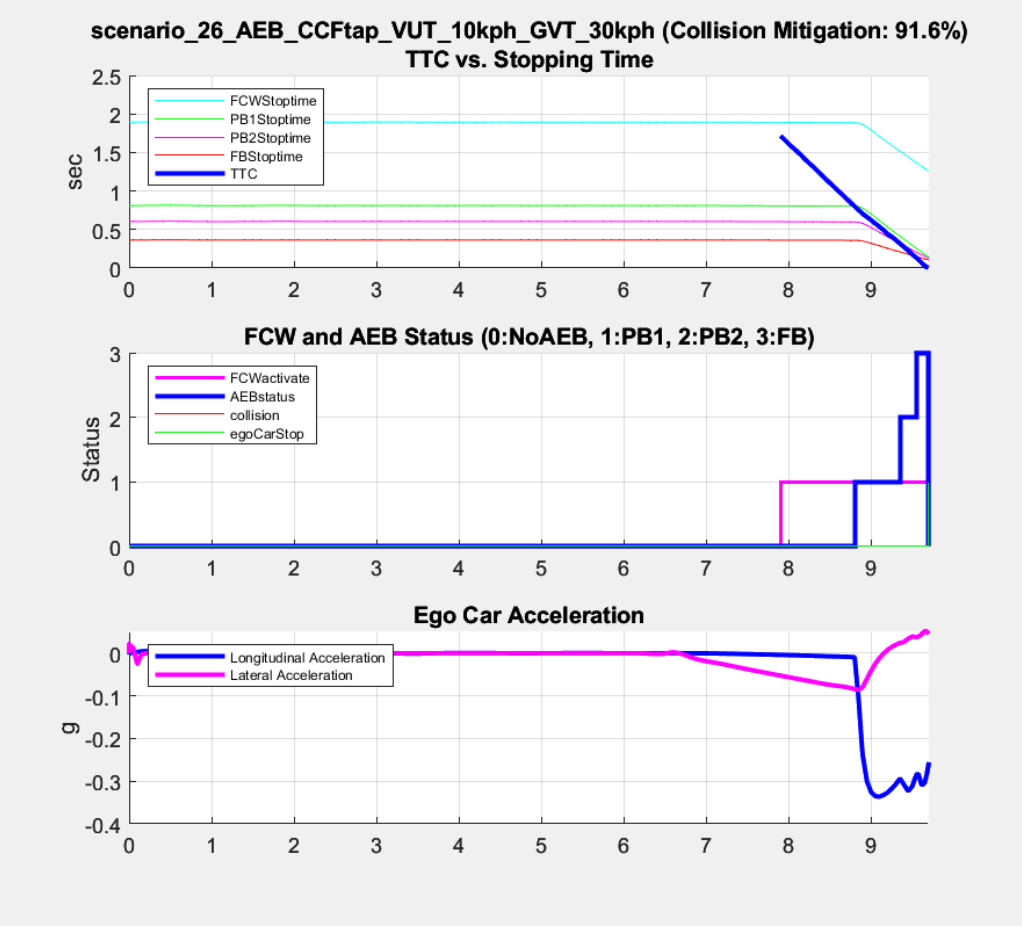
Run closed-loop simulation

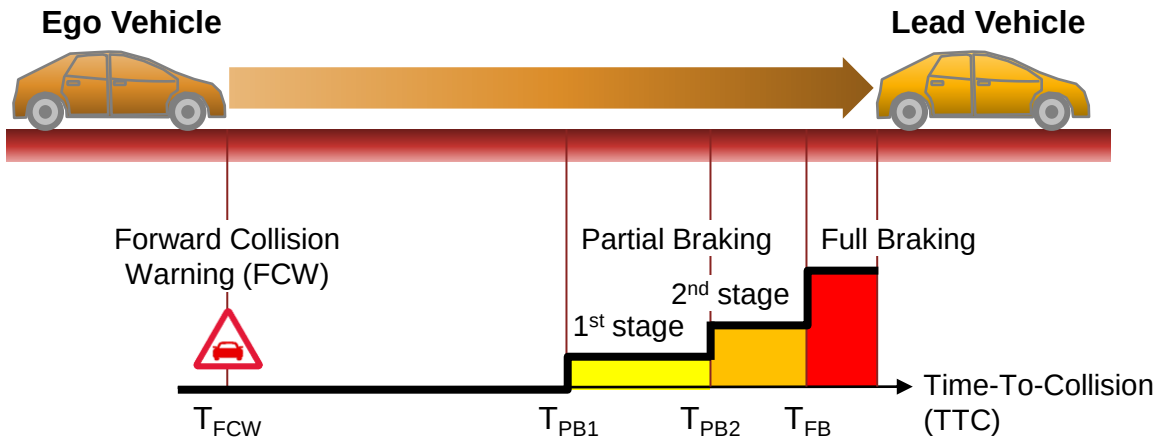
Load scene → Scenario variations → Add sensors → Simulink Export → Closed loop simulation



Results from the simulation

Load scene → Scenario variations → Add sensors → Simulink Export → Closed loop simulation





Agenda

1. Introduction
2. Testing ADAS Systems
3. Cuboid scene and trajectory-driven scenario design
4. Photorealistic scene and logic-driven scenario design
5. Conclusion

RoadRunner Product Family

OpenDRIVE
OpenStreetMap

Geographic Information System Files

Custom Assets

RoadRunner Asset Library

RoadRunner Scene Builder

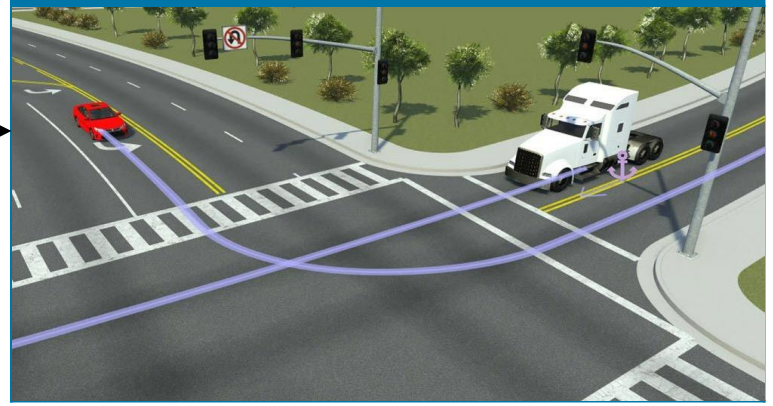
HD Maps

RoadRunner

OpenDRIVE

Third Party Simulators
- CARLA
- Unreal Engine®
- Unity®
- Others

RoadRunner Scenario



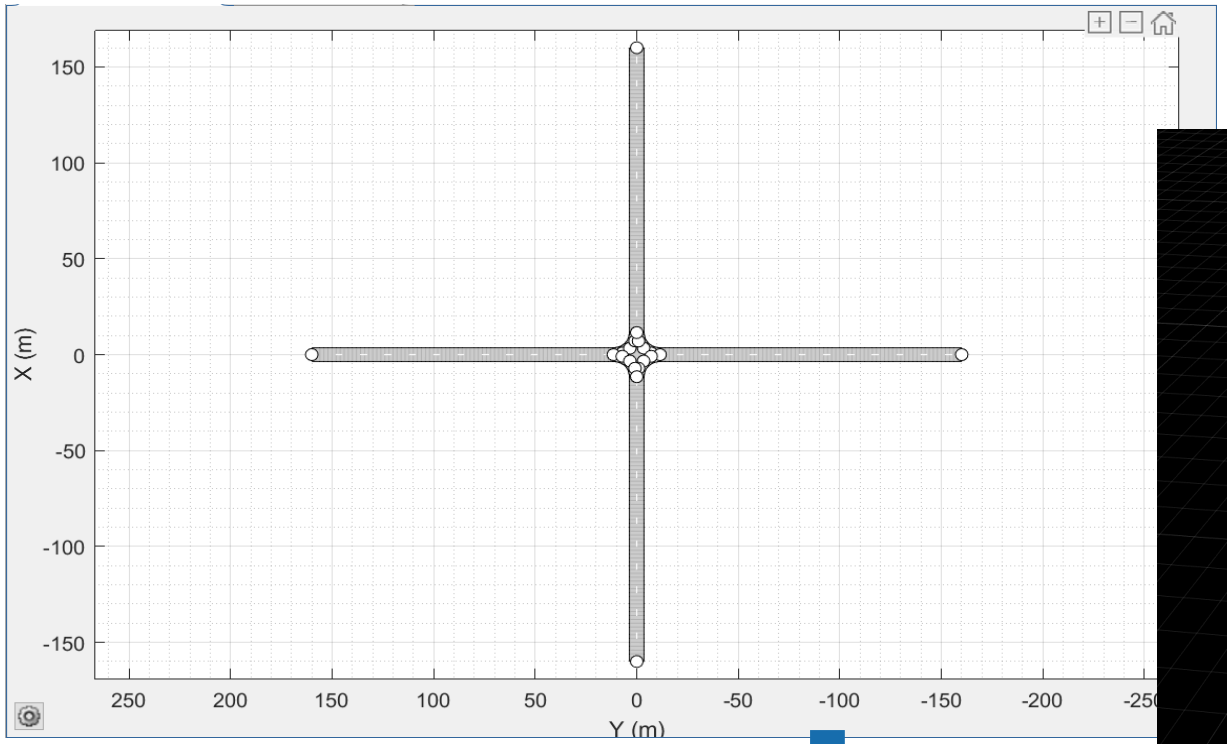
3D Scene and Scenario workflow

Using RoadRunner tool suite and the Automated Driving Toolbox

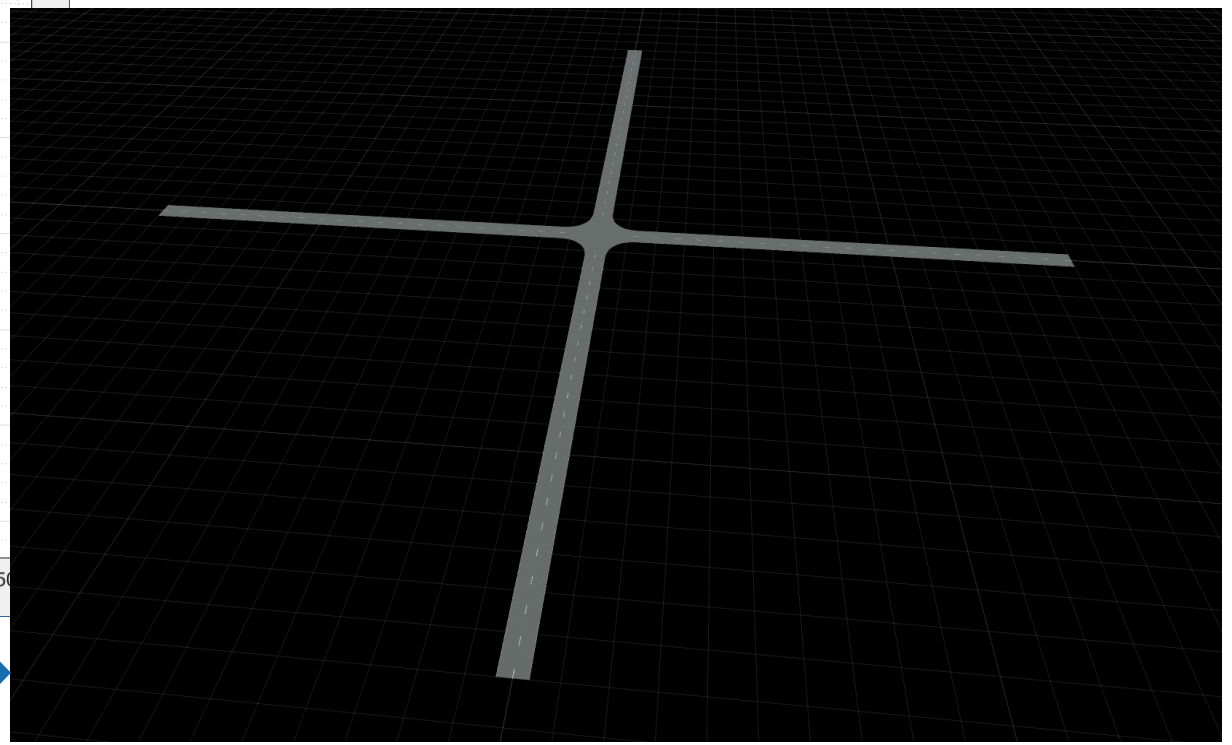
Import scene using Industry Standards

And customize it

Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation



OpenDRIVE Import



Driving Scenario Designer - AEB_CCFtap_VUT_10kph_GVT_30kph - Scenario Canvas

DESIGNER

New Open Save Import — FILE
Add Road Add Actor — SCENARIO
Add Camera Add Radar Add Lidar Add INS Add Ultrasonic — SENSORS
Display — PROPERTIES
Go to Start Step Back Run Step Forward Settings Repeat — SIMULATE
Default Layout 3D Display — VIEW
Export — EXPORT

Roads | Actors

1: Vehicle Under Test (ego vehicle)

Name: Vehicle Under Test — Set As Ego Vehicle
Class: Car
3D Display Type: Sedan

▼ Actor Properties

Length (m):	Width (m):	Height (m):
4.7	1.8	1.4

Front Overhang:	Rear Overhang:
0.9	1

Roll (°):	Pitch (°):	Yaw (°):
0	0	-90

▼ Radar Cross Section — Import

Azimuth Angles (°): [-180 180]
Elevation Angles (°): [-90 90]

Pattern (dBsm)

	-180	180
-90	10	10
90	10	10

▼ Trajectory

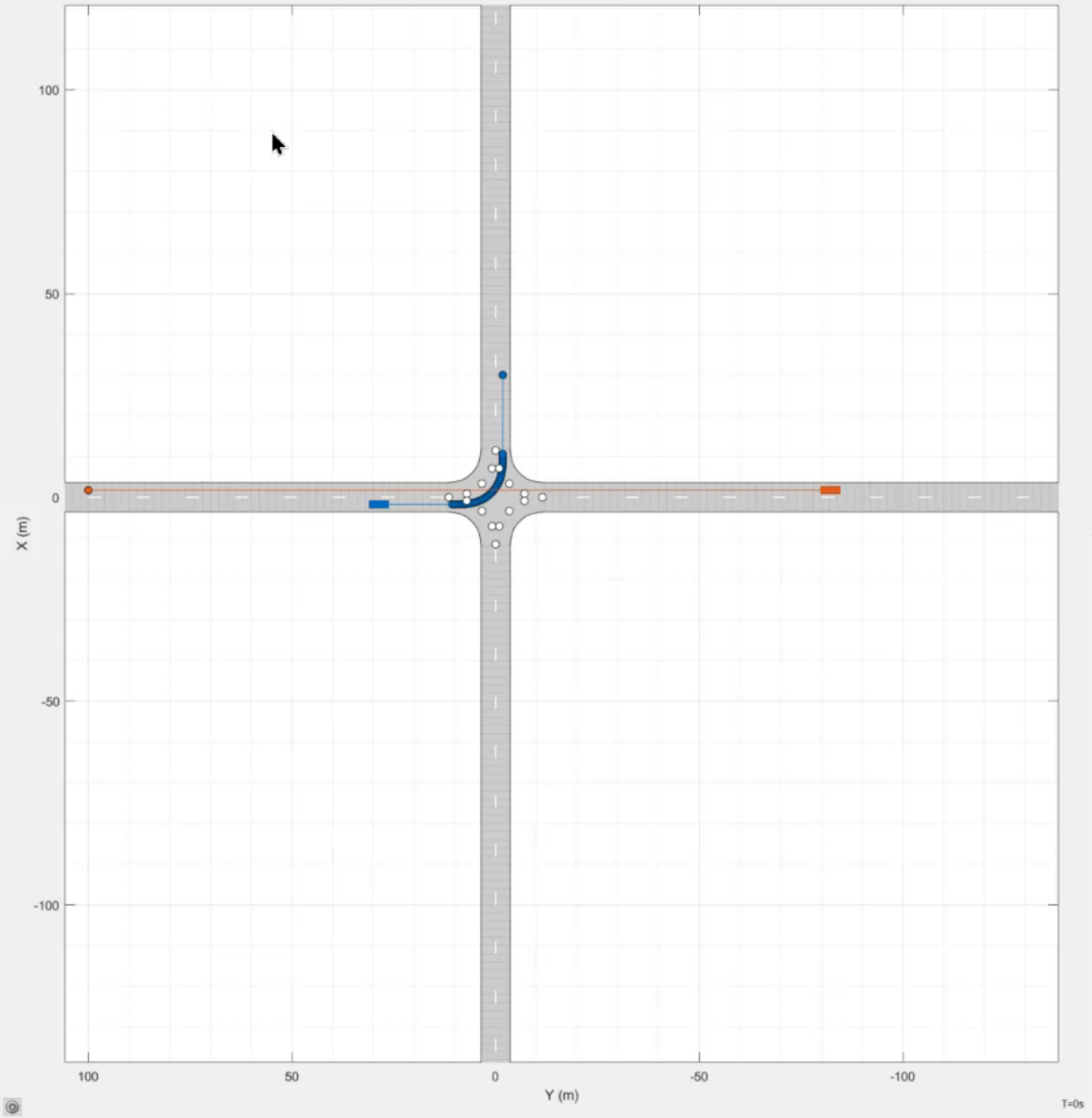
Constant Speed (m/s): 2.77778

Waypoints, Speeds, Wait Times, and Yaw

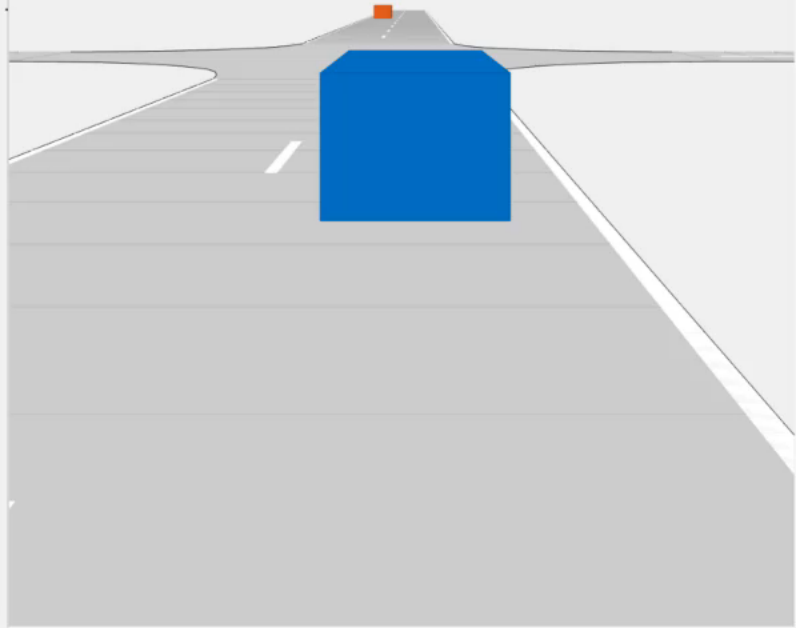
	x (m)	y (m)	z (m)	v (m/s)	wait (s)	yaw (°)
1	-1.7500	30	0	2.7778	0	-90.0038
2	-1.7500	10.6298	0	2.7778	0	-89.9924
3	-1.7488	9.9216	0	2.7778	0	-89.7285
4	-1.7412	9.2134	0	2.7778	0	-88.9596
5	-1.7211	8.5055	0	2.7778	0	-87.7010
6	-1.6824	7.7984	0	2.7778	0	-85.9487
7	-1.6190	7.0931	0	2.7778	0	-83.7037
8	-1.5251	6.3912	0	2.7778	0	-80.9660
9	-1.3947	5.6952	0	2.7778	0	-77.7347
10	-1.2224	5.0084	0	2.7778	0	-74.0130
11	-1.0029	4.3352	0	2.7778	0	-69.7894
12	-0.7325	3.6809	0	2.7778	0	-65.2905
13	-0.4114	3.0499	0	2.7778	0	-60.7794
14	-0.0417	2.4460	0	2.7778	0	-56.2715
15	0.3743	1.8731	0	2.7778	0	-51.7628
16	0.8340	1.3346	0	2.7778	0	-47.2543
17	1.3346	0.8340	0	2.7778	0	-42.7457
18	1.8731	0.3743	0	2.7778	0	-38.2372
19	2.4460	-0.0417	0	2.7778	0	-33.7285
20	3.0499	-0.4114	0	2.7778	0	-29.2206
21	3.6809	-0.7325	0	2.7778	0	-24.7095
22	4.3352	-1.0029	0	2.7778	0	-20.2106
23	5.0084	-1.2224	0	2.7778	0	-15.9870
24	5.6952	-1.3947	0	2.7778	0	-12.2653
25	6.3912	-1.5251	0	2.7778	0	-9.0340
26	7.0931	-1.6190	0	2.7778	0	-6.2963
27	7.7984	-1.6824	0	2.7778	0	-4.0513
28	8.5055	-1.7211	0	2.7778	0	-2.2990
29	9.2134	-1.7412	0	2.7778	0	-1.0404
30	9.9216	-1.7488	0	2.7778	0	-0.2715
31	10.6298	-1.7500	0	2.7778	0	-0.0076
32	30	-1.7500	0	2.7778	0	0.0038

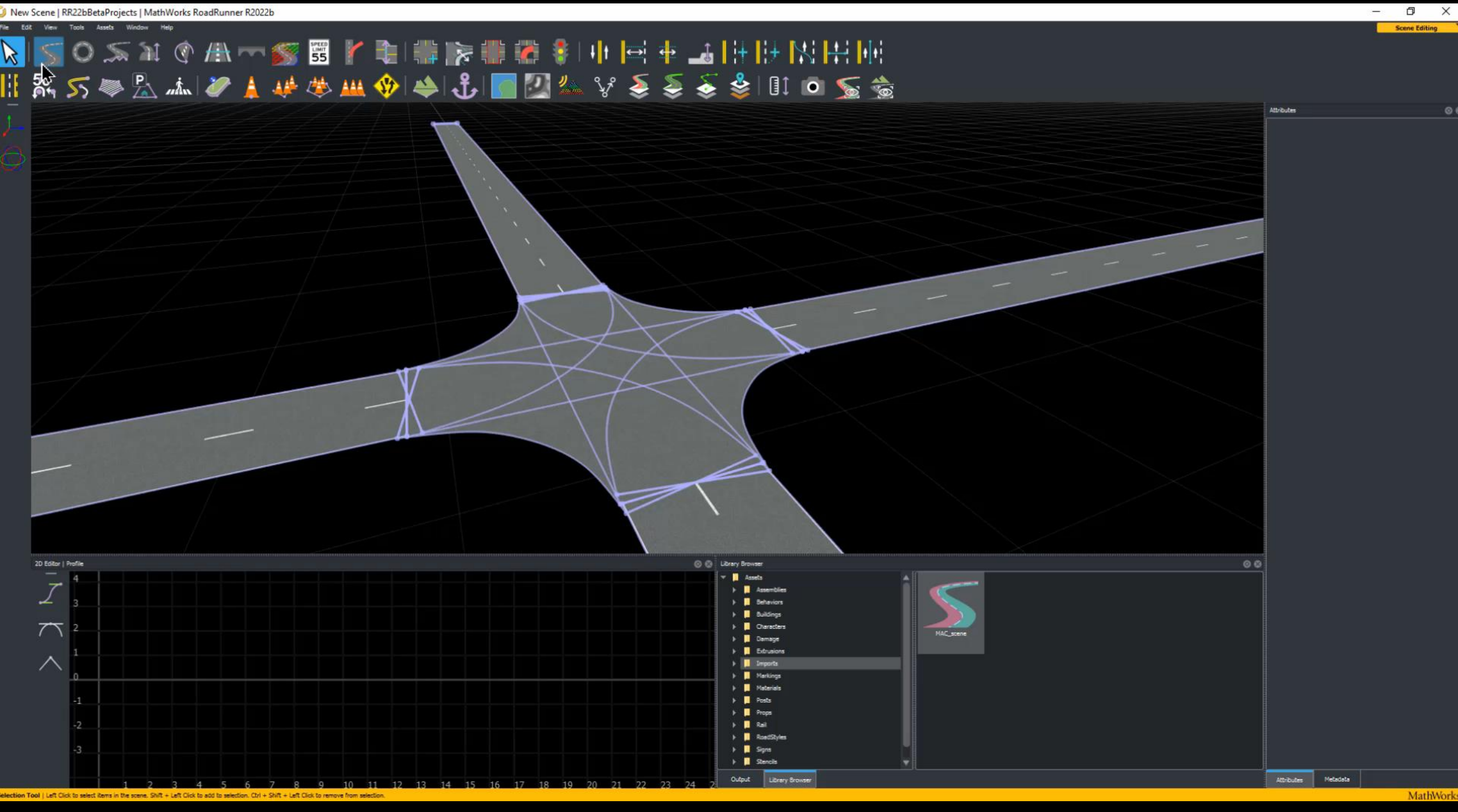
Use smooth, jerk-limited trajectory
Jerk (m/s³):
Actor spawn and despawn
Entry Time (s):
Exit Time (s):

Scenario Canvas



Ego-Centric View

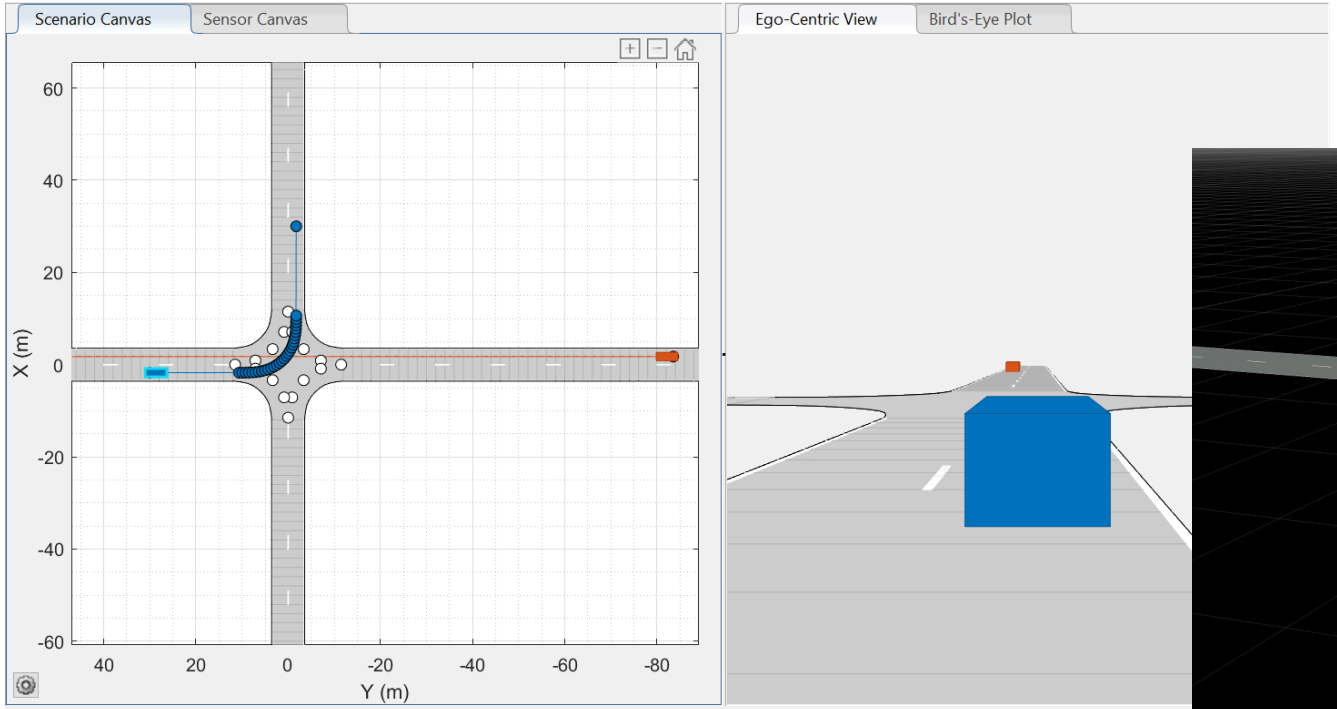



New Scene | RR22bBetaProjects | MathWorks RoadRunner R2022b
File
Edit
View
Tools
Assets
Window
Help
Scene Editing
Attributes
2D Editor | Profile
Library Browser
Assets
Assemblies
Behaviors
Buildings
Characters
Damage
Extrusions
Imports
Markings
Materials
Posts
Props
Rail
RoadStyles
Signs
Stencils
MAC_scene
Output
Library Browser
Attributes
Metadata
Selection Tool | Left Click to select items in the scene. Shift + Left Click to add to selection. Ctrl + Shift + Left Click to remove from selection.
MathWorks

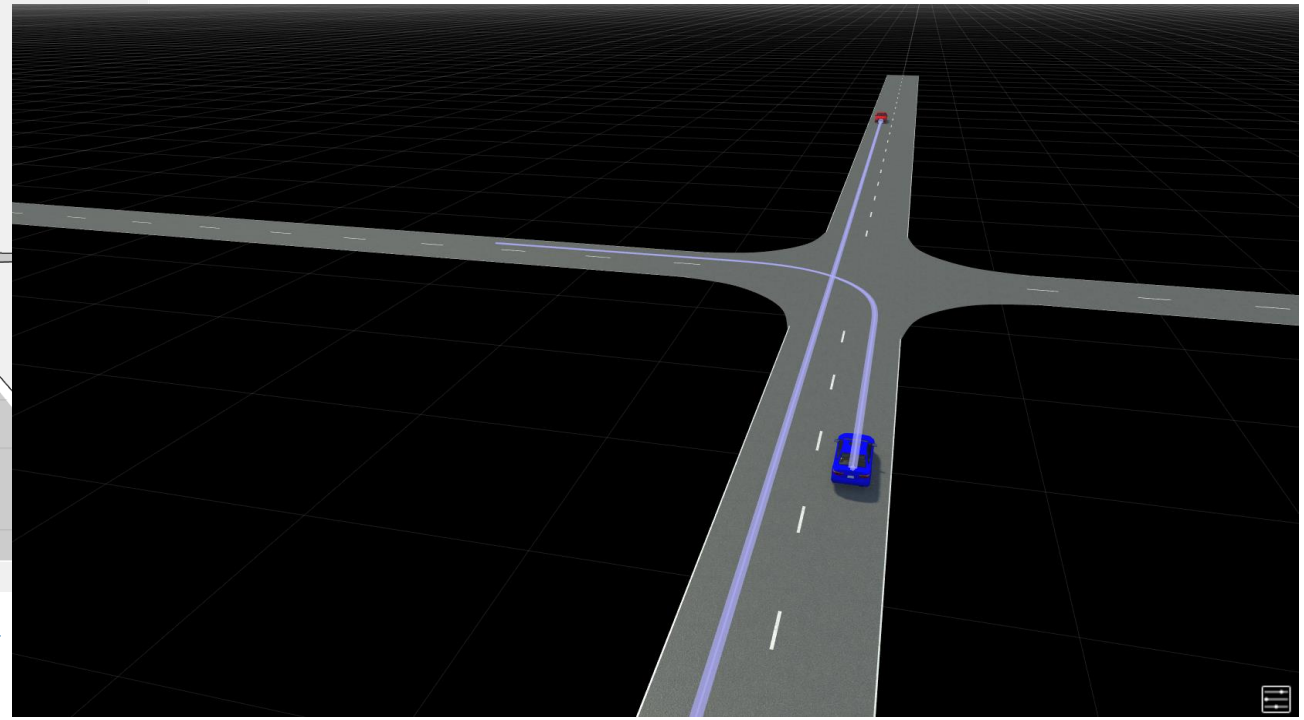
Import scenario using Industry Standards

Or re-design it using RoadRunner's logic editor

Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation



OpenSCENARIO Import



Driving Scenario Designer - AEB_CCFtap_VUT_10kph_GVT_30kph - Scenario Canvas

DESIGNER

New Open Save Import — FILE
Add Road Add Actor — SCENARIO
Add Camera Add Radar Add Lidar Add INS Add Ultrasonic — SENSORS
Display — PROPERTIES
Go to Start Step Back Run Step Forward Settings Repeat — SIMULATE
Default Layout 3D Display — VIEW
Export — EXPORT

Roads | Actors

1: Vehicle Under Test (ego vehicle)

Name: Vehicle Under Test — Set As Ego Vehicle
Class: Car
3D Display Type: Sedan

▼ Actor Properties

Length (m):	Width (m):	Height (m):
4.7	1.8	1.4

Front Overhang:	Rear Overhang:
0.9	1

Roll (°):	Pitch (°):	Yaw (°):
0	0	-90

▼ Radar Cross Section — Import

Azimuth Angles (°): [-180 180]
Elevation Angles (°): [-90 90]

Pattern (dBsm)

	-180	180
-90	10	10
90	10	10

▼ Trajectory

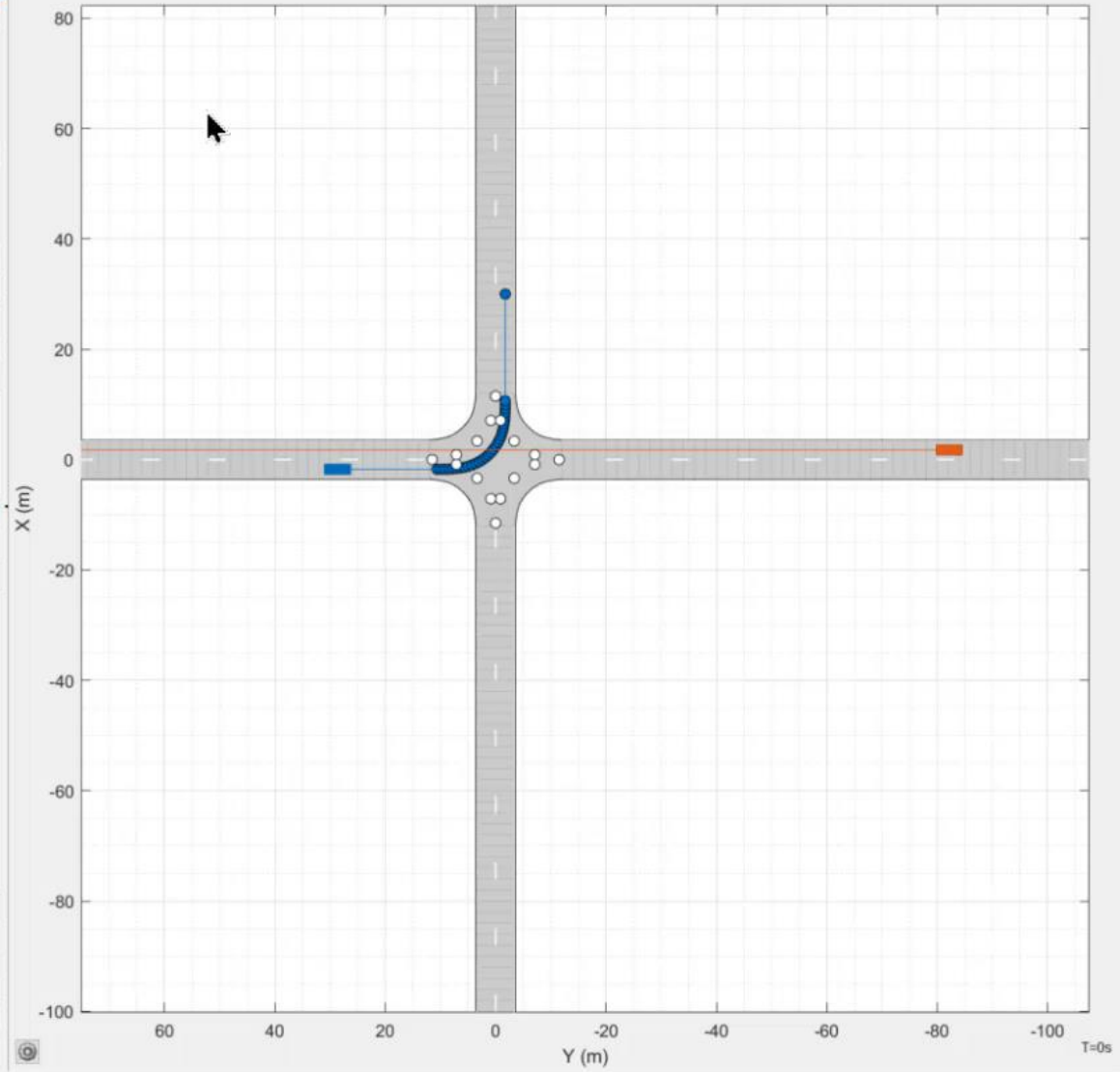
Constant Speed (m/s): 2.77778

Waypoints, Speeds, Wait Times, and Yaw

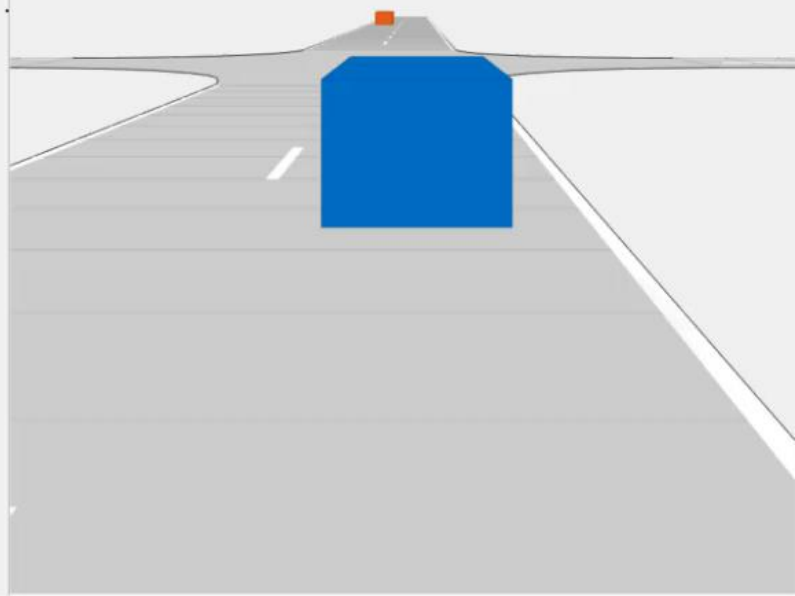
	x (m)	y (m)	z (m)	v (m/s)	wait (s)	yaw (°)
1	-1.7500	30	0	2.7778	0	-90.0038
2	-1.7500	10.6298	0	2.7778	0	-89.9924
3	-1.7488	9.9216	0	2.7778	0	-89.7285
4	-1.7412	9.2134	0	2.7778	0	-88.9596
5	-1.7211	8.5055	0	2.7778	0	-87.7010
6	-1.6824	7.7984	0	2.7778	0	-85.9487
7	-1.6190	7.0931	0	2.7778	0	-83.7037
8	-1.5251	6.3912	0	2.7778	0	-80.9660
9	-1.3947	5.6952	0	2.7778	0	-77.7347
10	-1.2224	5.0084	0	2.7778	0	-74.0130
11	-1.0029	4.3352	0	2.7778	0	-69.7894
12	-0.7325	3.6809	0	2.7778	0	-65.2905

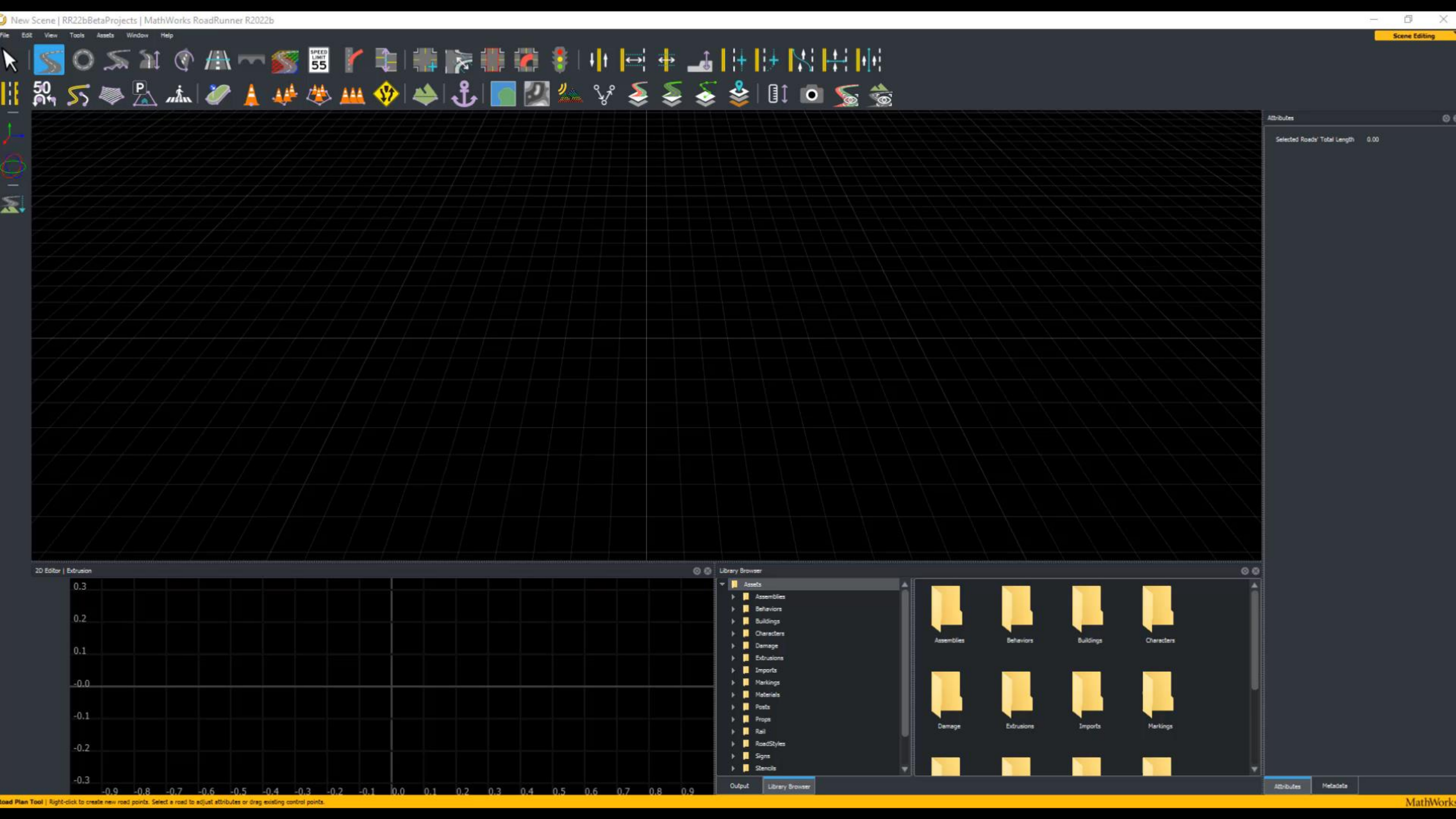
☐ Use smooth, jerk-limited trajectory
Jerk (m/s³):
☐ Actor spawn and despawn
Entry Time (s):
Exit Time (s):

Scenario Canvas



Ego-Centric View




New Scene | RR22bBetaProjects | MathWorks RoadRunner R2022b
File
Edit
View
Tools
Assets
Window
Help
Scene Editing
Attributes
Selected Roads' Total Length
0.00
2D Editor | Extrusion
0.3
0.2
0.1
-0.0
-0.1
-0.2
-0.3
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Library Browser
Assets
Assemblies
Behaviors
Buildings
Characters
Damage
Extrusions
Imports
Markings
Materials
Posts
Props
Rail
RoadStyles
Signs
Stencils
Output
Library Browser
Attributes
Metadata
oad Plan Tool | Right-click to create new road points. Select a road to adjust attributes or drag existing control points.
MathWorks

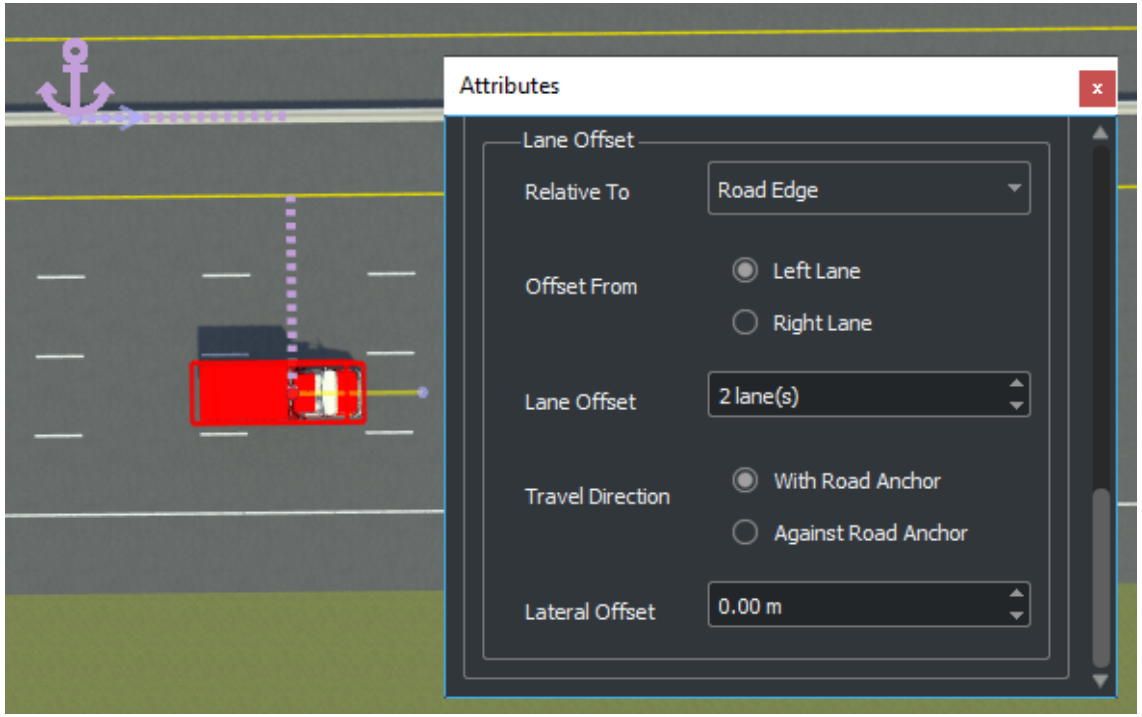
Import scenario using Industry Standards

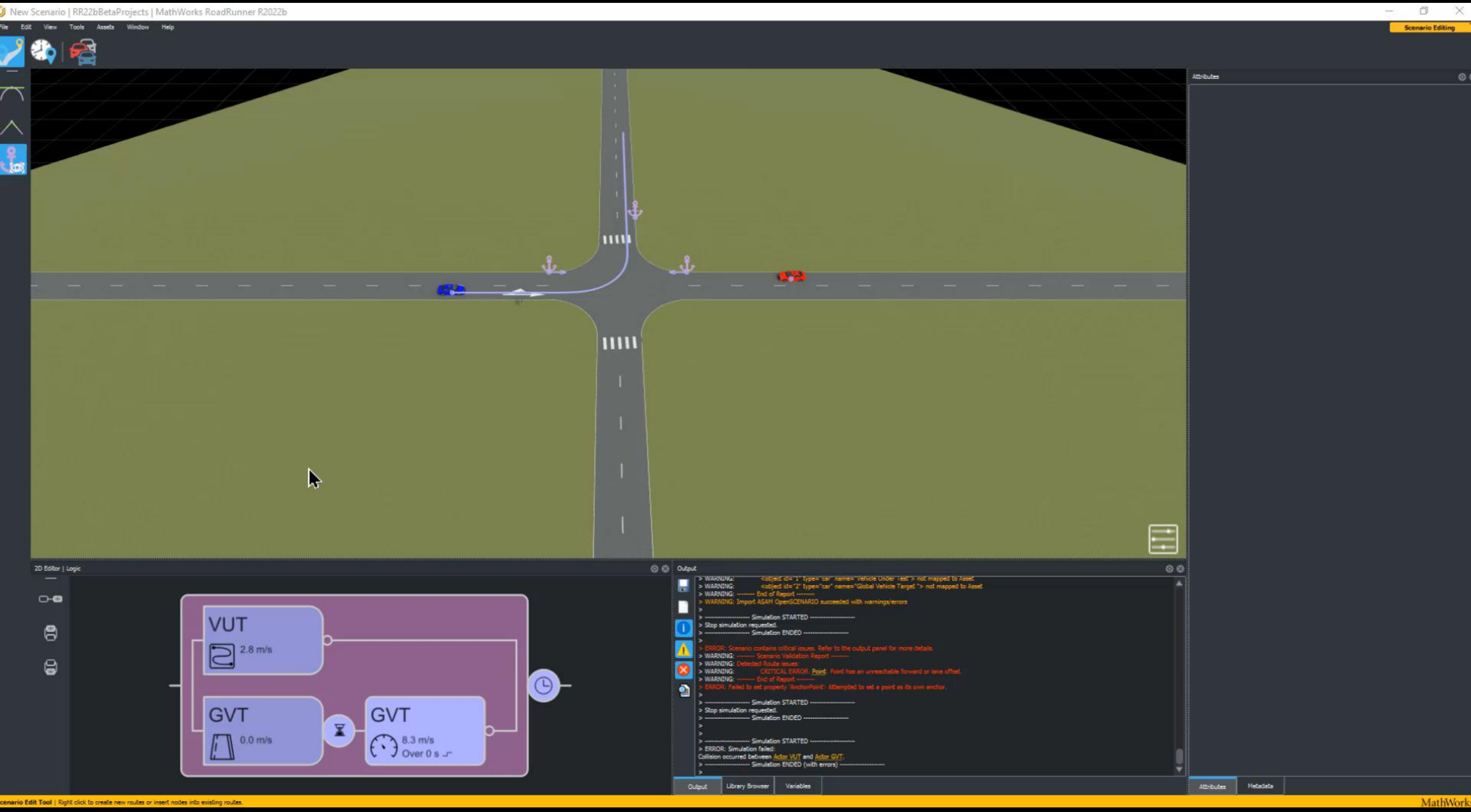
Or re-design it using RoadRunner's logic editor

Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation

Anchoring system in RoadRunner Scenario

- Position actor and waypoints relatively to an anchors
- The anchors allow reusability of the scenario within multiple scenes



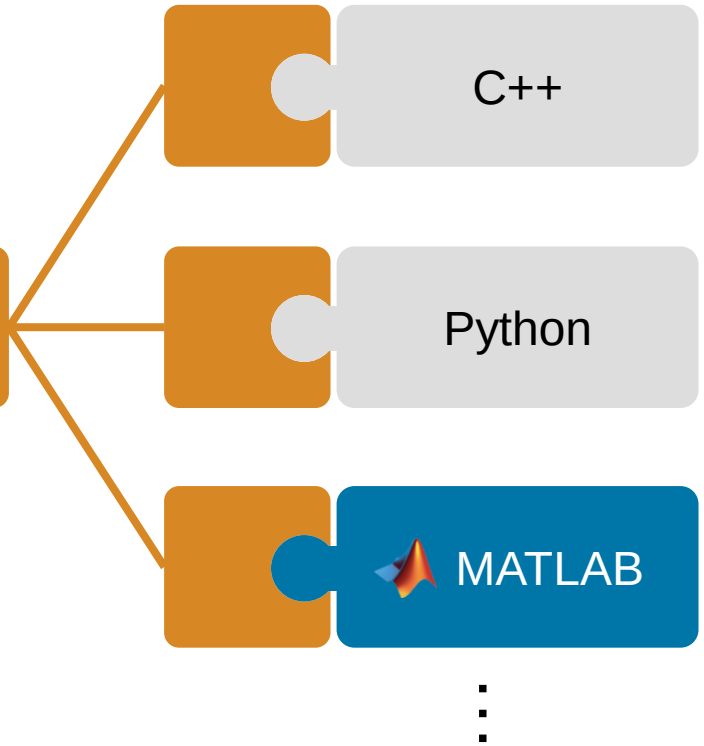

New Scenario | RR22bBetaProjects | MathWorks RoadRunner R2022b
File
Edit
View
Tools
Assets
Window
Help
Scenario Editing
Attributes
2D Editor | Logic
VUT
2.8 m/s
GVT
0.0 m/s
GVT
8.3 m/s
Over 0 s
Output
> WARNING: <object id="1" type="car" name="Vehicle Under Test" > not mapped to Asset
> WARNING: <object id="2" type="car" name="Global Vehicle Target "> not mapped to Asset
> WARNING: -------- End of Report --------
> WARNING: Import ASAM OpenSCENARIO succeeded with warnings/errors
> ---------- Simulation STARTED ----------
> Stop simulation requested.
> ---------- Simulation ENDED ----------
> ERROR: Scenario contains critical issues. Refer to the output panel for more details.
> WARNING: -------- Scenario Validation Report --------
> WARNING: Detected Route issues:
> WARNING: CRITICAL ERROR: Point: Point has an unreachable forward or lane offset.
> WARNING: -------- End of Report --------
> ERROR: Failed to set property 'AnchorPoint': Attempted to set a point as its own anchor.
> ---------- Simulation STARTED ----------
> Stop simulation requested.
> ---------- Simulation ENDED ----------
> ---------- Simulation STARTED ----------
> ERROR: Simulation failed:
Collision occurred between Actor VUT and Actor GVT.
> ---------- Simulation ENDED (with errors) ----------
Output
Library Browser
Variables
Attributes
Metadata
cenario Edit Tool | Right click to create new routes or insert nodes into existing routes.
MathWorks

Interact with RoadRunner Scenario using its gRPC API

In MATLAB

Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation

RoadRunner
RoadRunner Scenario



RoadRunner API

- Starting/Closing RoadRunner
- Change scenario variables
- Running scenario simulations
- Data acquisition such as trajectory and speed
- Export to OpenSCENARIO

MATLAB R2022b - prerelease use

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Compare Print Go To Find Bookmark Refactor Profiler Analyze Run Section Section Break Run and Advance Run to End Run Step Stop

FILE NAVIGATE CODE ANALYZE SECTION RUN

C: ▸ Users ▸ mfrancoi ▸ OneDrive - MathWorks ▸ Documents ▸ AE Projects ▸ MAC - RRS - 10-2022 ▸ AEB_originalFromSri ▸ AEBWithRoadRunner ▸

Current Folder

Name

- Folder
  - AEBWithRRScenario
  - slprj
- Script
  - AutonomousEmergencyBrakingWithRoadRunnerScenarioExamp...
  - ScenarioVariation.m
  - SetUpRRSCosimulation.m
- MAT-file
  - dsd_testMAC.mat
- PNG File
  - AEB_Behavior.png
  - AEB_Simulation.png
  - AutonomousEmergencyBrakingInterface.png
  - JunctionWithConnectingLanes.png
  - scenario_01_CarToCar_FrontTAP.png
  - Scenario_Variables.png
  - VUT_PathParams.png
- Simulink Cache
  - AEBController.slxc
  - AEBDecisionLogic.slxc

AEBWithRRScenario (Folder)

Workspace

Name	Value
ans	'off'
modelName	"AEBWithRRTestBench"
rrApp	*1x1 roadrunner*
rrAppPath	"C:\Program Files\RoadRunn
rrProjectPath	"C:\Users\mfrancoi\OneDrive
rrSim	*1x1 ScenarioSimulation*
s	*1x1 SettingsGroup*
scenarioFileName	"scenario_01_CarToCar_Front

Editor - SetUpRRSCosimulation.m

Project - AutonomousEmergencyBraking

SetUpRRSCosimulation.m | ScenarioVariation.m

```
1   %%
2   rrAppPath = "C:\Program Files\RoadRunner R2022b\bin\win64";
3   %%
4   % Specify the path to your RoadRunner project. This code shows a sample
5   % project folder on Windows.
6   %%
7   rrProjectPath = "C:\Users\mfrancoi\OneDrive - MathWorks\Documents\RoadRunner Projects\RR22bBetaProjects";
8   %%
9   % To update the path for the RoadRunner installation folder, get the root
10  % object within the settings hierarchical tree. For more information, see
11  % <docid:matlab_ref#mw_0a889445-8d8f-4c7e-be0a-b293c755302f SettingsGroup>.
12  %%
13  s = settings;
14  s.roadrunner.application.InstallationFolder.TemporaryValue = rrAppPath;
15  %%
16  % Open RoadRunner using the specified path to your project.
17  %%
18  rrApp = roadrunner(rrProjectPath);
19  %%
20  % To reduce Command Window output, turn off model predictive controller
21  % (MPC) update messages.
22  %%
23  mpcverbosity("off");
24  %% Open Scene
25  openScene(rrApp,"DownTownIntersection_MAC.rrscene")
26  %% Open Scenario
27  openScenario(rrApp,"scenario_01_CarToCar_FrontTAP_Masterclass.rrscenario",...
28      'KeepCurrentScene', true);
29  %%
30  % Connect to the RoadRunner Scenario server for cosimulation using the
```

Command Window

fx >>

Zoom: 110% UTF-8 CRLF script Ln 2 Col 60

Leverage co-simulation to control vehicles' behavior

Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation

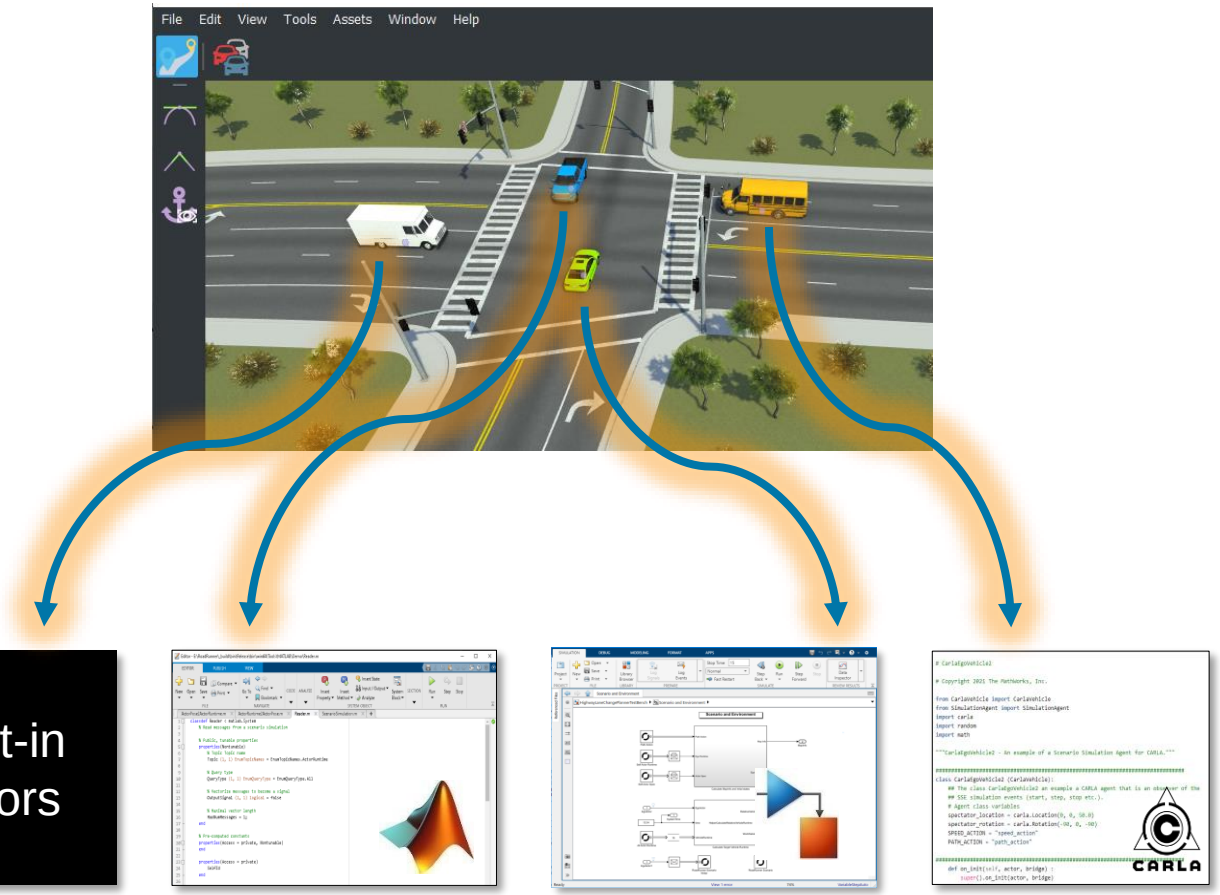
RoadRunner Scenario connects with actors in MATLAB, Simulink, and CARLA

Actors can read scenario states

- Action commands (path, speed, lane change, lateral offset)
- Pose and velocity of all actors in the scenario
- Dimensions of all actors
- Map lanes and lane boundaries

Actors write scenario states

- Their pose and velocity for each scenario simulation step

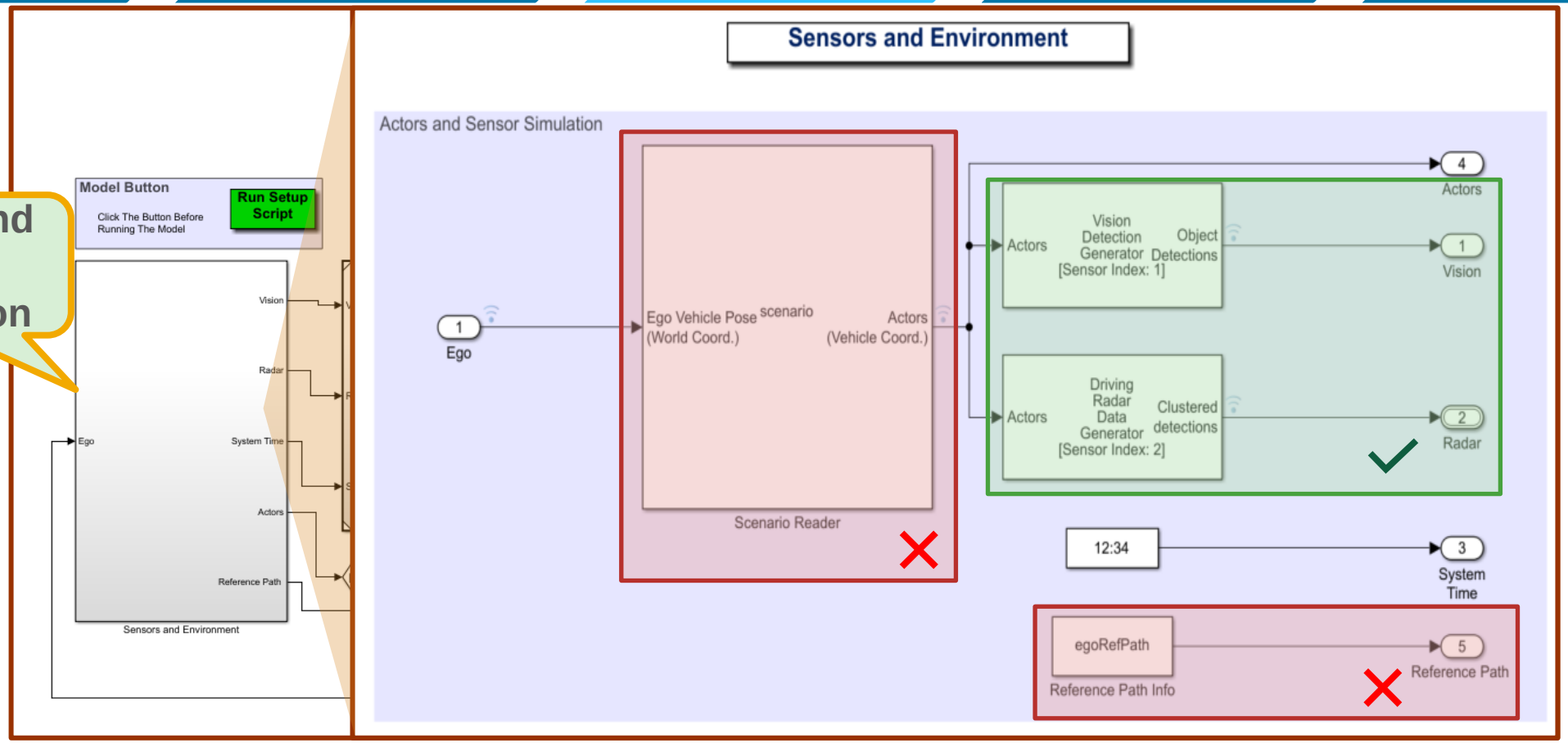


Leverage co-simulation to control vehicles' behavior

In Simulink

Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation

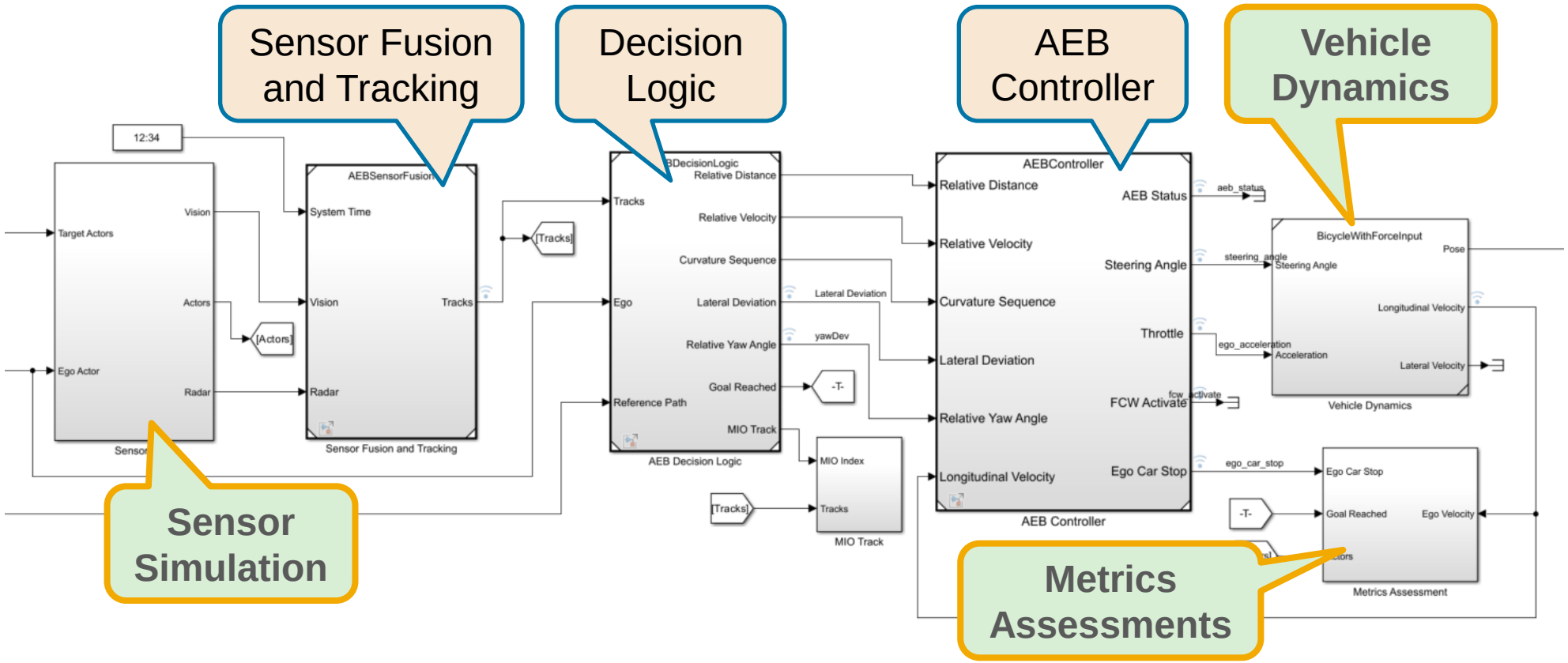
Actors and Sensor Simulation



Leverage co-simulation to control vehicles' behavior

In Simulink

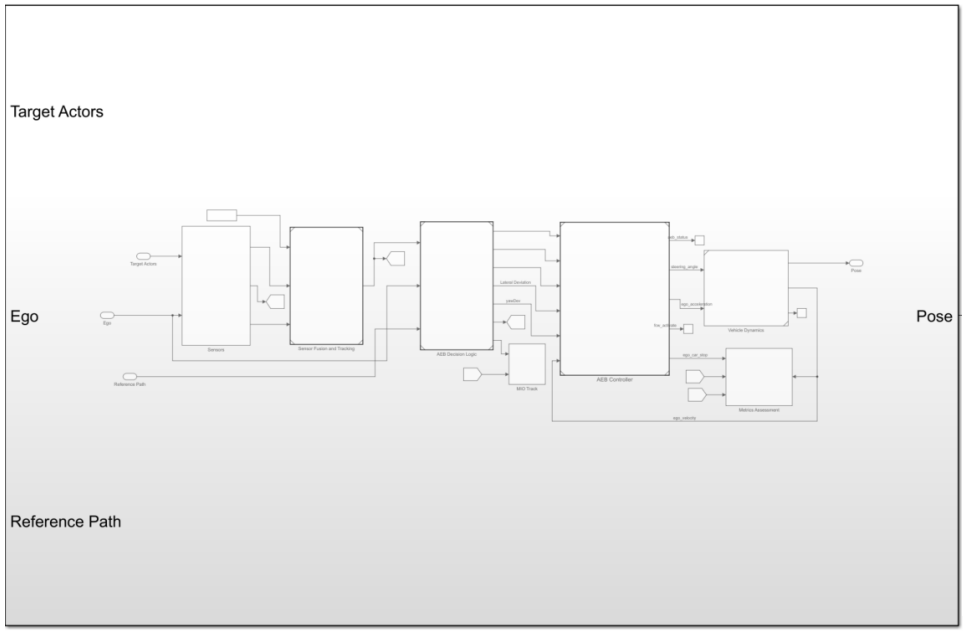
Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation



Leverage co-simulation to control vehicles' behavior

In Simulink

Scene & Scenario design → Connect MATLAB to RoadRunner → **Define behavior** → Run Simulation → Scenario variation

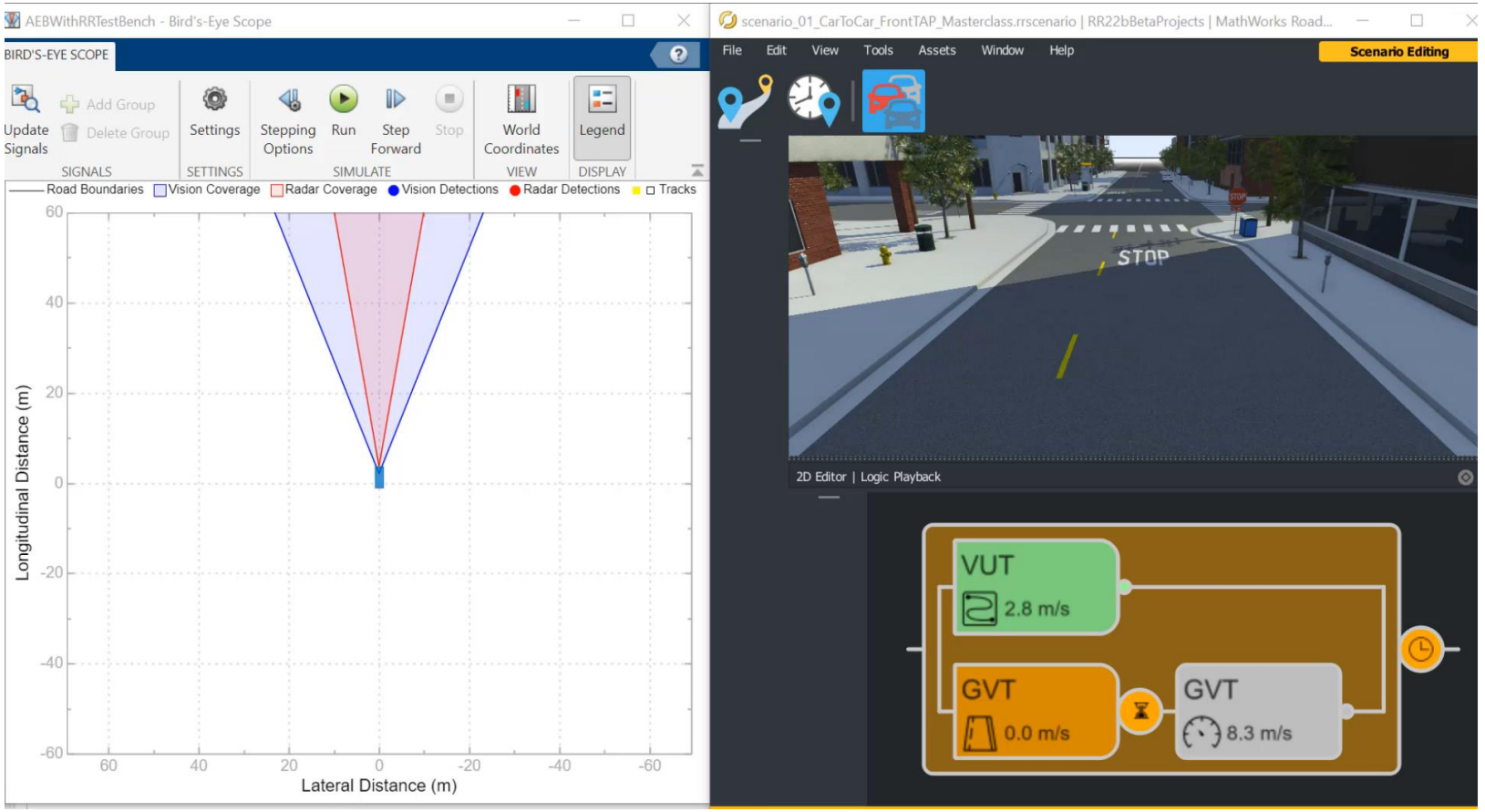


AEB Sysyem


scenario_01_CarToCar_FrontTAP_Masterclass.rrscenario | RR22bBetaProjects | MathWorks RoadRunner R2022b
File Edit View Tools Assets Window Help
Scenario Editing
Attributes
2D Editor | Logic
2.8 m/s
GVT
GVT
Library Browser
Assets
Assemblies
Behaviors
Buildings
Characters
Damage
Extrusions
Imports
Markings
Materials
Posts
ADT Vehicles
Vehicle Textures
Output
Library Browser
Variables
Attributes
Metadata
Scenario Edit Tool | Right click to create new routes or insert nodes into existing routes.
MathWorks

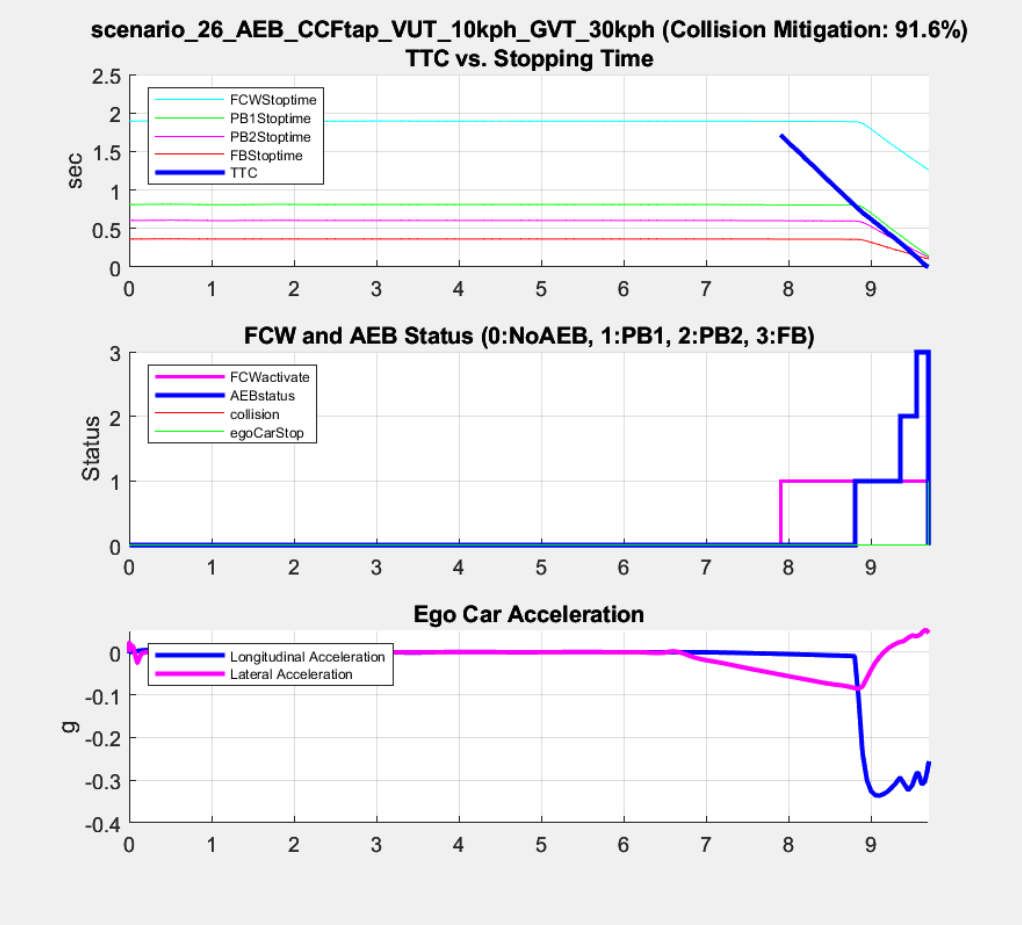
Run closed-loop simulation

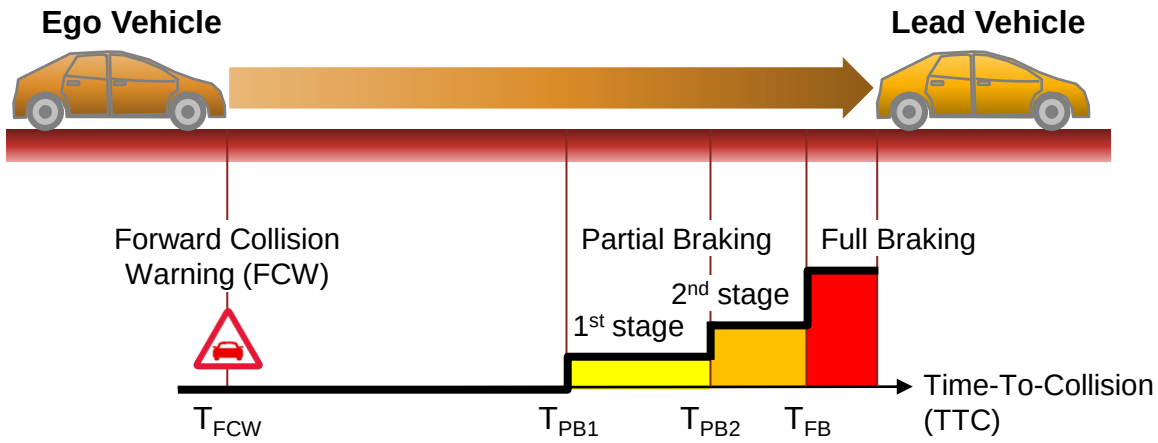
Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation



Run closed-loop simulation

Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → **Run Simulation** → Scenario variation





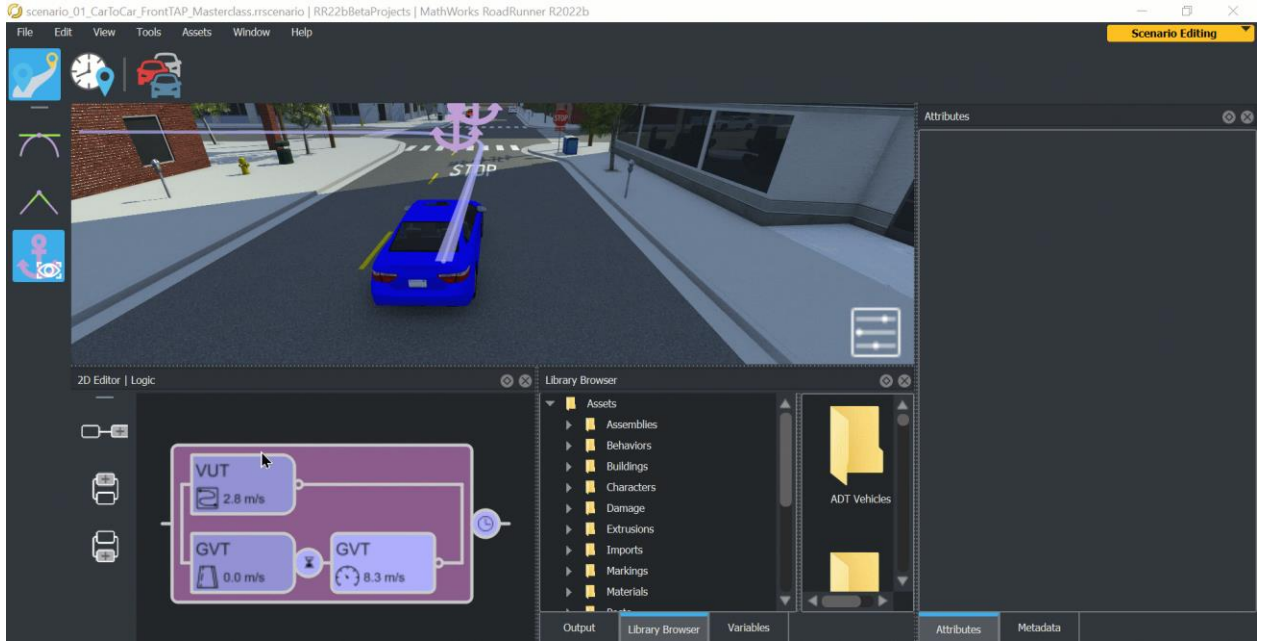
Generate scenario variations programmatically

Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation

The **Euro NCAP** standard recommends testing with scenarios by varying **VUT and GVT speeds**.

Test speed	Part 1 (clothoid)			Part 2 (constant radius)		
	Start Radius R1 [m]	End Radius R2 [m]	Angle α [deg]	Start Radius R2 [m]	End Radius R2 [m]	Angle β [deg]
10 km/h	1500	9.00	20.62	9.00	9.00	48.76
15 km/h	1500	11.75	20.93	11.75	11.75	48.14
20 km/h	1500	14.75	21.79	14.75	14.75	46.42

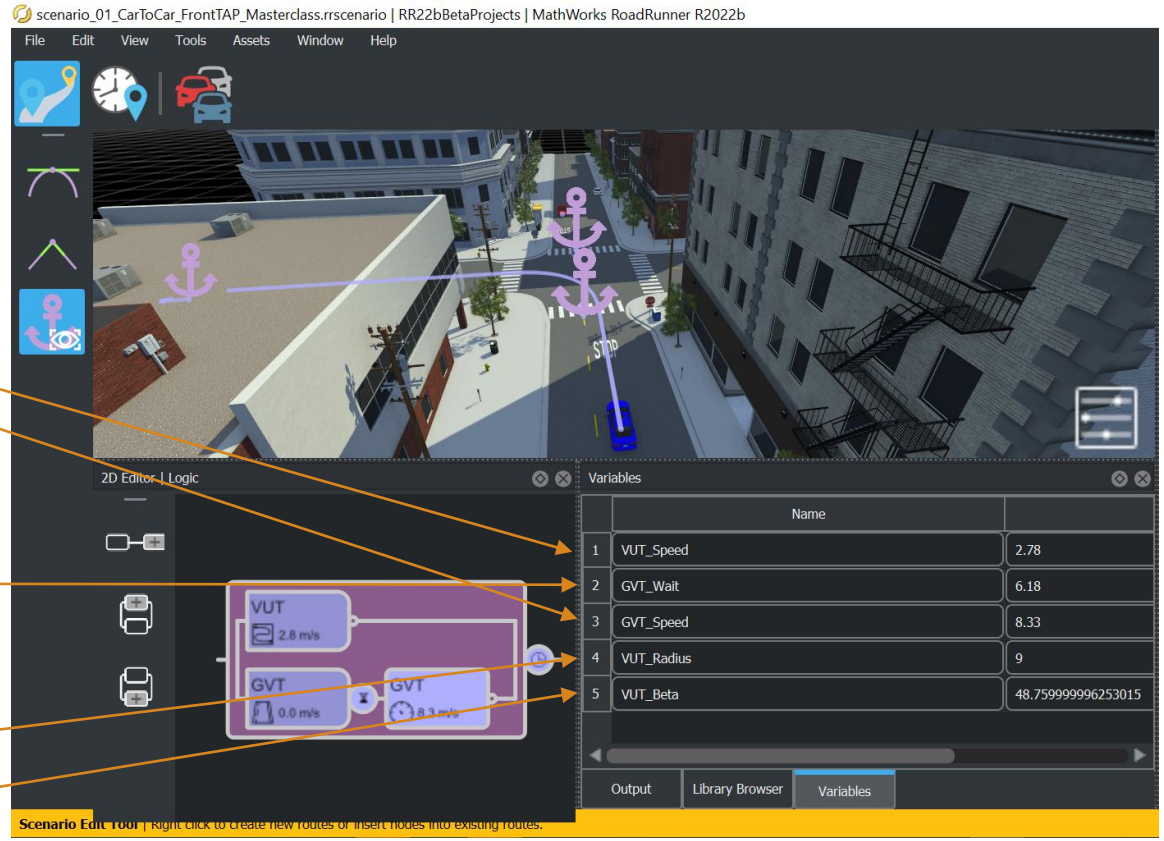
Create a RoadRunner Scenario variable:



Generate scenario variations programmatically

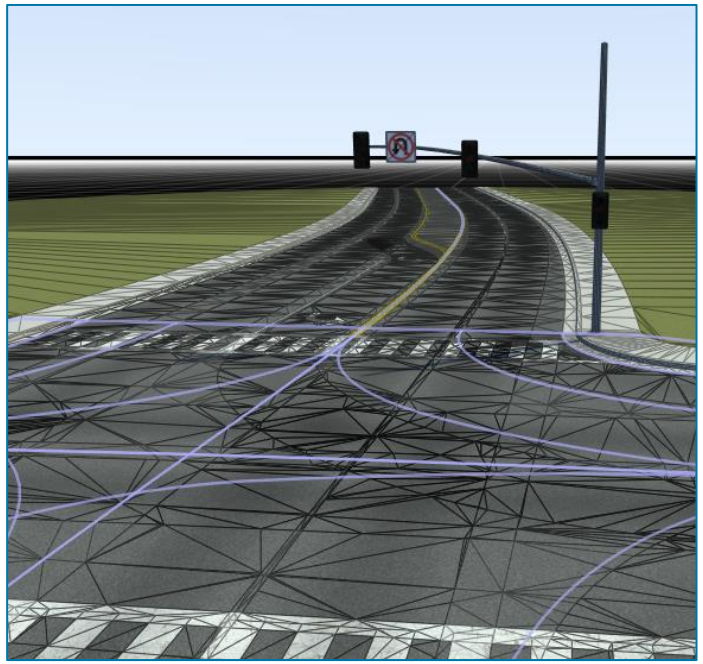
Scene & Scenario design → Connect MATLAB to RoadRunner → Define behavior → Run Simulation → Scenario variation

```
%%
load("CCFtapVariationData.mat")
disp(CCFTapVariationData)
%%
variantID = 9;
vutSpeed = table2array(CCFTapVariationData(variantID,1));
gvtSpeed = table2array(CCFTapVariationData(variantID,2));

vutSpeed  = vutSpeed * 0.2778; % convert VUT speed from km/hr to m/s
gvtSpeed  = gvtSpeed * 0.2778; % convert GVT speed from km/hr to m/s
rrApp.setScenarioVariable("VUT_Speed",num2str(vutSpeed))
rrApp.setScenarioVariable("GVT_Speed",num2str(gvtSpeed))
%%
% 2. Get the GVT wait time value from |CCFtapVariationData|.
%%
gvtWaitTime = table2array(CCFTapVariationData(variantID,6));
%%
% Update the |GVT_WaitTime| variable of RoadRunner Scenario.
%%
rrApp.setScenarioVariable("GVT_Wait",num2str(gvtWaitTime))
%%
% Based on the VUT speed, set the Euro NCAP path parameters in the
% RoadRunner Scenario.
%%
rrApp.setScenarioVariable("VUT_Radius", ...
    num2str(table2array(CCFTapVariationData(variantID,4))))
rrApp.setScenarioVariable("VUT_Beta", ...
    num2str(table2array(CCFTapVariationData(variantID,5))))
```



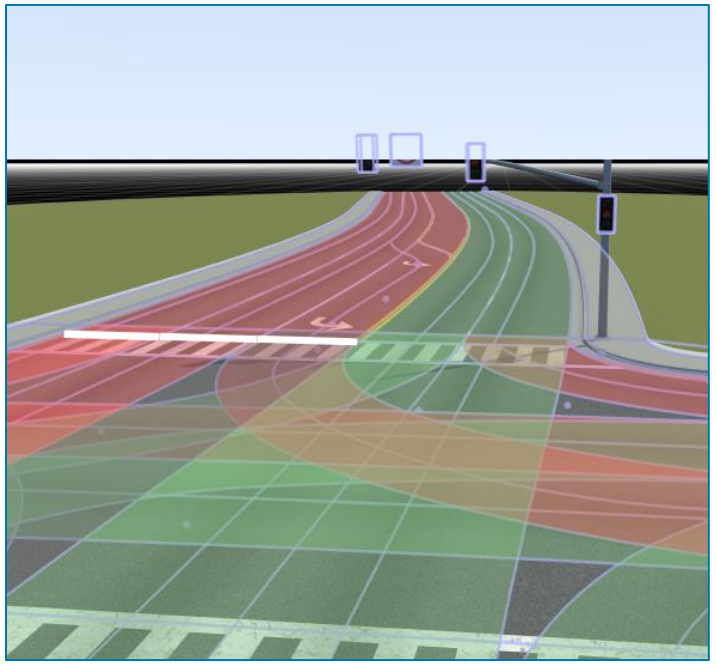
Export scenes to driving simulators and graphics engines

- Export to common file formats for use in third-party applications

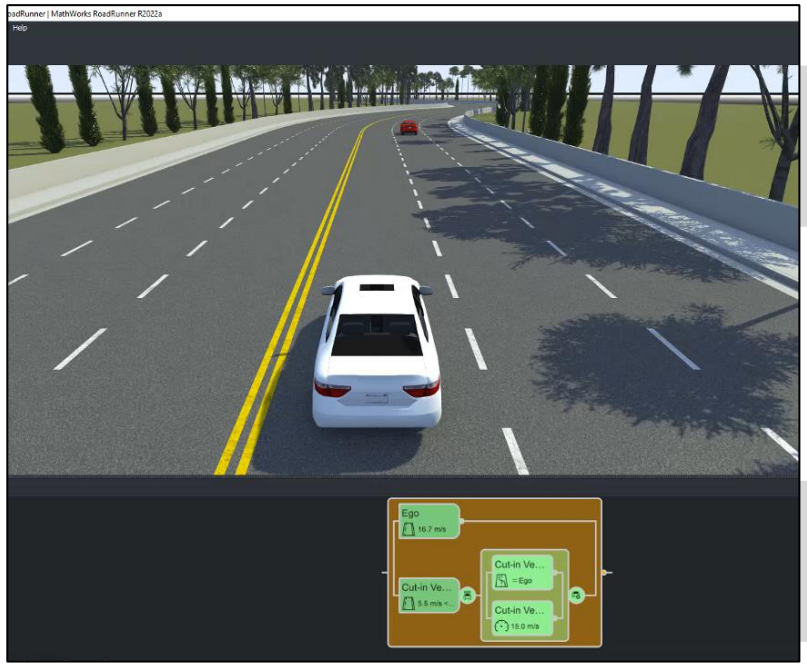


Filmbox
(meshes)

OpenDRIVE
(semantics)



Export scenarios to OpenSCENARIO



OpenSCENARIO V1.x

```
<Condition name="Start Condition of Event_Vehicle2" conditionEdge="none"
    <ByValueCondition>
        <SimulationTimeCondition value="0" rule="greaterThan"/>
    </ByValueCondition>
</Condition>
</ConditionGroup>
</StartTrigger>
</Event>
<Event name="Event_Vehicle2_2" priority="overwrite">
<Action name="Speed_Action_Vehicle2_2">
<PrivateAction>
<LongitudinalAction>
<SpeedAction>
<SpeedActionDynamics dynamicsShape="
<SpeedActionTarget>
<RelativeTargetSpeed entityRef="
</SpeedActionTarget>
</SpeedAction>
</LongitudinalAction>
</PrivateAction>
</Action>
<StartTrigger>
<ConditionGroup>
<Condition name="Start Condition of Event_Ve
<ByEntityCondition>
<TriggeringEntities triggeringEntiti
<EntityRef entityRef="Ego"/>
</TriggeringEntities>
<EntityCondition>
```

https://github.com/esmini/esmini

OpenSCENARIO V2.0

```
81      do parallel:
82          ego.drive() with:
83              along(sedan__route)
84              speed(16.66mps, at: start)
85          serial:
86              cut-in__vehicle.drive() with:
87                  along(sedan2__route)
88                  speed(5.5mps, slo
89                  until (cut-in__v
90              parallel:
91                  cut-in__vehicle.
92                  cut-in__vehicle.
93                      speed(15mps,
94              with:
95                  until (ego.time_
96
```

MathWorks is an ASAM Member and actively participates in the **OpenSCENARIO 2.0 Implementers Forum**

Partner with MathWorks to extend scenario workflows

Extend verification workflows with MATLAB & Simulink

RoadRunner Scenario

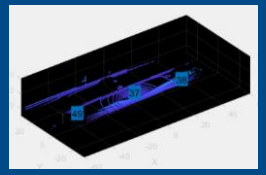
Extend actor design workflows with MATLAB & Simulink

Engage with MathWorks engineers through proof-of-concept projects or Consulting Services to generate scenes and scenarios from your recorded data

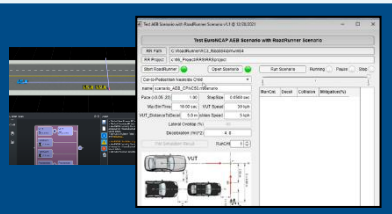
Partner with MathWorks to extend scenario workflows

Verification workflows

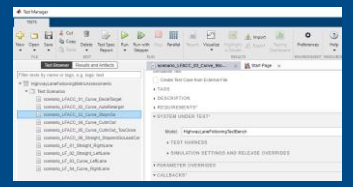
Generate scenarios from recorded data



Generate scenario variations



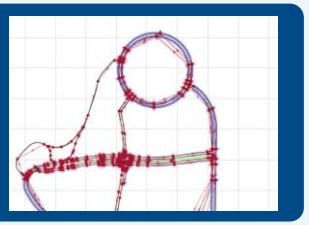
Automate testing scenarios



RoadRunner Scenario

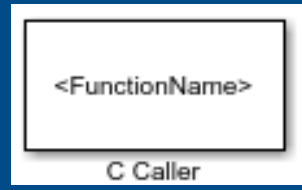
Actor design Workflows

Integrate planning & controls



Integrate sensors

Integrate external code



Engage with MathWorks engineers through proof-of-concept projects or Consulting Services to generate scenes and scenarios from your recorded data

Agenda

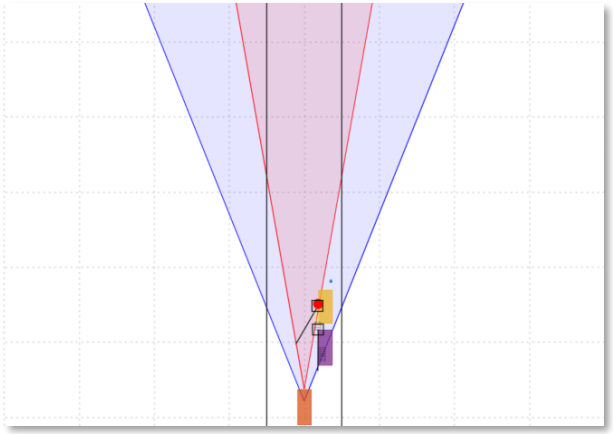
1. Introduction
2. Testing ADAS Systems
3. Cuboid scene and trajectory-driven scenario design
4. Photorealistic scene and logic-driven scenario design
5. Conclusion

Key Takeaways

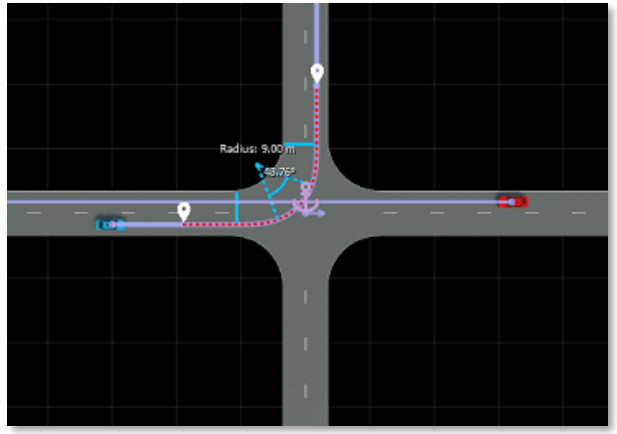
- Design and test simple driving scenarios using Automated Driving Toolbox
- Use RoadRunner to create realistic 3D scenes
- Design scenarios interactively using RoadRunner
- Perform co-simulation for scenario testing between RoadRunner and Simulink

Call to action

- Explore examples available in the documentation
- Partner with MathWorks to extend scenario workflows



[Autonomous Emergency Braking with Sensor Fusion]



[Autonomous Emergency Braking with RoadRunner Scenario]

Masterclass

Scene and Scenario Design for ADAS Simulation

October 20, 2022 | Stuttgart

Maxime François
Peter Fryscak

MathWorks