MATLAB EXPO 2018

使用 MATLAB 和 Simulink 开发自动驾驶

王鸿钧，MathWorks 中国

您可以如何使用 MATLAB 和 Simulink 开发自动驾驶算法？

感知
Perception

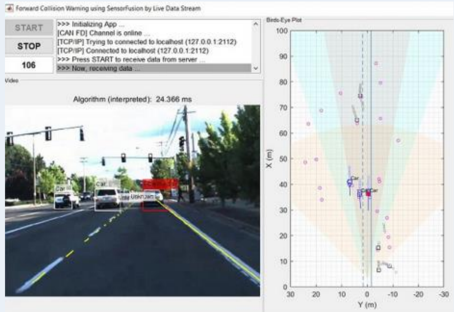
控制
Control

规划 Planning

使用 MATLAB 和 Simulink 开发自动驾驶算法的案例

深度学习

对实时数据的
传感器融合

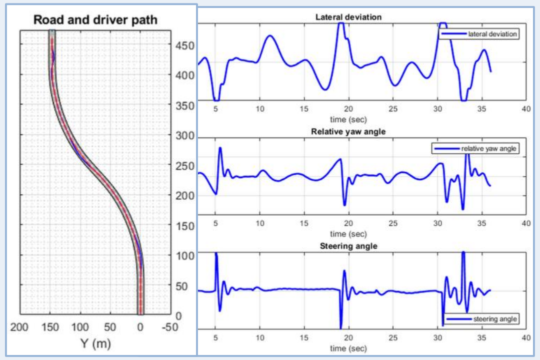


感知
Perception

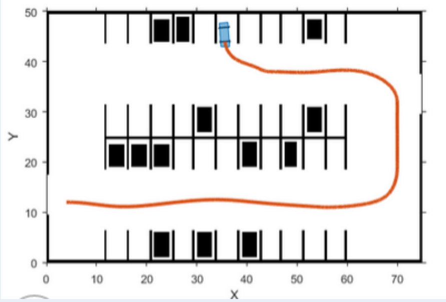
控制
Control

规划 Planning

传感器建模 &
模型预测控制



路径规划



使用 MATLAB 和 Simulink 开发自动驾驶算法的案例

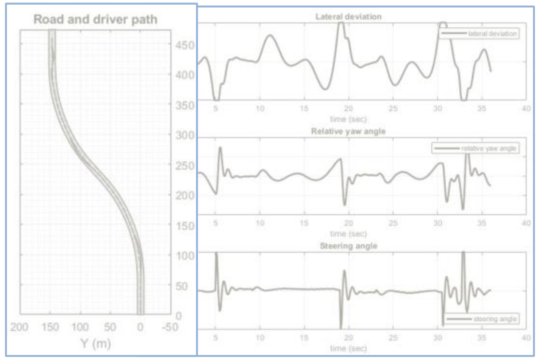
深度学习

感知
Perception

对实时数据的
传感器融合

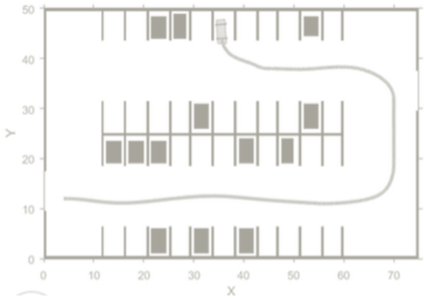
控制
Control

传感器建模 &
模型预测控制



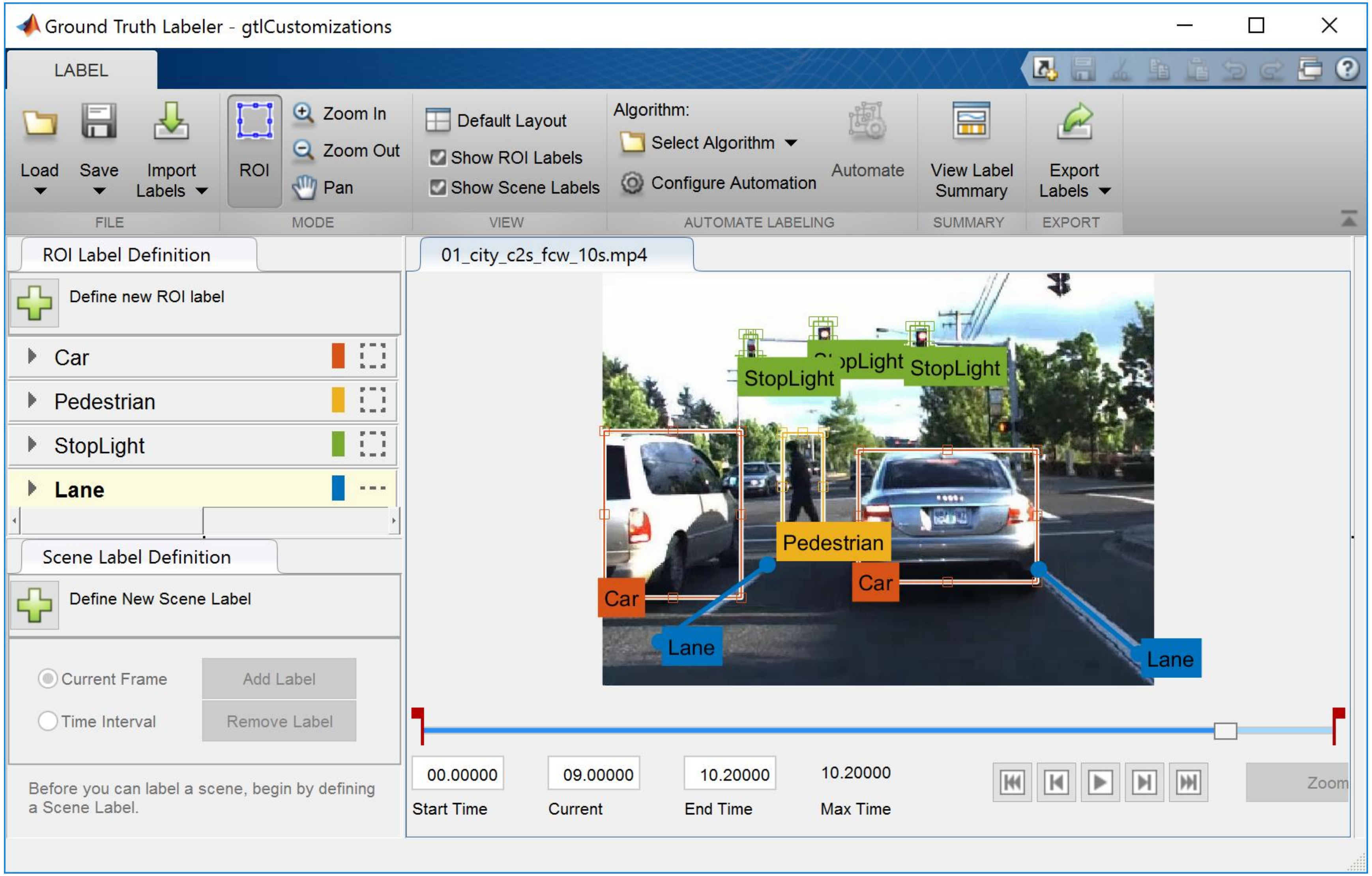
规划 Planning

路径规划



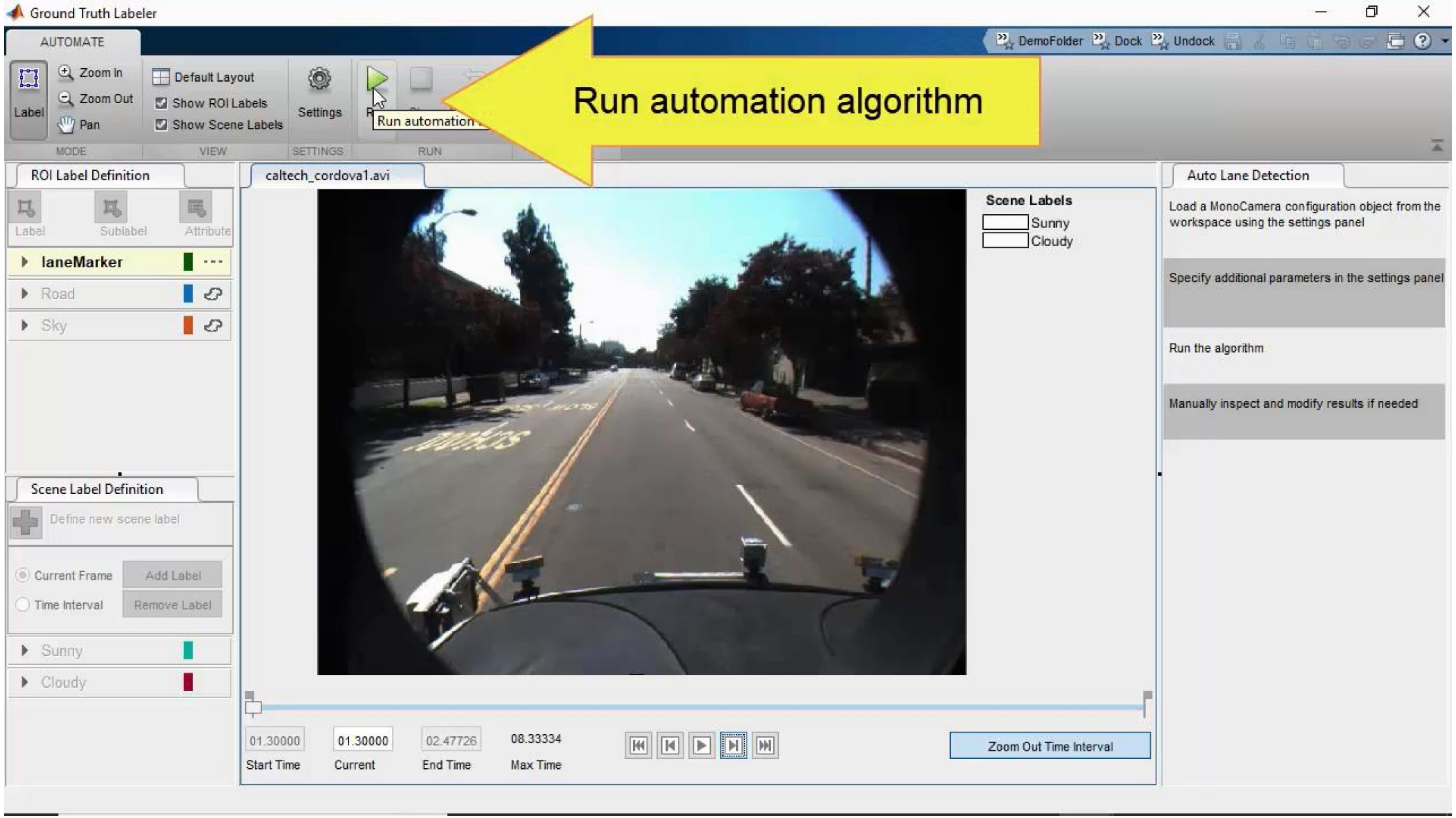
自动驾驶系统工具箱 Automated Driving System Toolbox

为标注视频数据，引入真实值标注程序（Ground Truth Labeler）



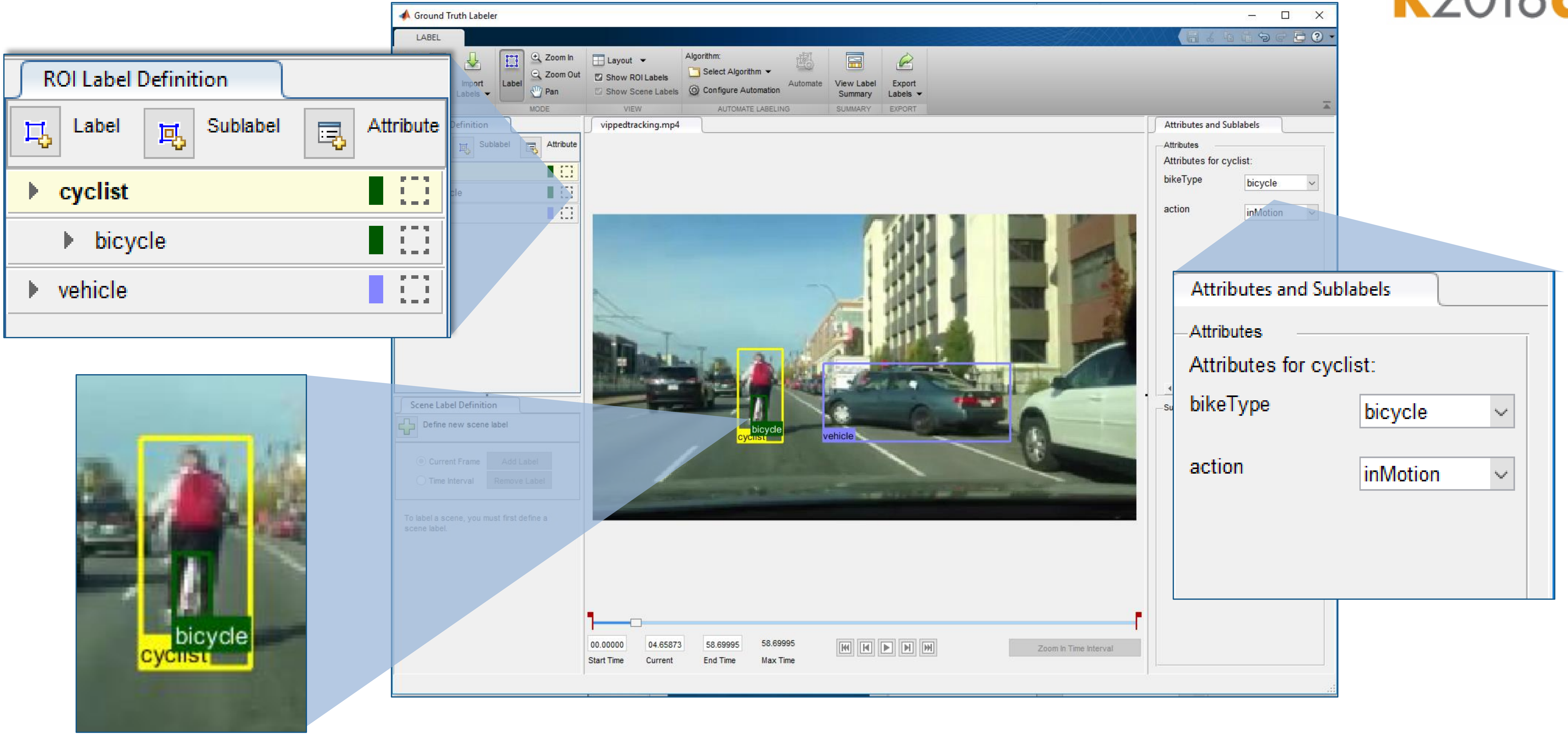
R2017a

使用 Ground Truth Labeler 自动标注车道线

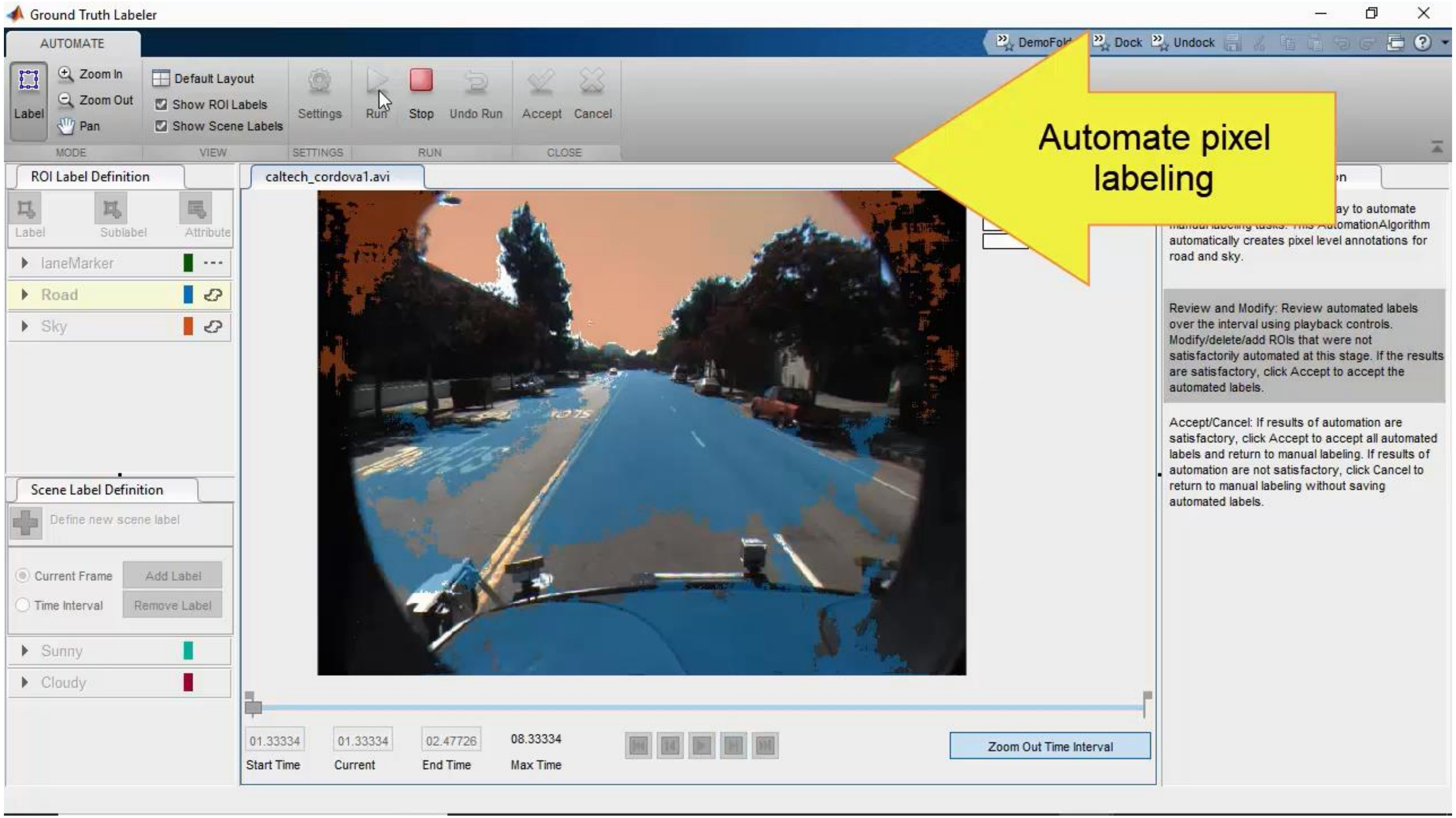


在 Ground Truth Labeler 中定义区域的子标签和属性

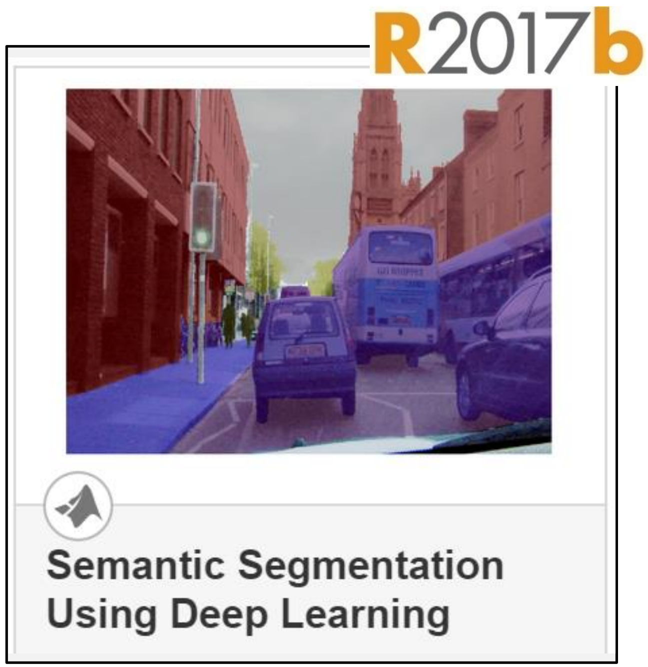
R2018a



使用 Ground Truth Labeler 自动标注像素点

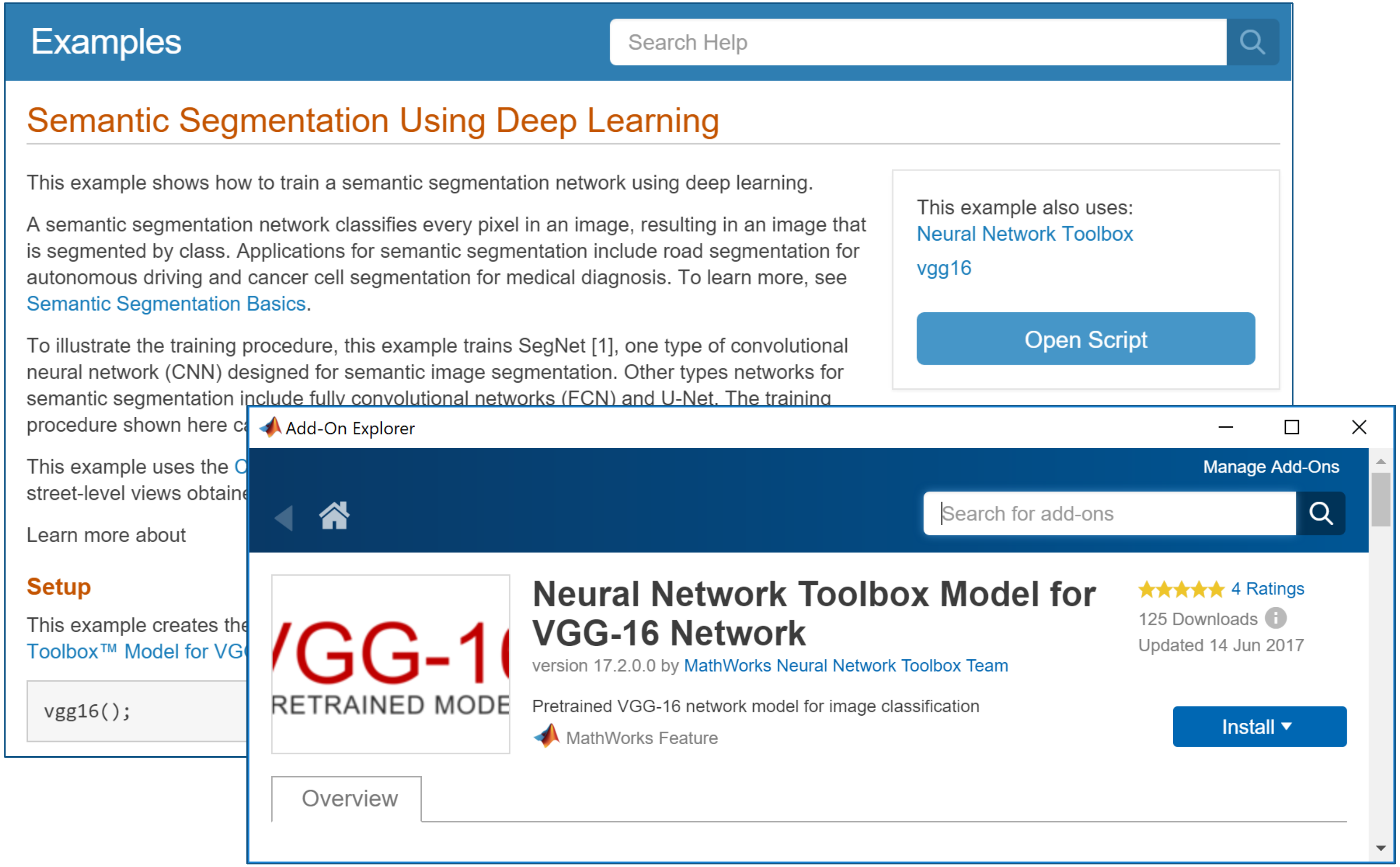


通过这个案例了解如何训练一个深度学习网络



- 使用深度学习训练可行驶区域检测网络

Computer Vision System Toolbox™



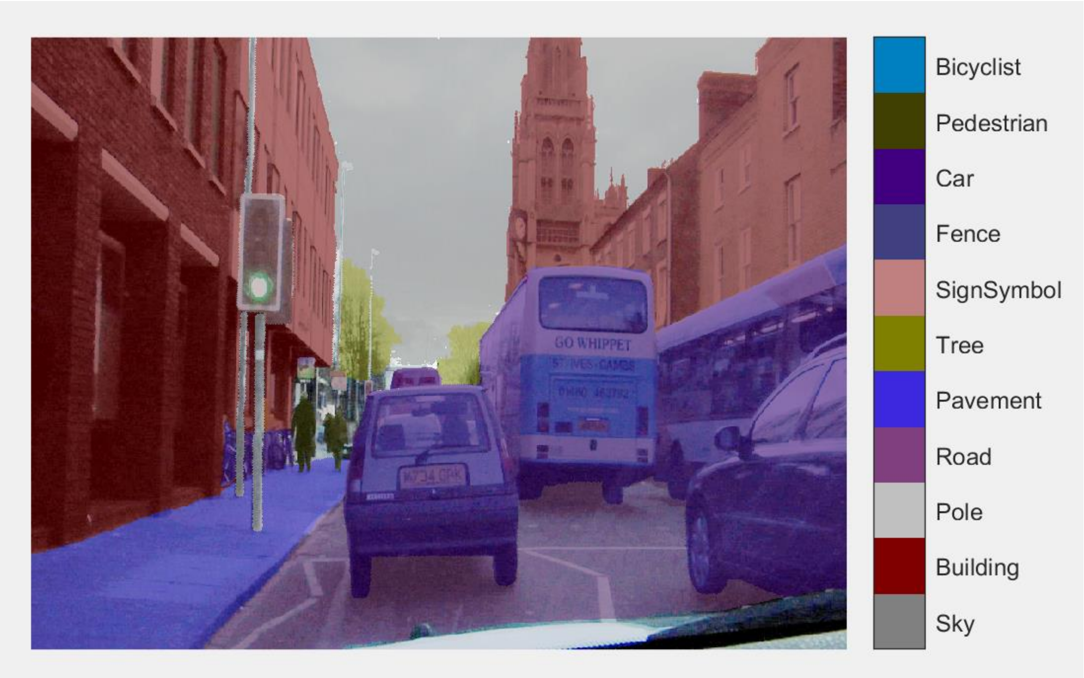
载入和显示训练图像

```
% Create datastore for images
imds = imageDatastore(imgDir);
I = readimage(imds, 1);
I = histeq(I);
imshow(I)
```

imageDatastore
可管理大型的图像数据集

载入和叠加显示被标注像素点

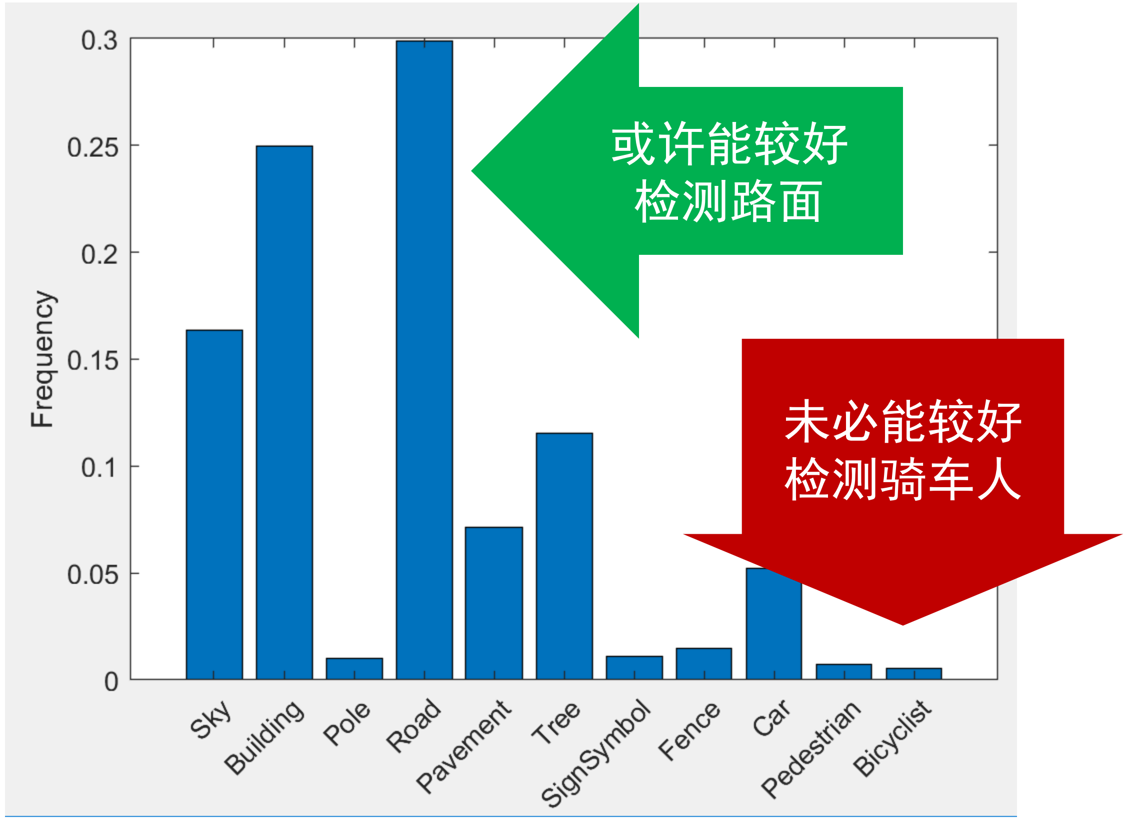
```
% Load pixel labels
classes = ["Sky";  "Building";...
  "Pole";  "Road"; "Pavement"; "Tree";...
  "SignSymbol";     "Fence";     "Car";...
  "Pedestrian";     "Bicyclist"];
pxds = pixelLabelDatastore(...
    labelDir,classes,labelIDs);

% Display labeled image
C = readimage(pxds, 1);
cmap = camvidColorMap;
B = labeloverlay(I,C,'ColorMap',cmap);
imshow(B)
```



pixelLabelDatastore
可管理大型的像素点标签数据集

可视化被标注像素点的分布

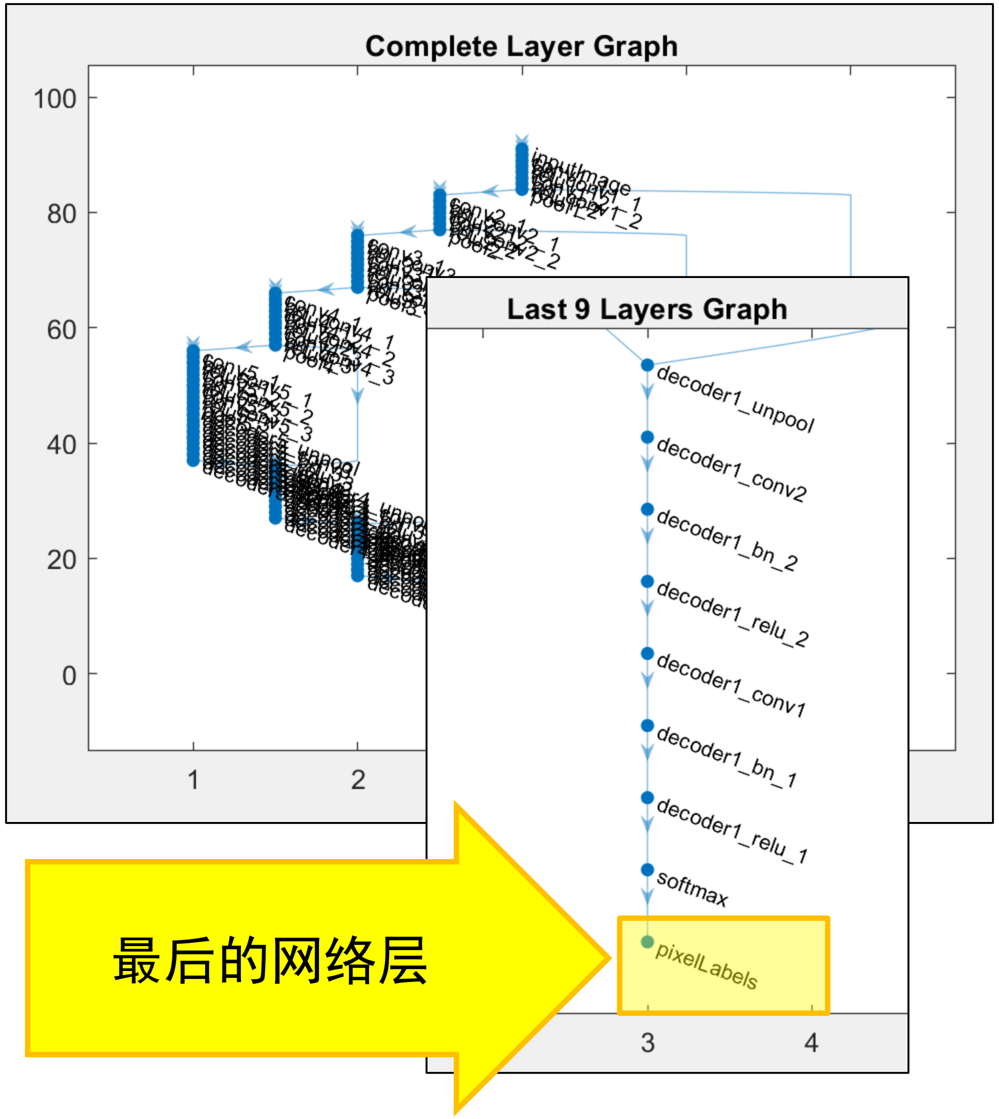
```
% Visualize label count by class
tbl = countEachLabel(pxds)

frequency = tbl.PixelCount / ...
            sum(tbl.PixelCount);

bar(1:numel(classes),frequency)
xticks(1:numel(classes))
xticklabels(tbl.Name)
xtickangle(45)
ylabel('Frequency')
```



在这个集合中被标注像素点的分布是不均衡的

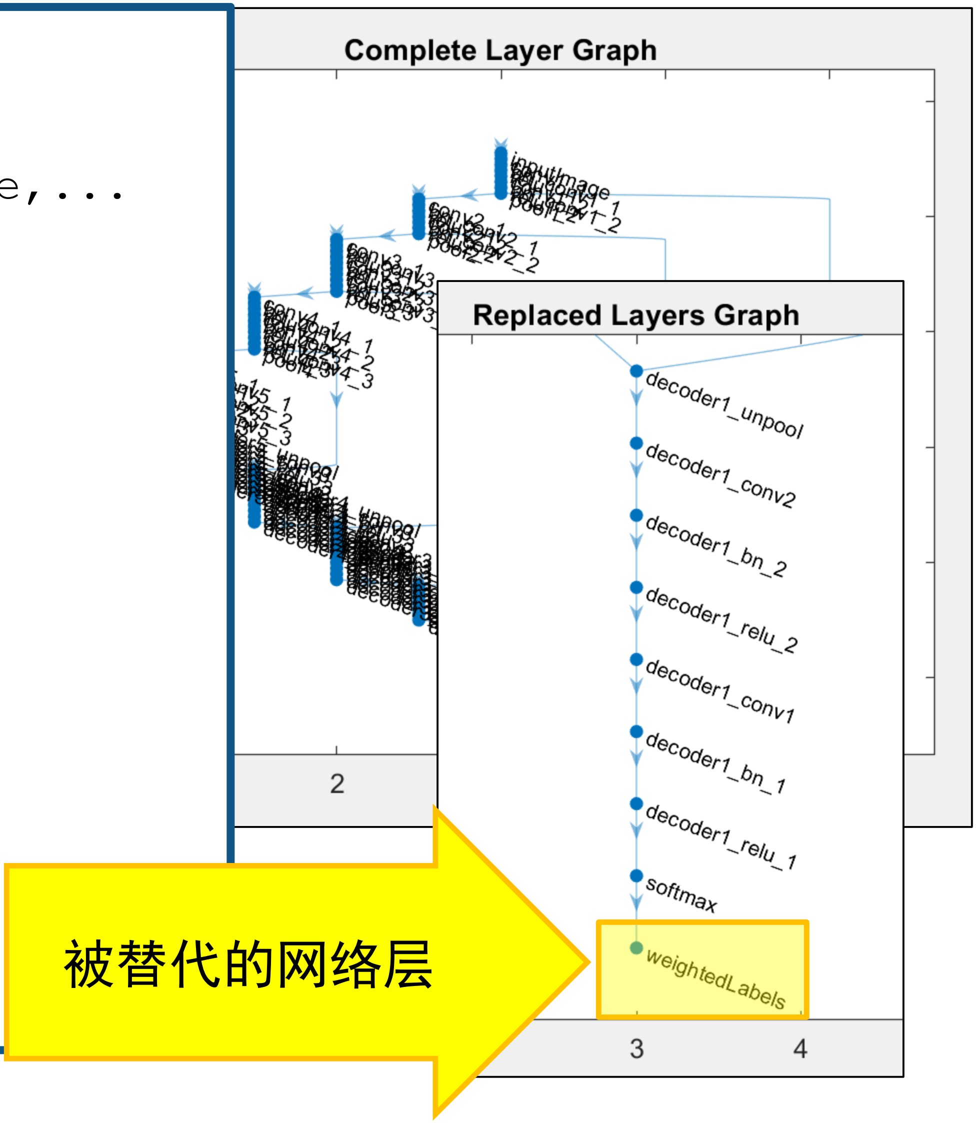
创建和可视化基础网络

```
% Create SegNet architecture
lgraph = segnetLayers(...
   imageSize, numClasses,...
   'vgg16');

% Display network structure
plot(lgraph)
title('Complete Layer Graph')%

% Display last layers
plot(lgraph); ylim([0 9.5])
title('Last 9 Layers Graph')
```



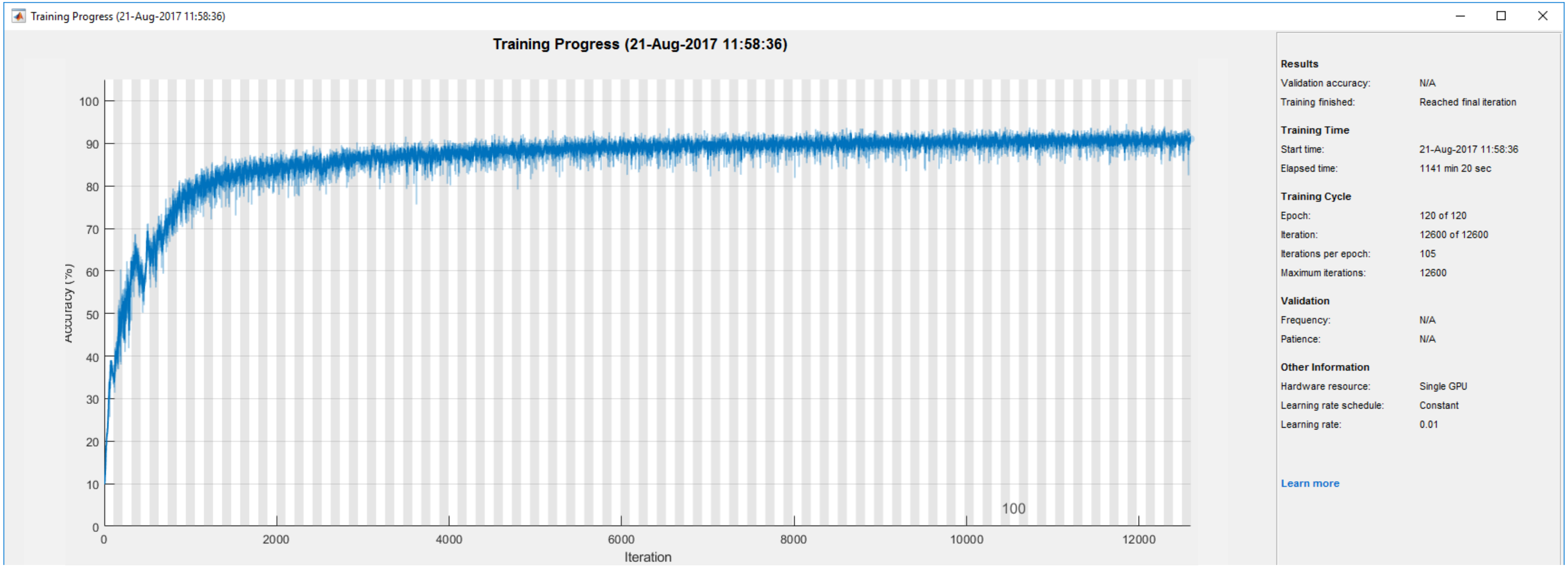
添加权重层，补偿不均衡的数据集

```
% Create weighted layer
pxLayer = pixelClassificationLayer(...
    'Name','weightedLabels', 'ClassNames',tbl.Name,...
    'ClassWeights',classWeights)

% Replace layer
lgraph = removeLayers(lgraph, 'pixelLabels');
lgraph = addLayers(lgraph, pxLayer);
lgraph = connectLayers(lgraph,...
    'softmax', 'weightedLabels');

% Display network structure
plot(lgraph); ylim([0 9.5])
title('Replaced Layers Graph')
```



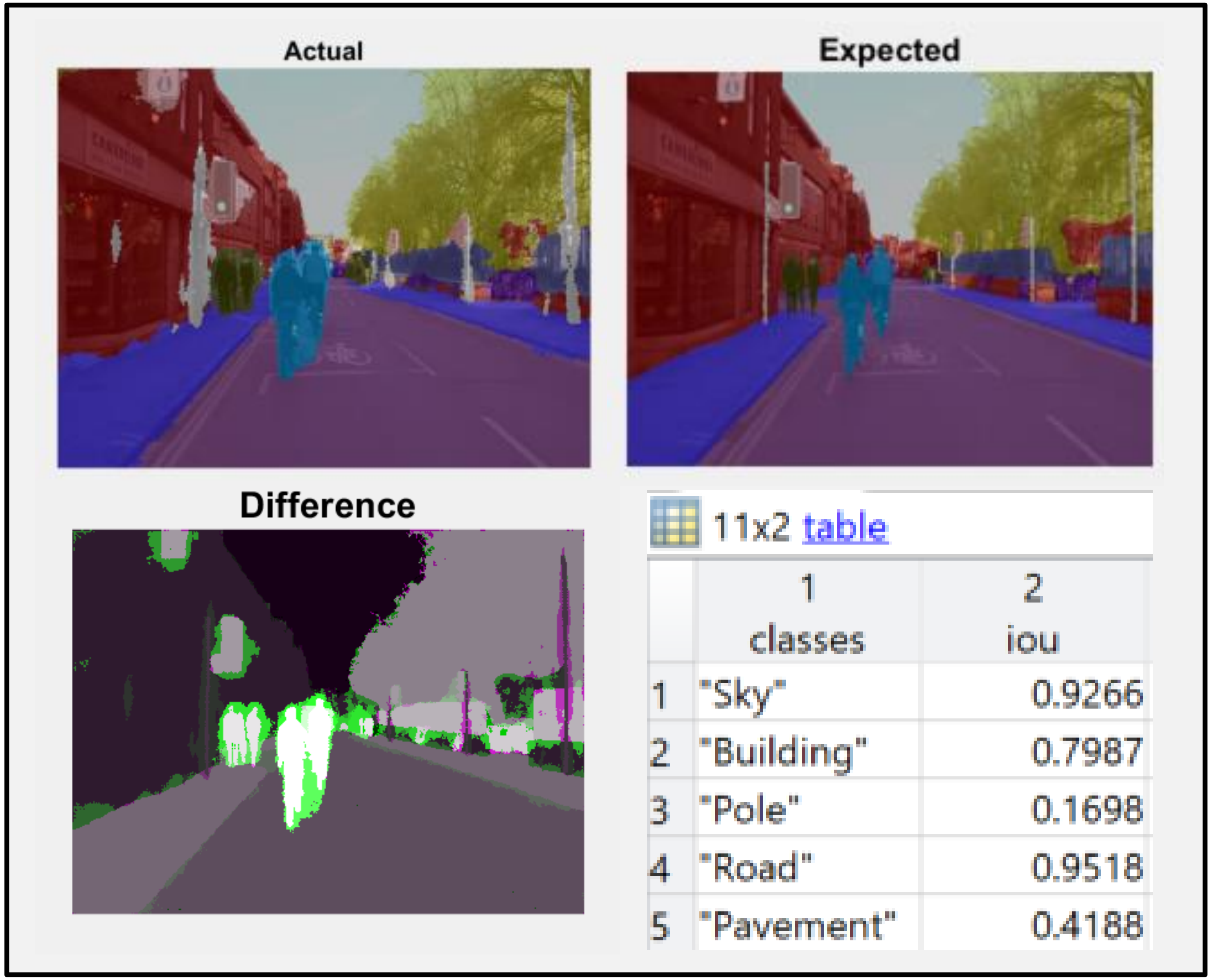
训练网络并观察训练过程

```
[net, info] = trainNetwork(datasource, lgraph, options);
```



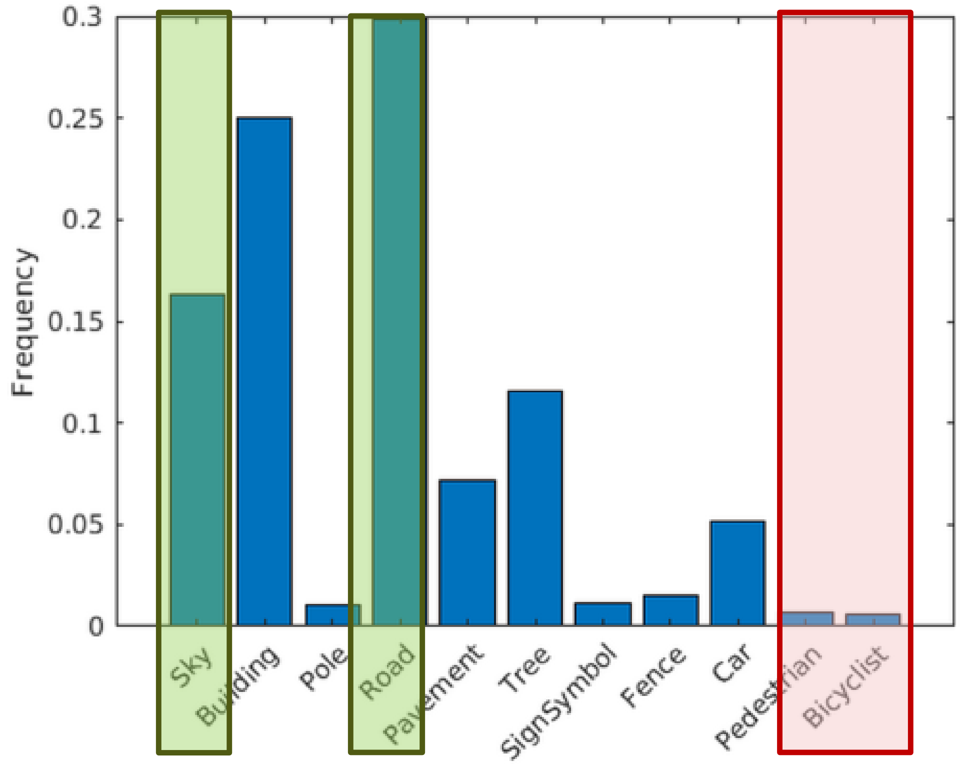
计量交并比（IoU），评估相似度

```
iou = jaccard(actual,...
              expected);
table(classes,iou)

ans =
  11×2 table
      classes          iou
    ____________    _______

    "Sky"            0.92659
    "Building"        0.7987
    "Pole"           0.16978
    "Road"           0.95177
    "Pavement"       0.41877
    "Tree"           0.43401
    "SignSymbol"     0.32509
    "Fence"            0.492
    "Car"           0.068756
    "Pedestrian"           0
    "Bicyclist"            0
```



数据中的标签分布对 IoU 的影响

在原始数据集中的标签分布



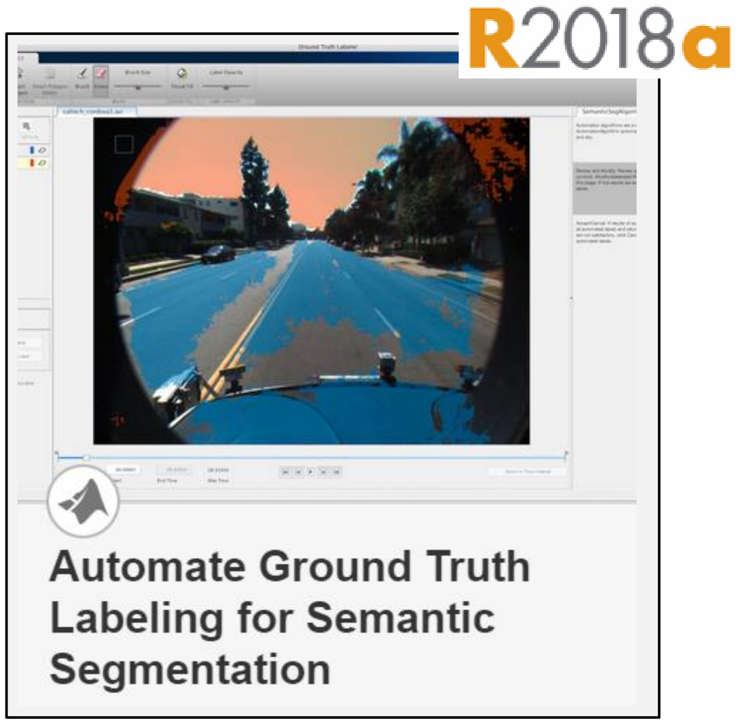
网络的计量评估结果

	Accuracy	IoU	MeanBFScore
Sky	0.93544	0.89279	0.88239
Building	0.79978	0.75543	0.59861
Pole	0.73166	0.18361	0.51426
Road	0.93644	0.90663	0.7086
Pavement	0.90624	0.72932	0.70585
Tree	0.86587	0.73694	0.67097
SignSymbol	0.76118	0.35339	0.44175
Fence	0.83258	0.49648	0.50265
Car	0.90961	0.75263	0.64837
Pedestrian	0.83751	0.35409	0.46796
Bicyclist	0.84156	0.5472	0.46933

未被充分表现的类例如行人、骑车人，没有像天空、道路一样获得良好的分割

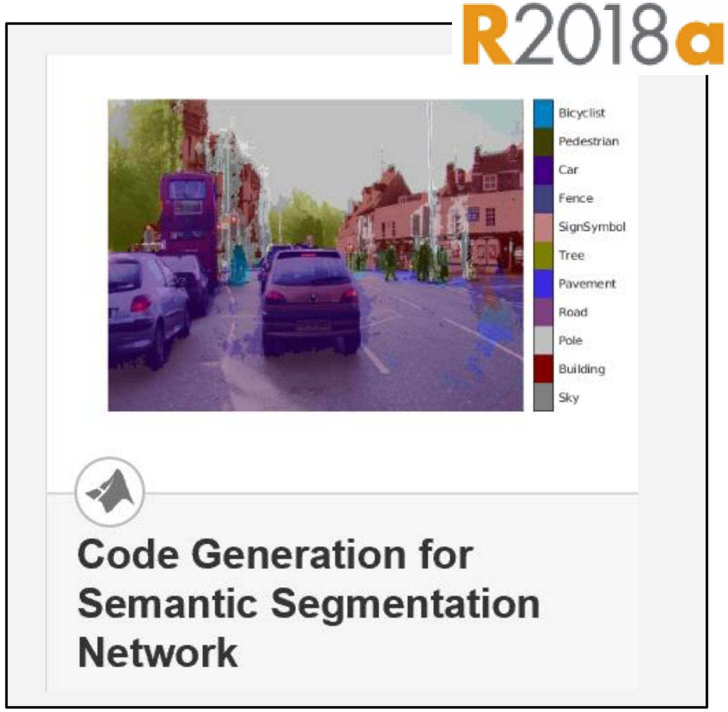
使用语义分割检测可行驶区域

了解更多的深度学习感知算法开发案例



- **添加语义分割自动算法**
  到 Ground Truth Labeler 工具
  Automated Driving
  System Toolbox™

- **训练深度学习网络**
  可行驶空间检测应用
  Computer Vision
  System Toolbox™

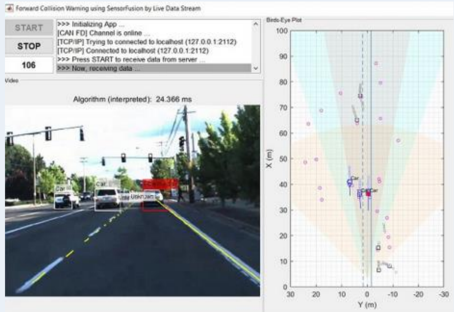


- **生成 CUDA® 代码**
  在 NVIDIA GPU 上运行 DAG 网络
  GPU Coder™

使用 MATLAB 和 Simulink 开发自动驾驶算法的案例

深度学习

对实时数据的
传感器融合

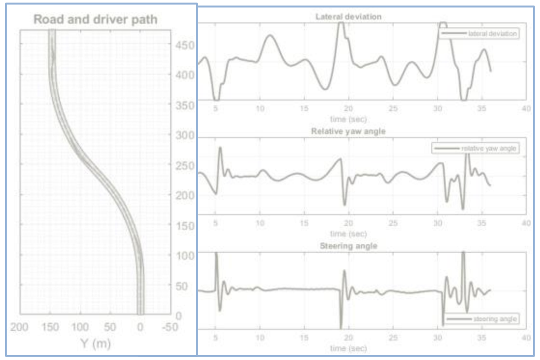


感知
Perception

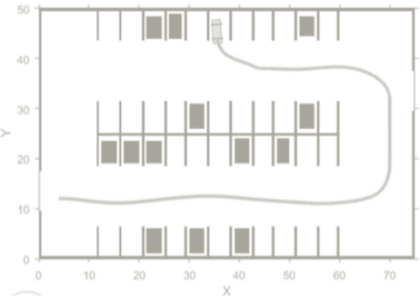
控制
Control

规划 Planning

传感器建模 &
模型预测控制

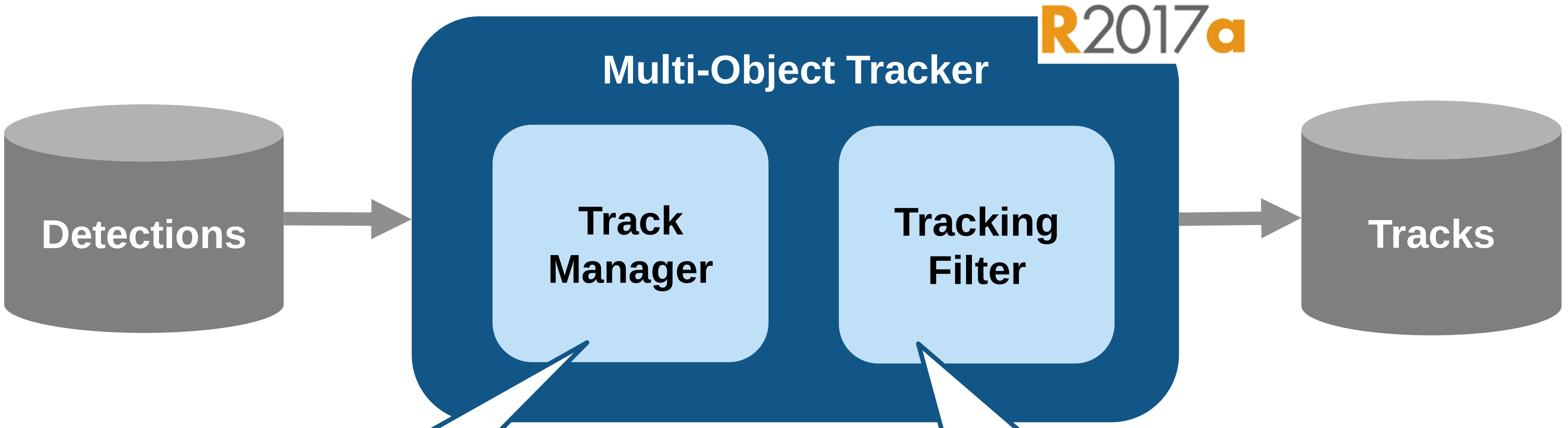


路径规划



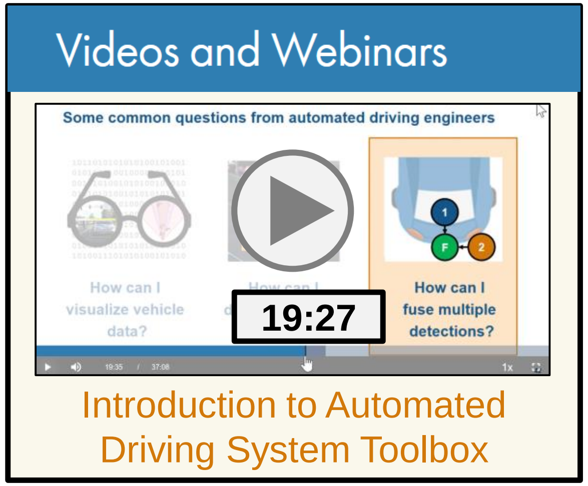
自动驾驶系统工具箱 Automated Driving System Toolbox

为开发传感器融合算法，引入多目标跟踪器（Multi-object tracker）

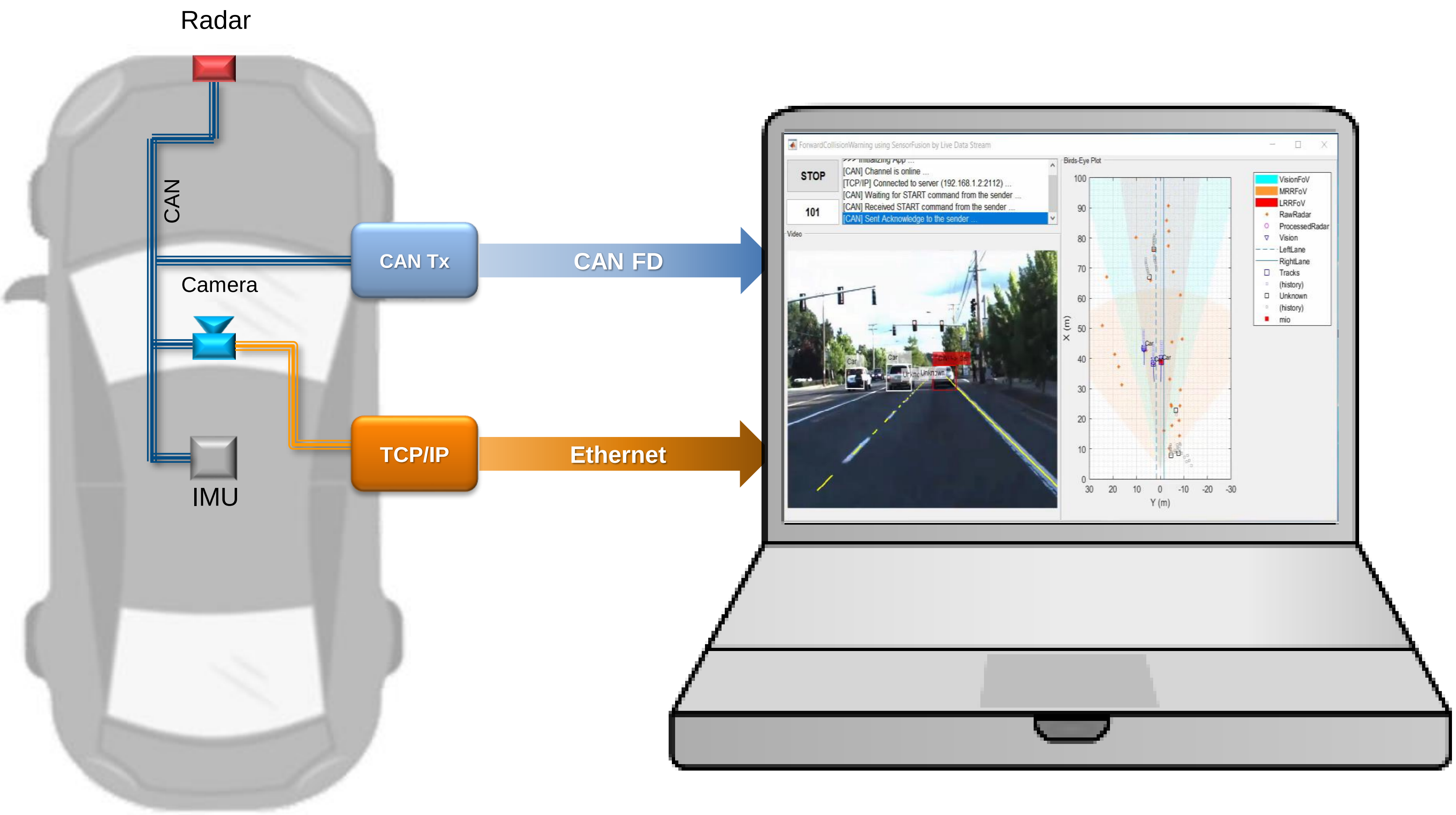


- 分配检测结果到目标跟踪
- 创建新的目标跟踪
- 更新已经存在的目标跟踪
- 删除过期的目标跟踪

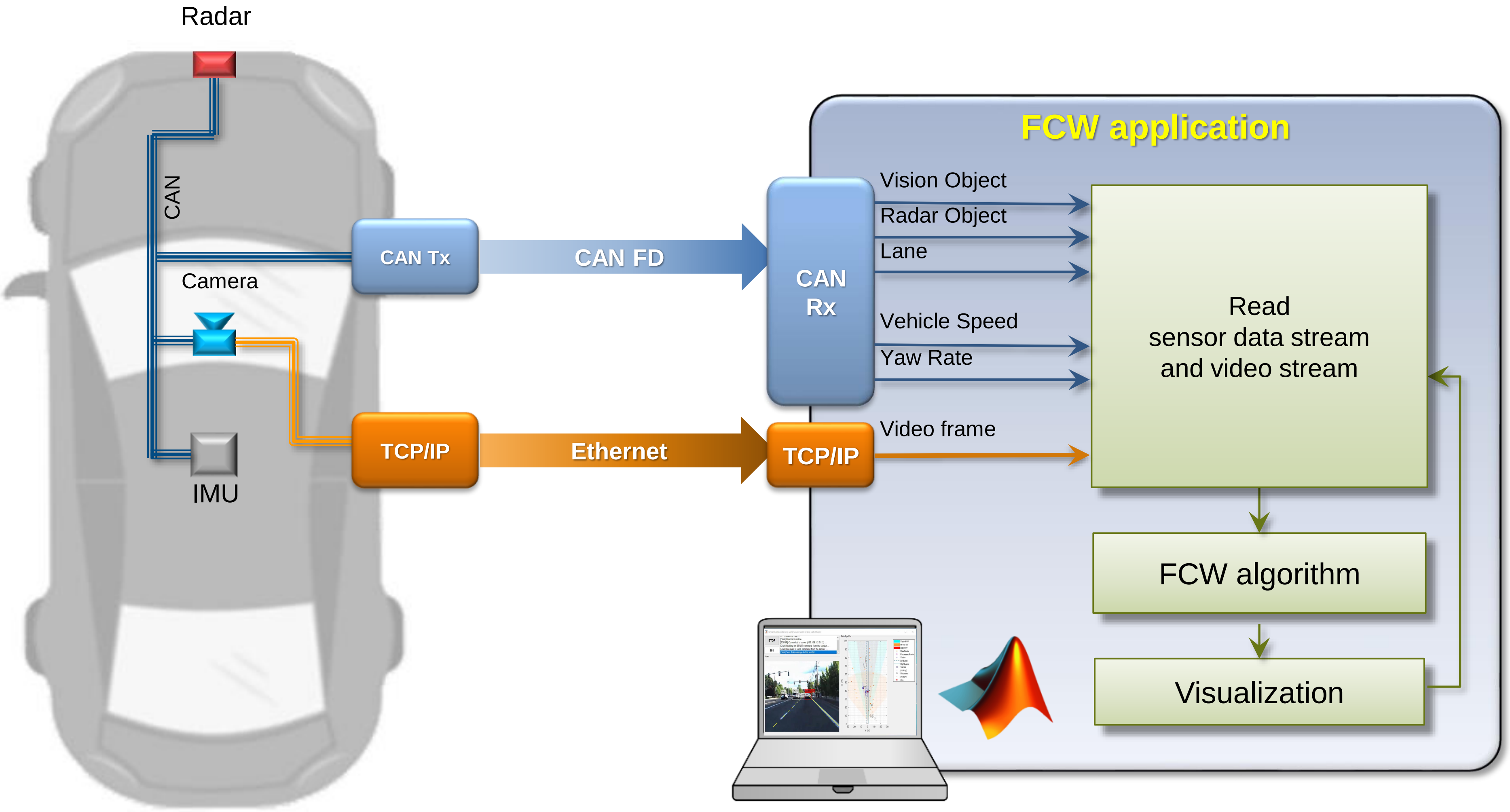
- 目标跟踪的状态预测与更新
- 支持线性、扩展、无迹卡尔曼滤波



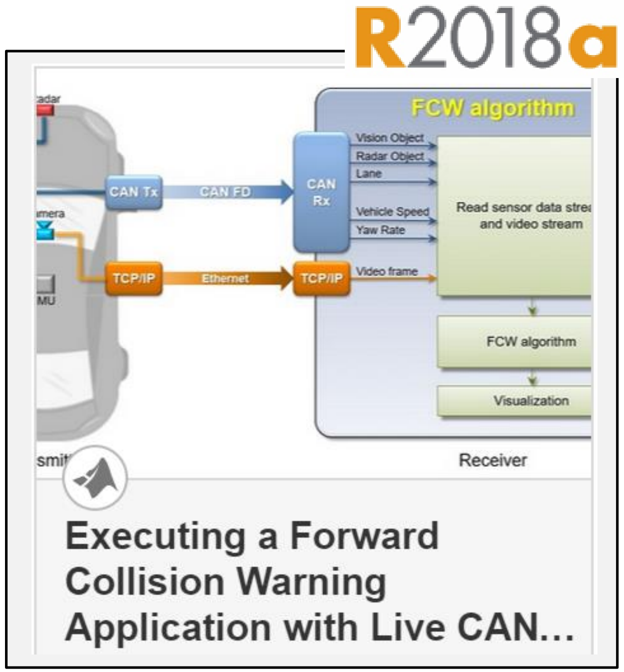
如何使用实时数据测试我的传感器融合算法？



使用来自车辆的实时数据测试前向碰撞预警算法



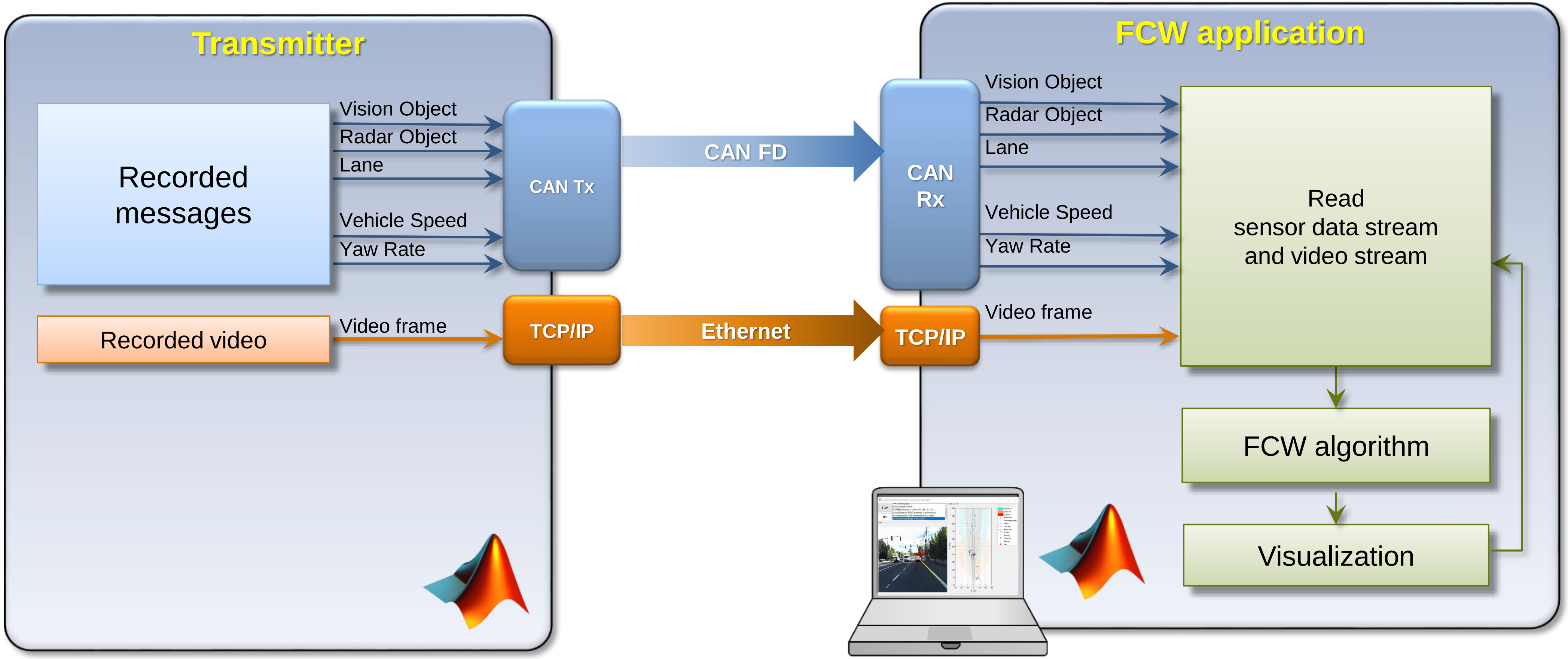
通过这个案例了解如何开发对实时数据的传感器融合算法



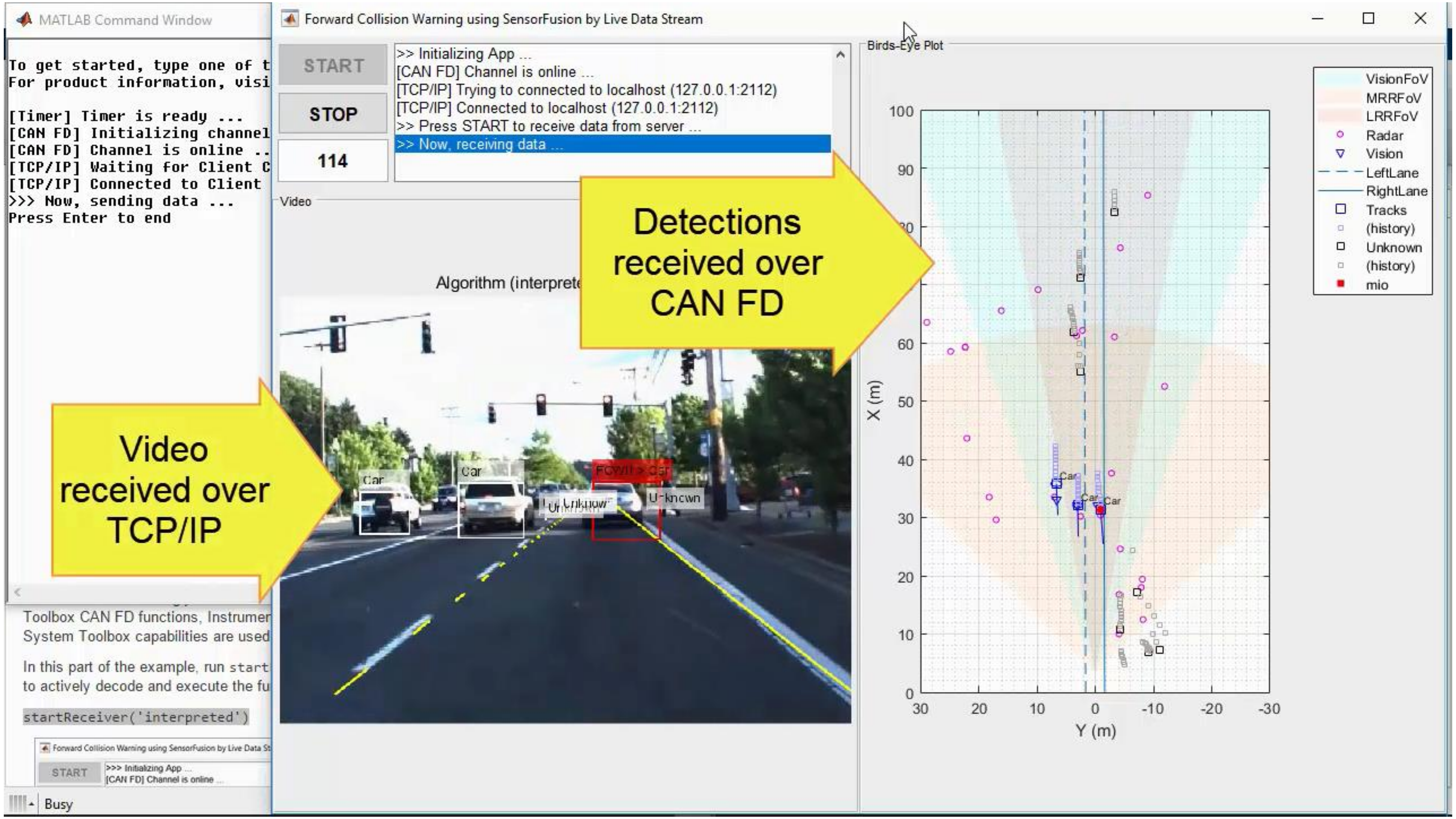
- **传输 CAN FD 数据流**
  在笔记本电脑上验证原型算法

  Vehicle Network Toolbox™

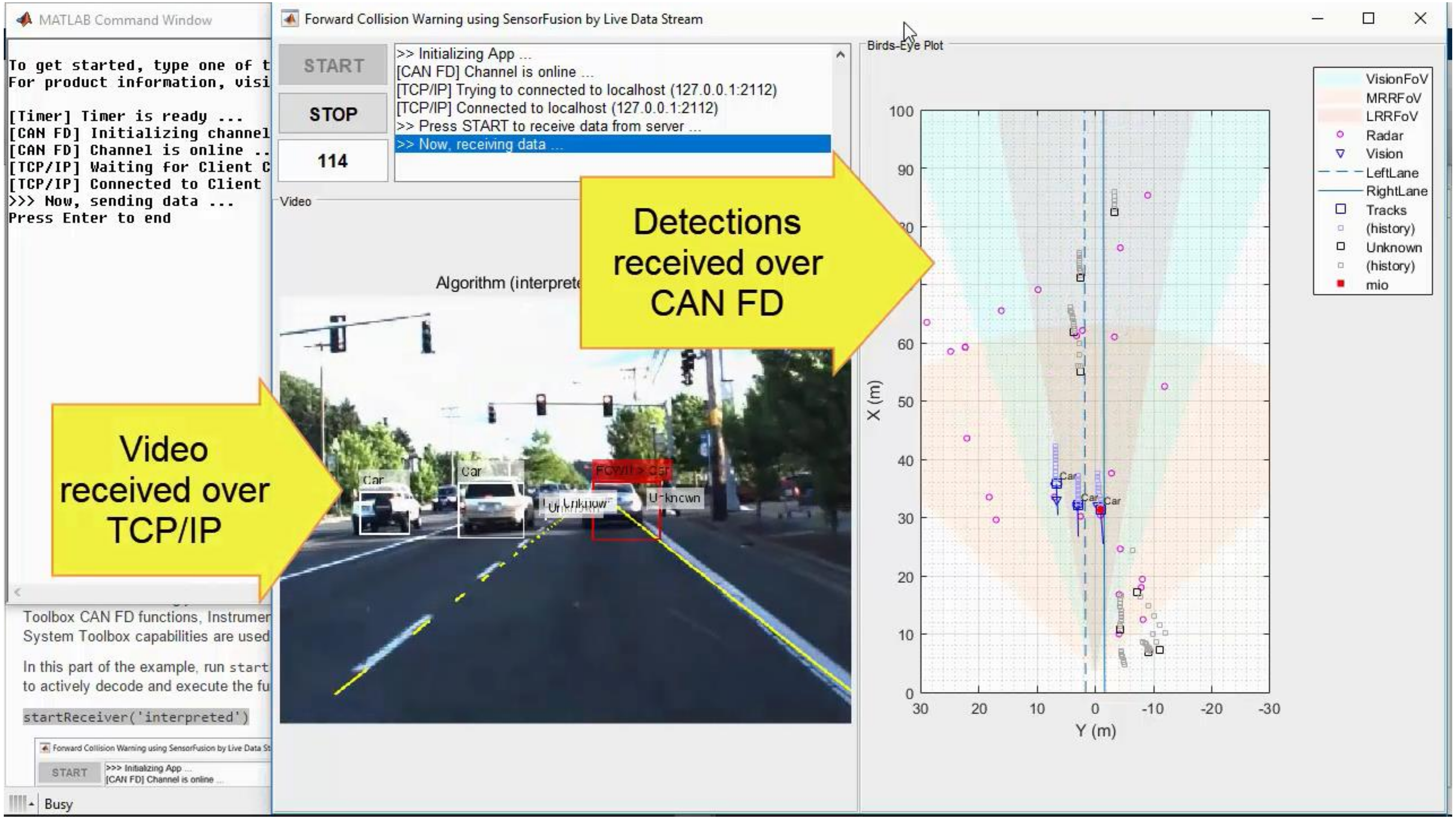
使用来自车辆替代者的实时数据测试前向碰撞预警算法



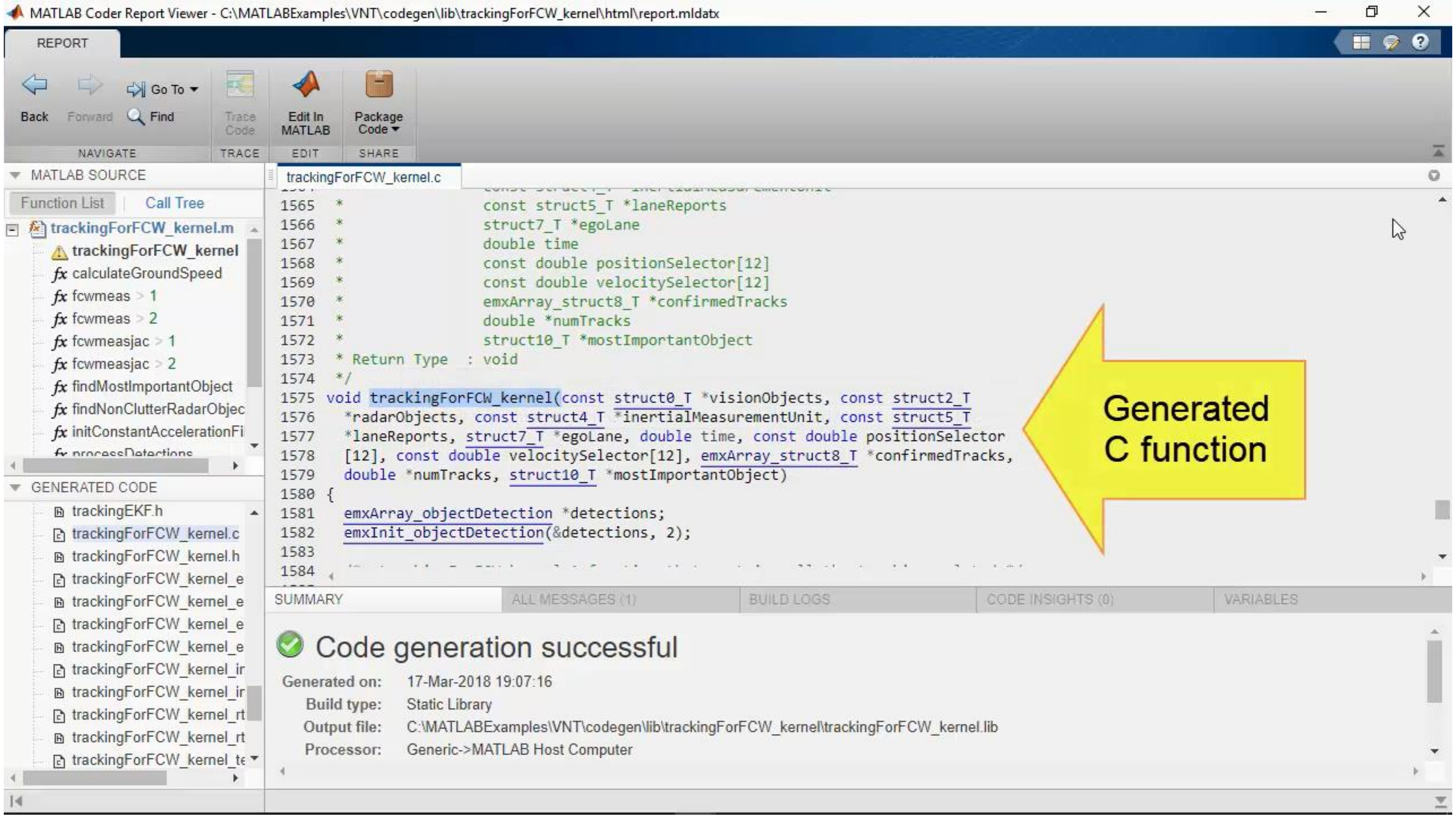
通过 CAN FD 和 TCP/IP 发送实时数据



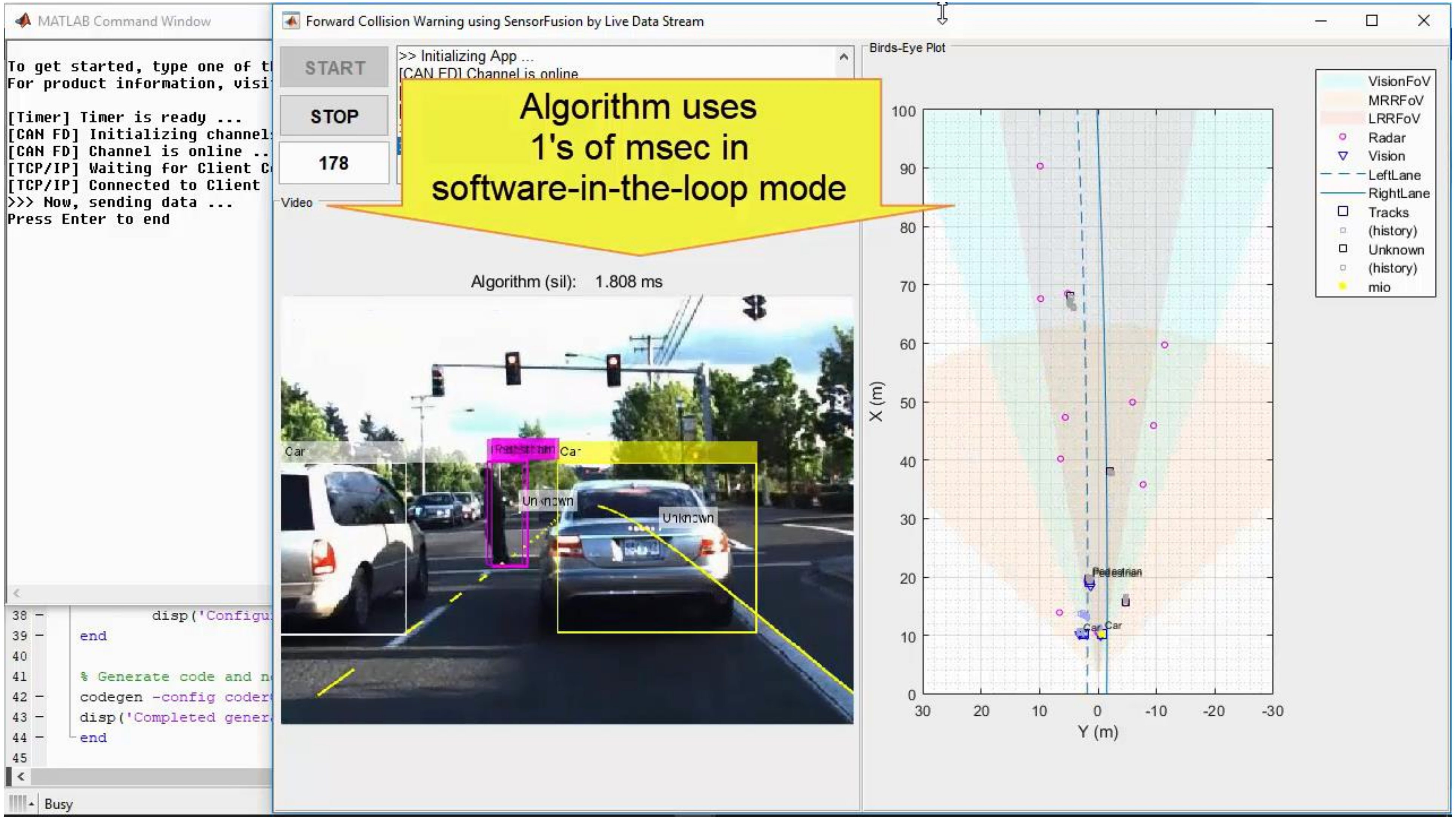
通过 CAN FD 和 TCP/IP 接收实时数据



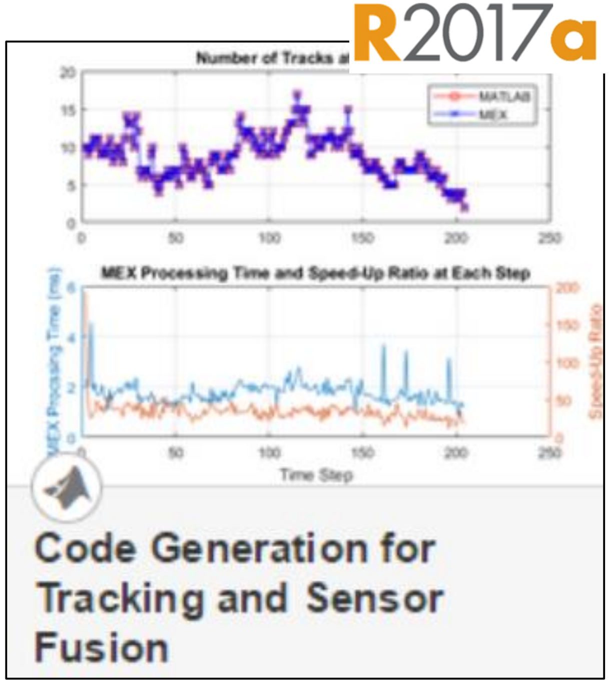
生成前向碰撞预警算法的 C/C++ 代码

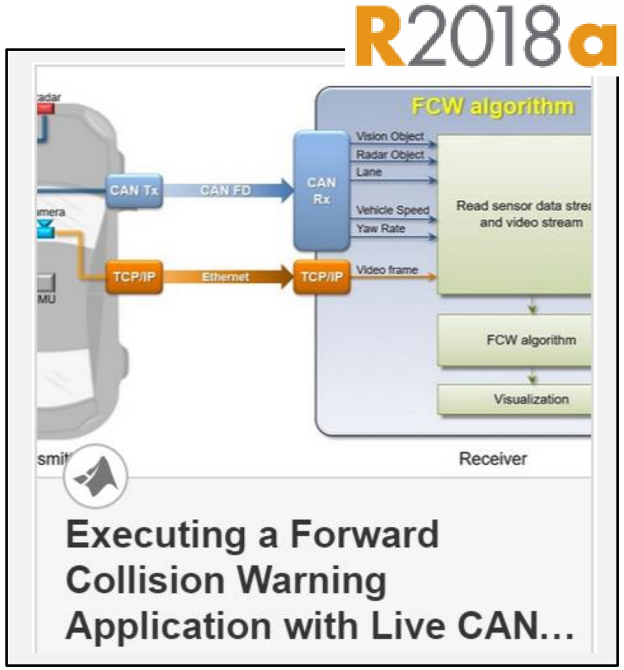


通过 CAN FD 和 TCP/IP 传输实时数据流，并运行编译后的代码



了解更多的传感器融合算法开发案例





- **使用记录的数据设计**
  包含多目标跟踪器的算法
  Automated Driving System Toolbox™

- **生成 C/C++ 代码**
  对包含多目标跟踪器的算法
  MATLAB Coder™

- **传输 CAN FD 数据流**
  在笔记本电脑上验证原型算法
  Vehicle Network Toolbox™

使用 MATLAB 和 Simulink 开发自动驾驶算法的案例

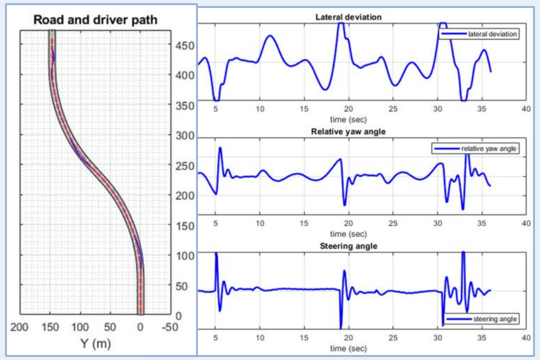
深度学习

对实时数据的
传感器融合

感知
Perception

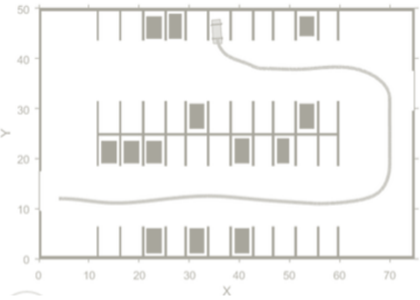
控制
Control

传感器建模 &
模型预测控制



规划 Planning

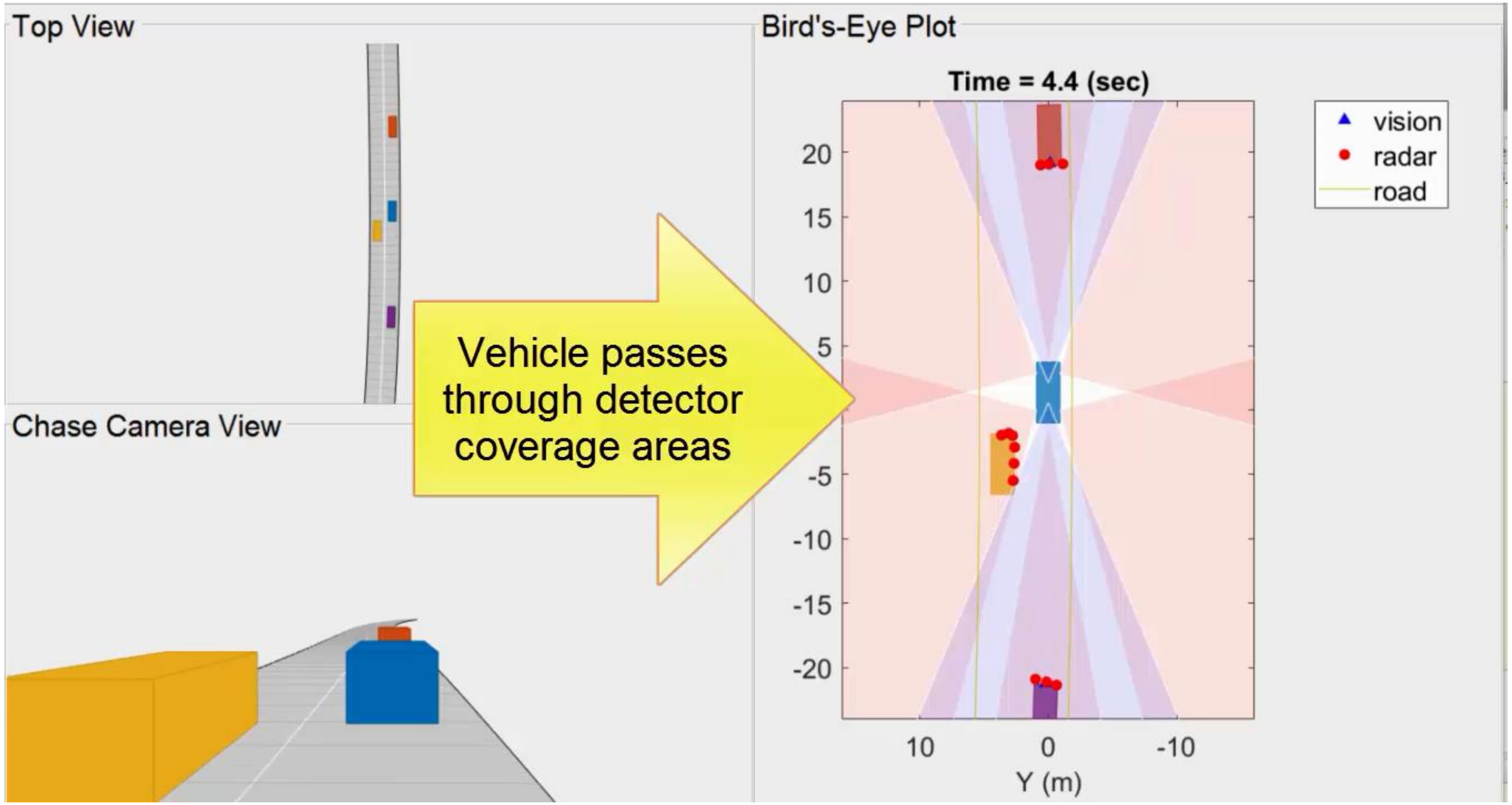
路径规划



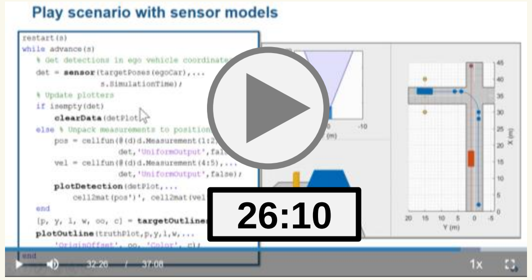
自动驾驶系统工具箱 Automated Driving System Toolbox

人工构建场景，测试传感器融合算法

R2017a



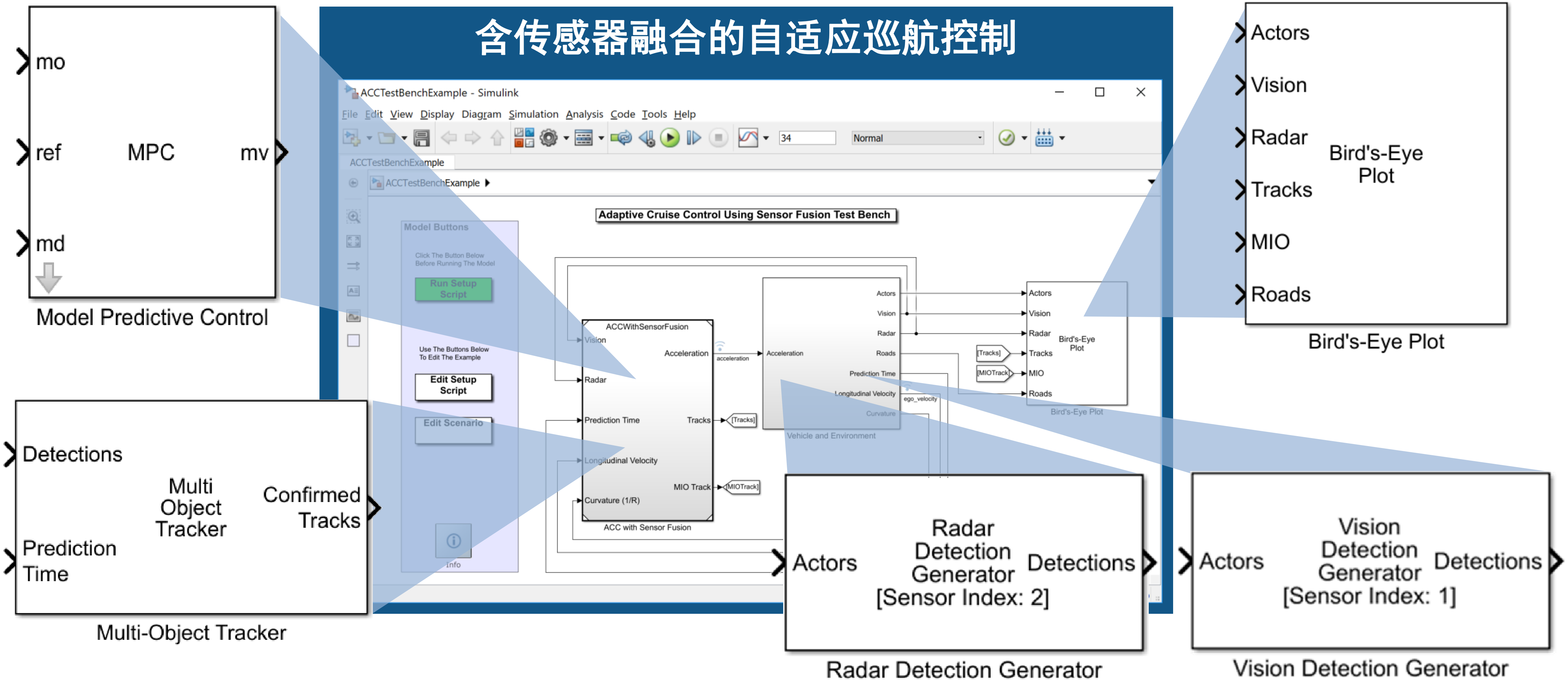
Videos and Webinars



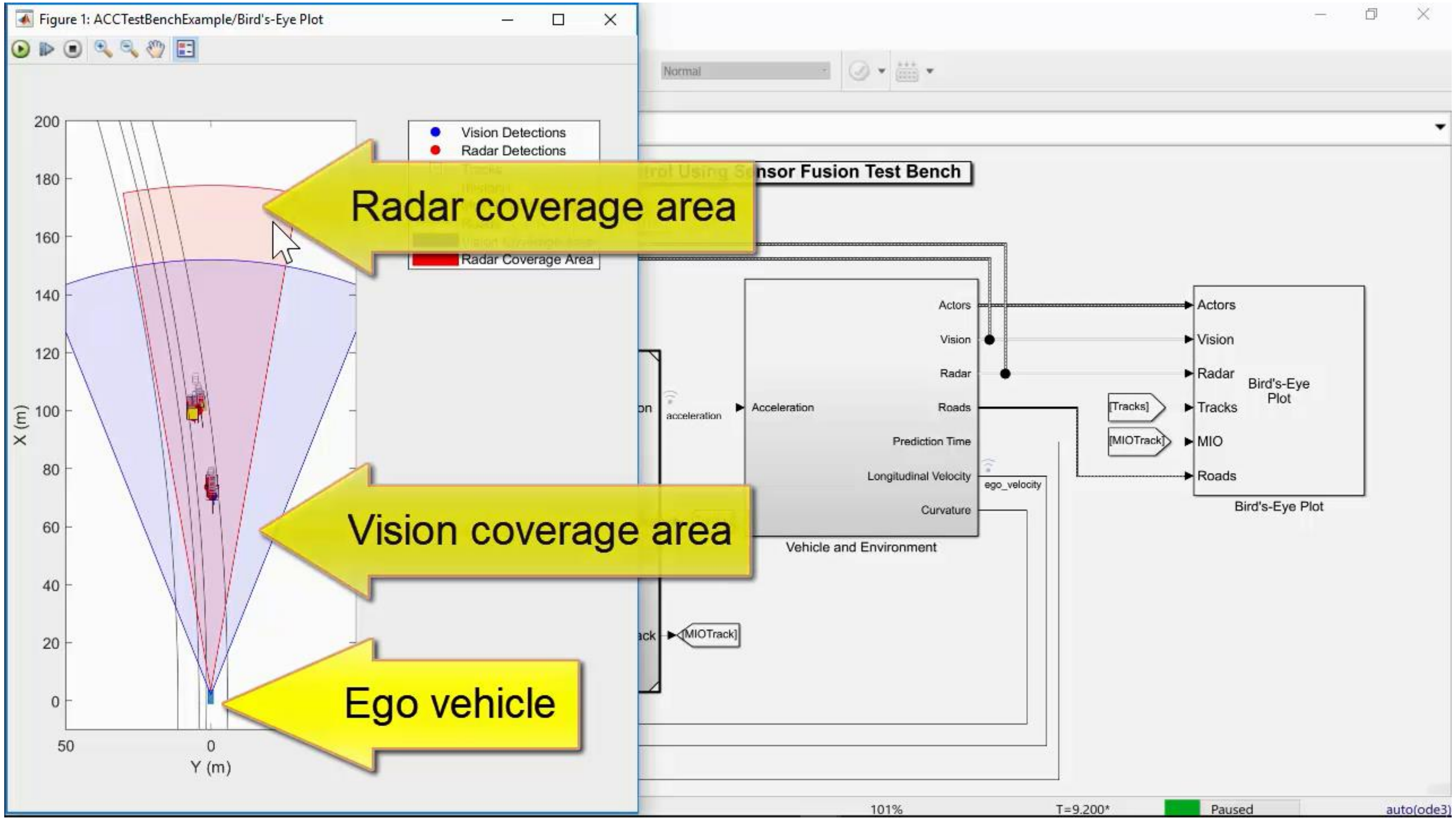
Introduction to Automated Driving System Toolbox

使用雷达和视觉检测器、传感器融合、模型预测控制等模块，进行系统的闭环仿真

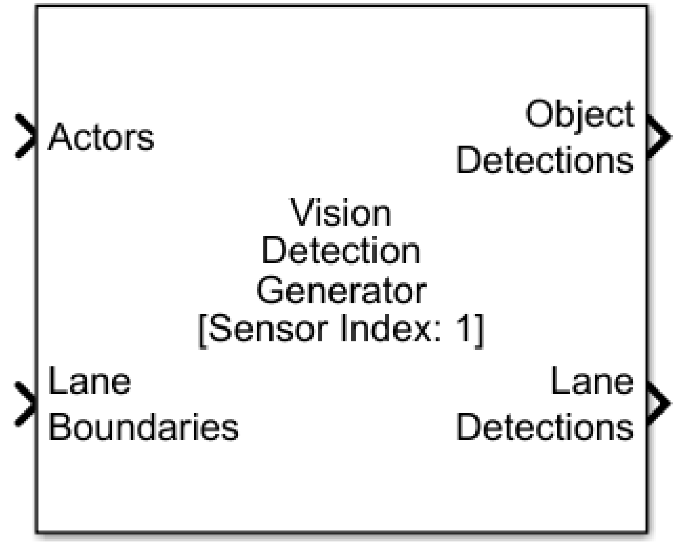
R2017b



模拟传感器检测结果，测试传感器融合与模型预测控制算法

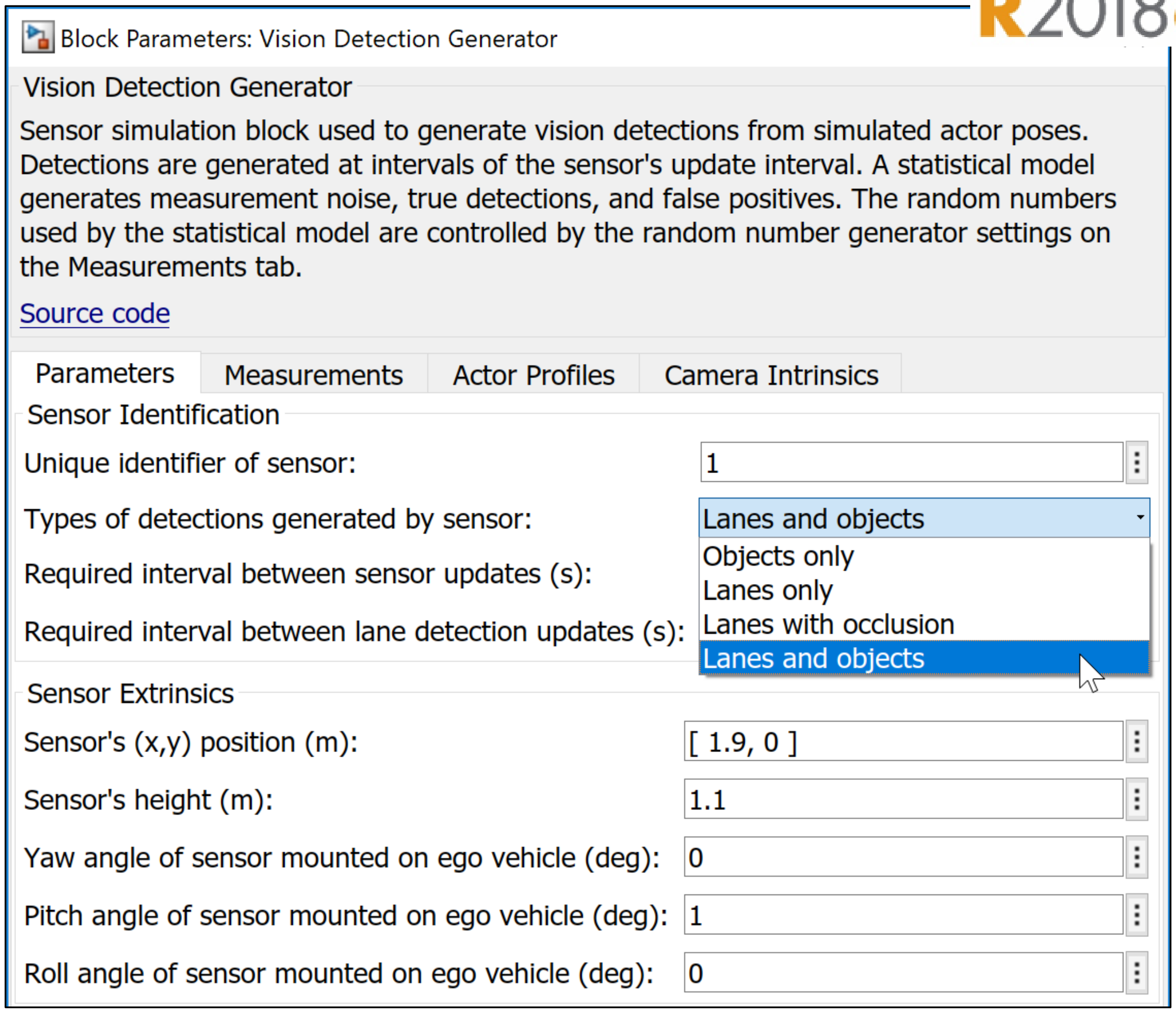


使用视觉检测器模拟车道线检测



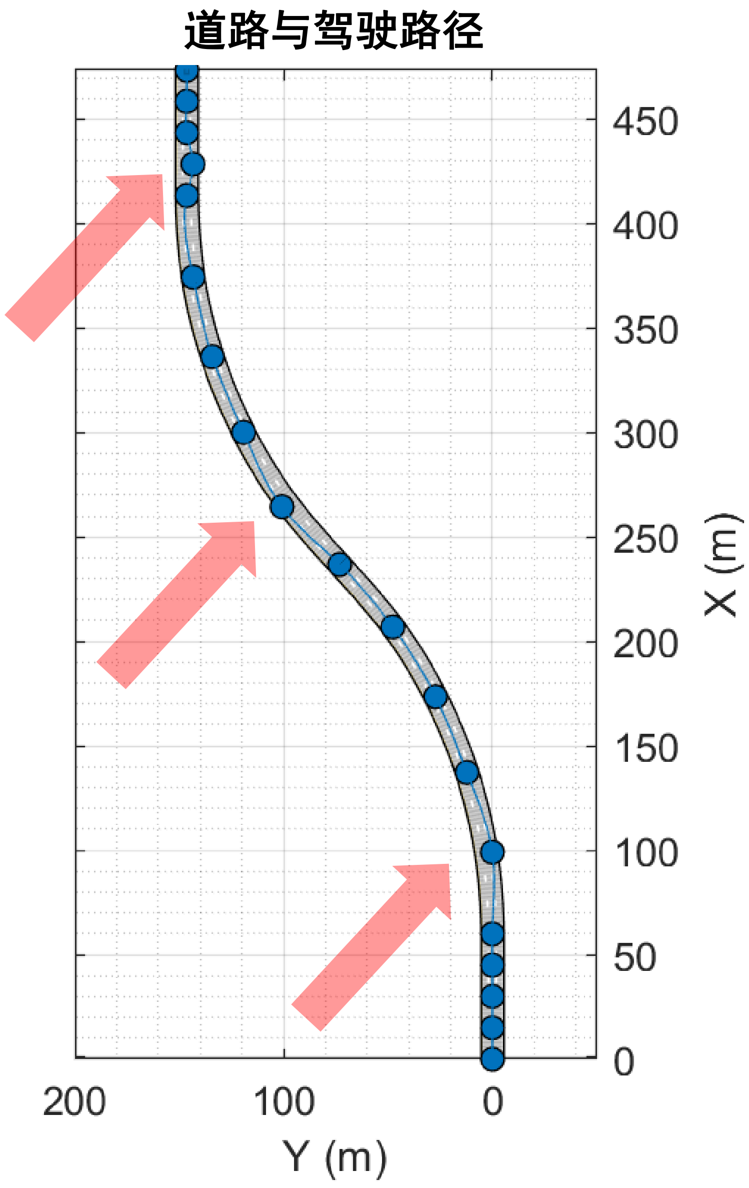
Vision Detection Generator

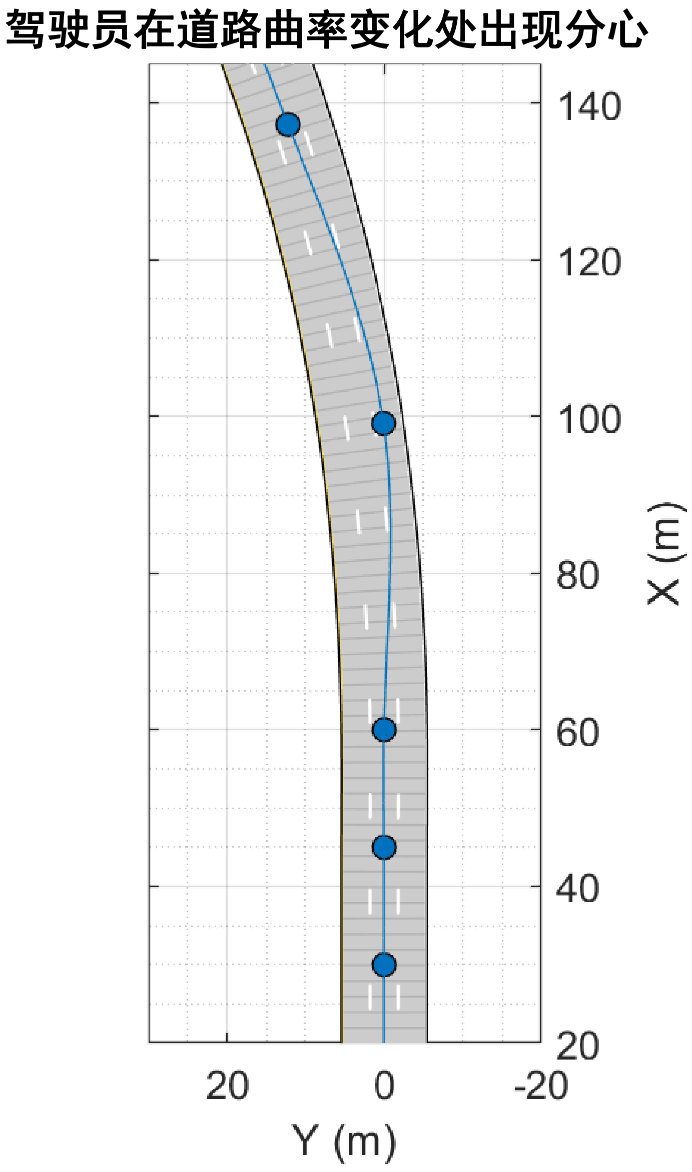
R2018a



使用驾驶场景工具（drivingScenario）创建双曲线道路

- 模拟驾驶路径
- 驾驶员在曲率变化处出现分心





分心驾驶员的仿真结果

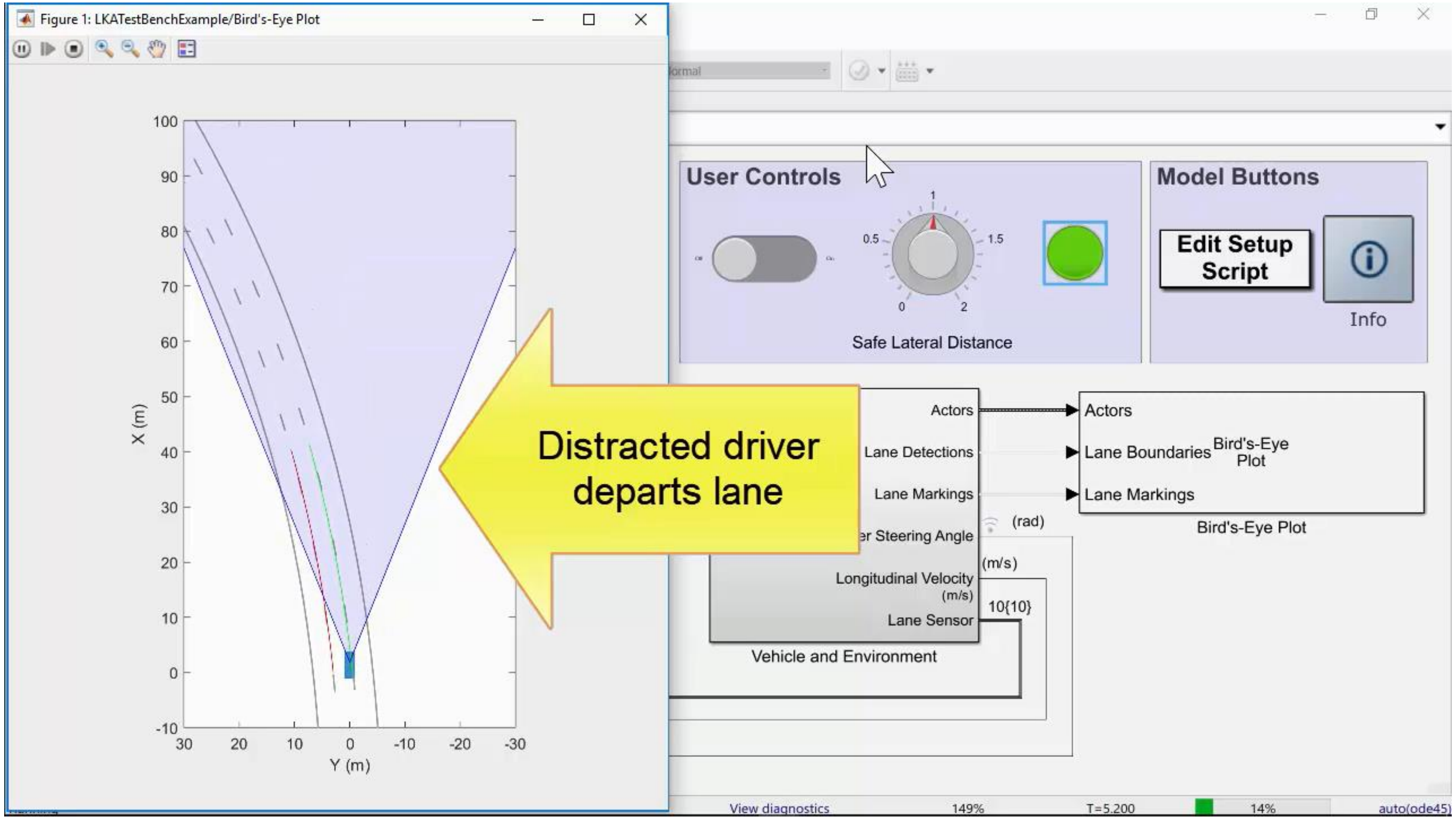
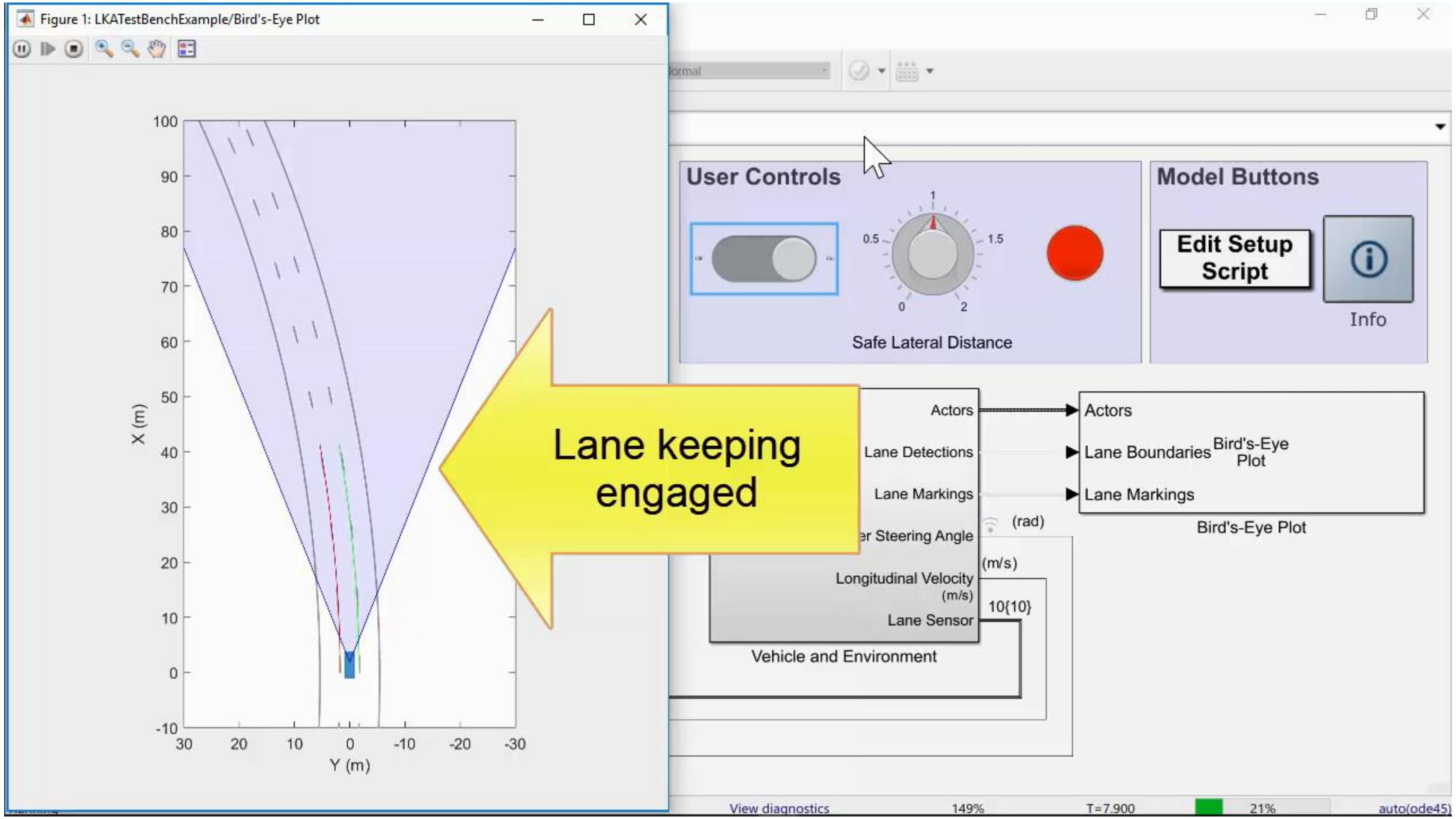


Figure 1: LKATestBenchExample/Bird's-Eye Plot
X (m)
Y (m)
Distracted driver departs lane
User Controls
Safe Lateral Distance
Model Buttons
Edit Setup Script
Info
Actors
Lane Detections
Lane Markings
Longitudinal Velocity (m/s)
Lane Sensor
Vehicle and Environment
Actors
Lane Boundaries
Lane Markings
Bird's-Eye Plot
View diagnostics
T=5.200
auto(ode45)

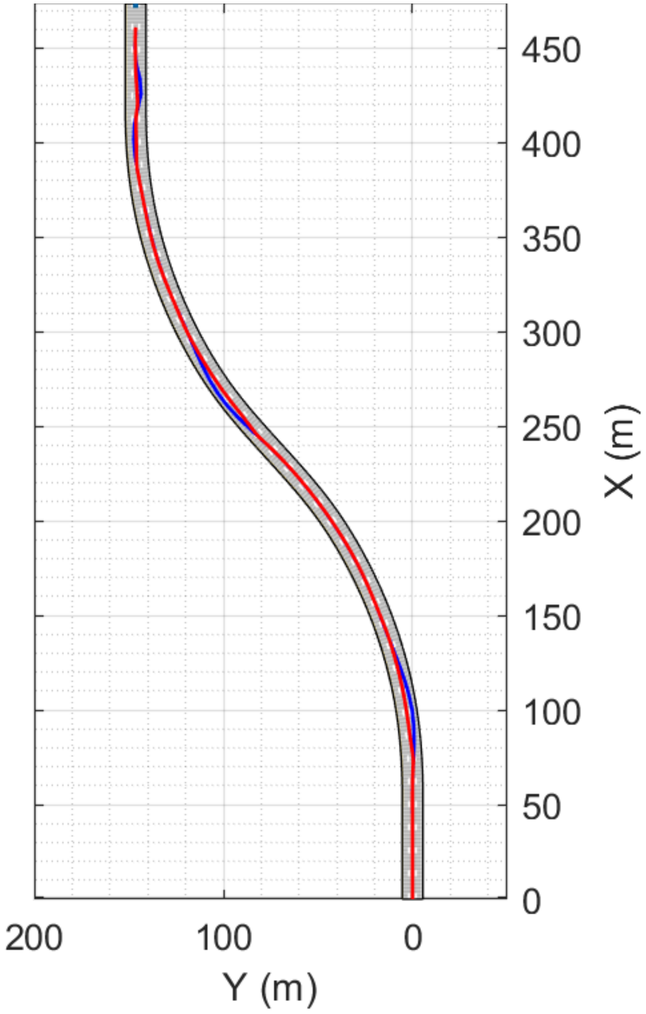
车道保持辅助介入后的仿真结果



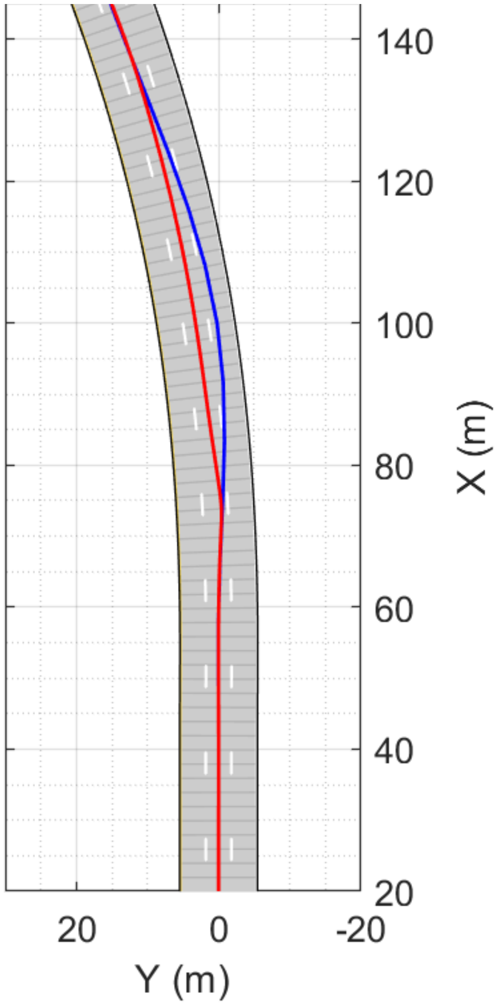
对比分心与辅助后的结果

- 检测车道偏离
- 在驾驶员分心时保持车道

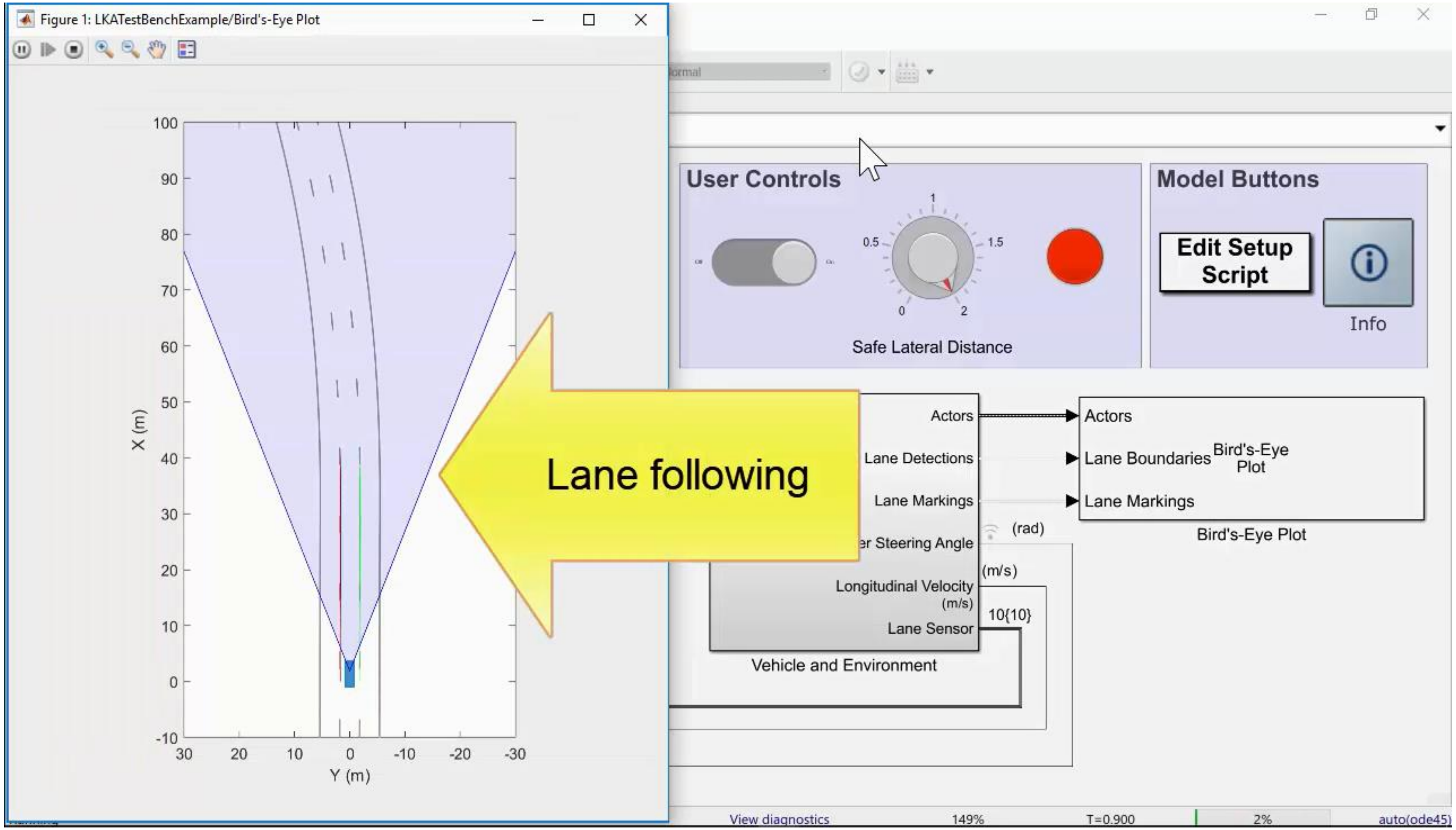
道路与驾驶路径



驾驶员在道路曲率变化处获得辅助



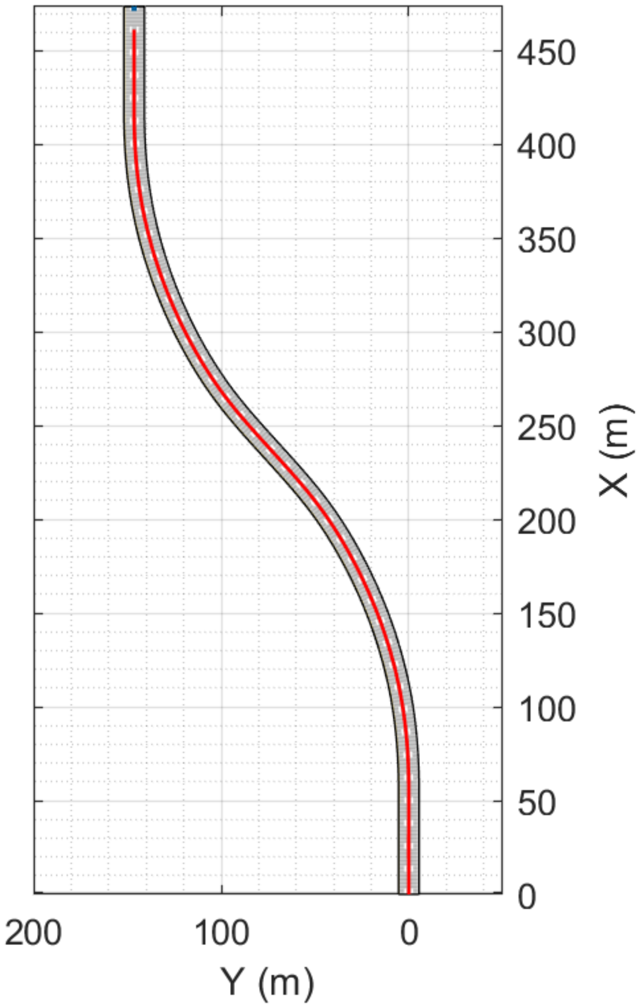
最大化横向安全距离后的车道跟随



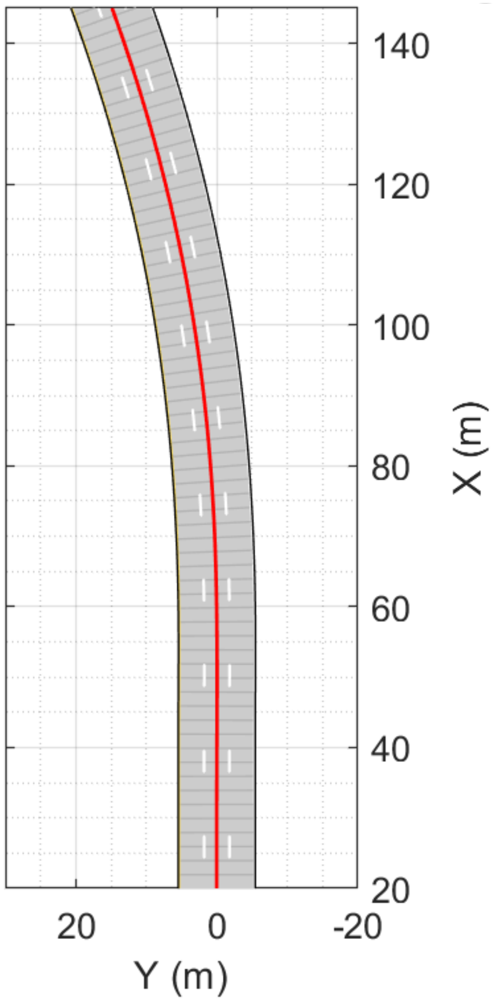
观察车道跟随结果

- 车辆始终保持在车道边界线内

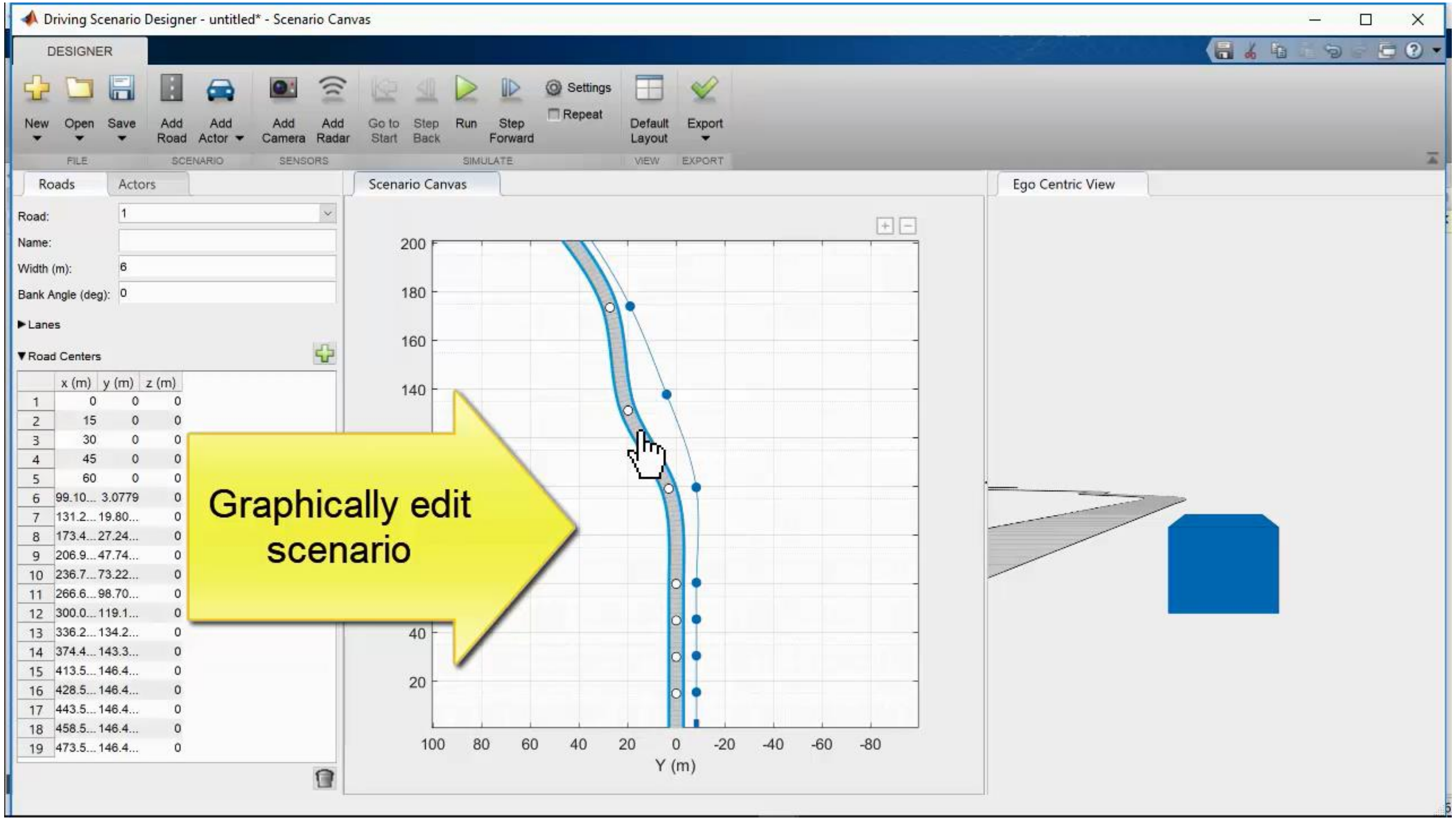
道路与驾驶路径



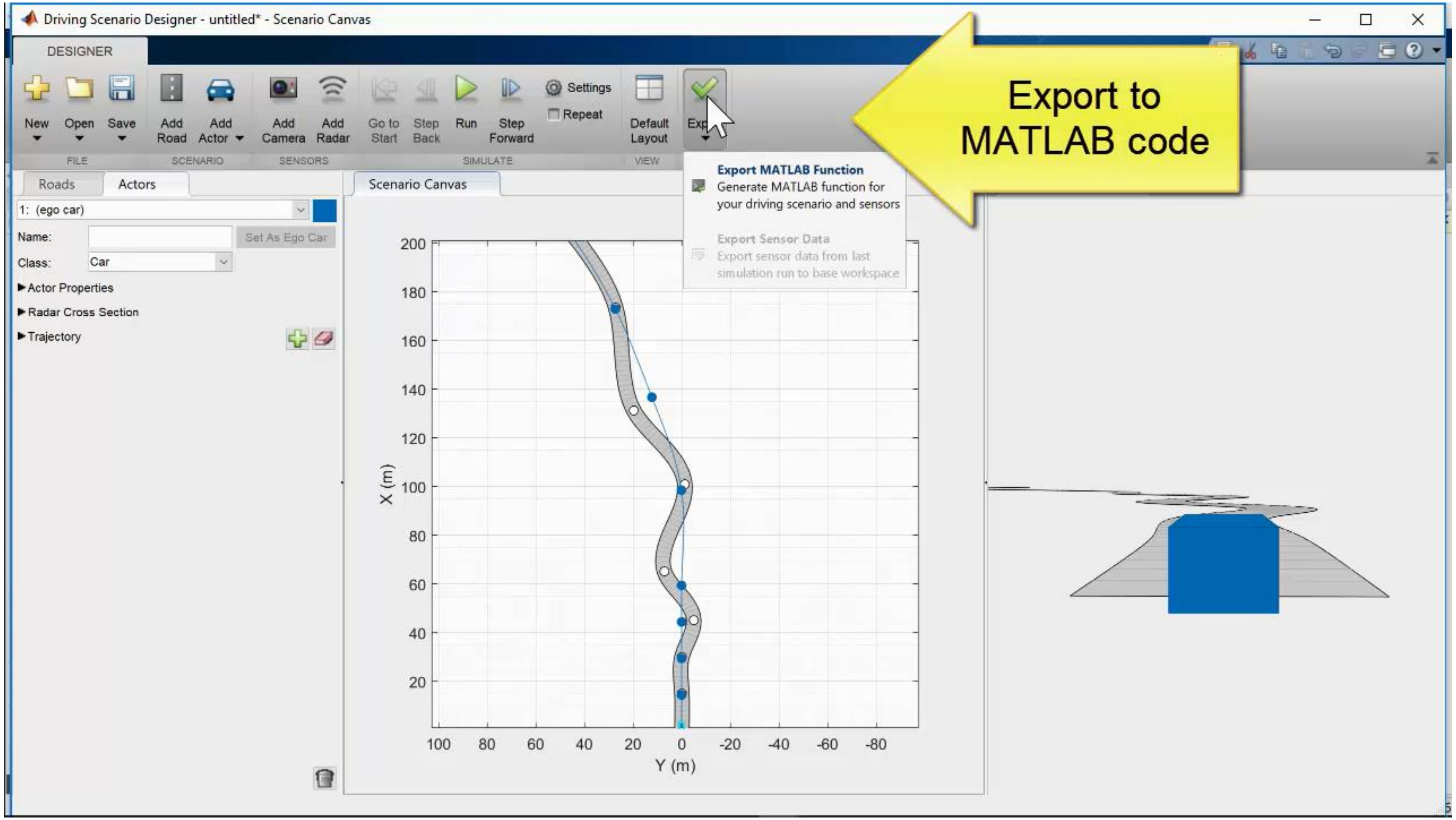
驾驶员在道路曲率变化处获得辅助



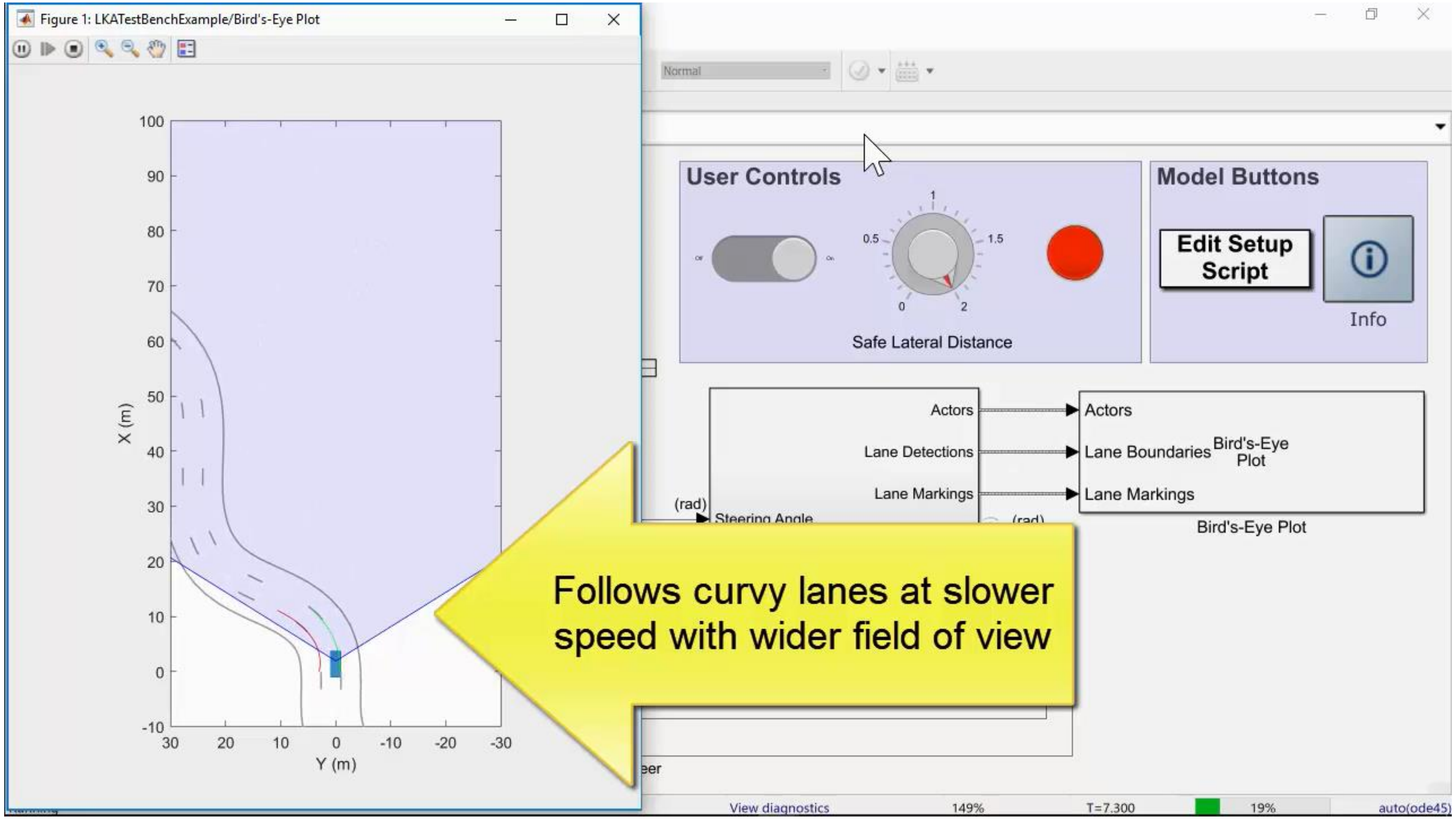
图形化的驾驶场景设计器 （Driving Scenario Designer）



导出用于生成场景的 MATLAB 代码

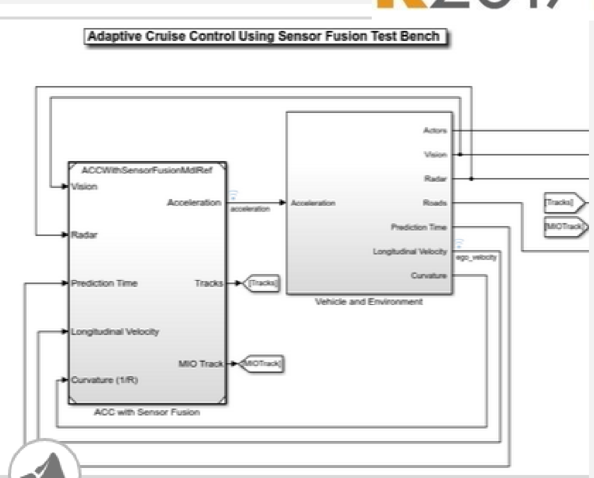


进一步调整参数，适应高曲率道路



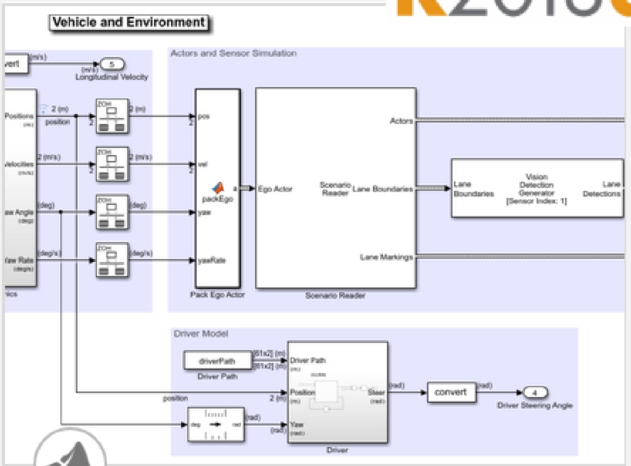
通过这些案例了解如何模拟传感器和驾驶场景，用于开发控制算法

R2017b



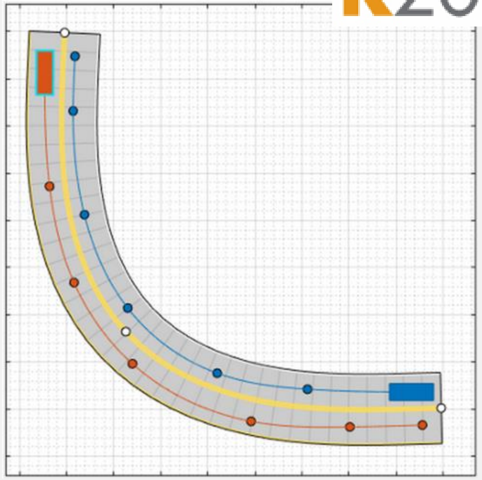
Adaptive Cruise Control with Sensor Fusion

R2018a



Lane Keeping Assist with Lane Detection

R2018a

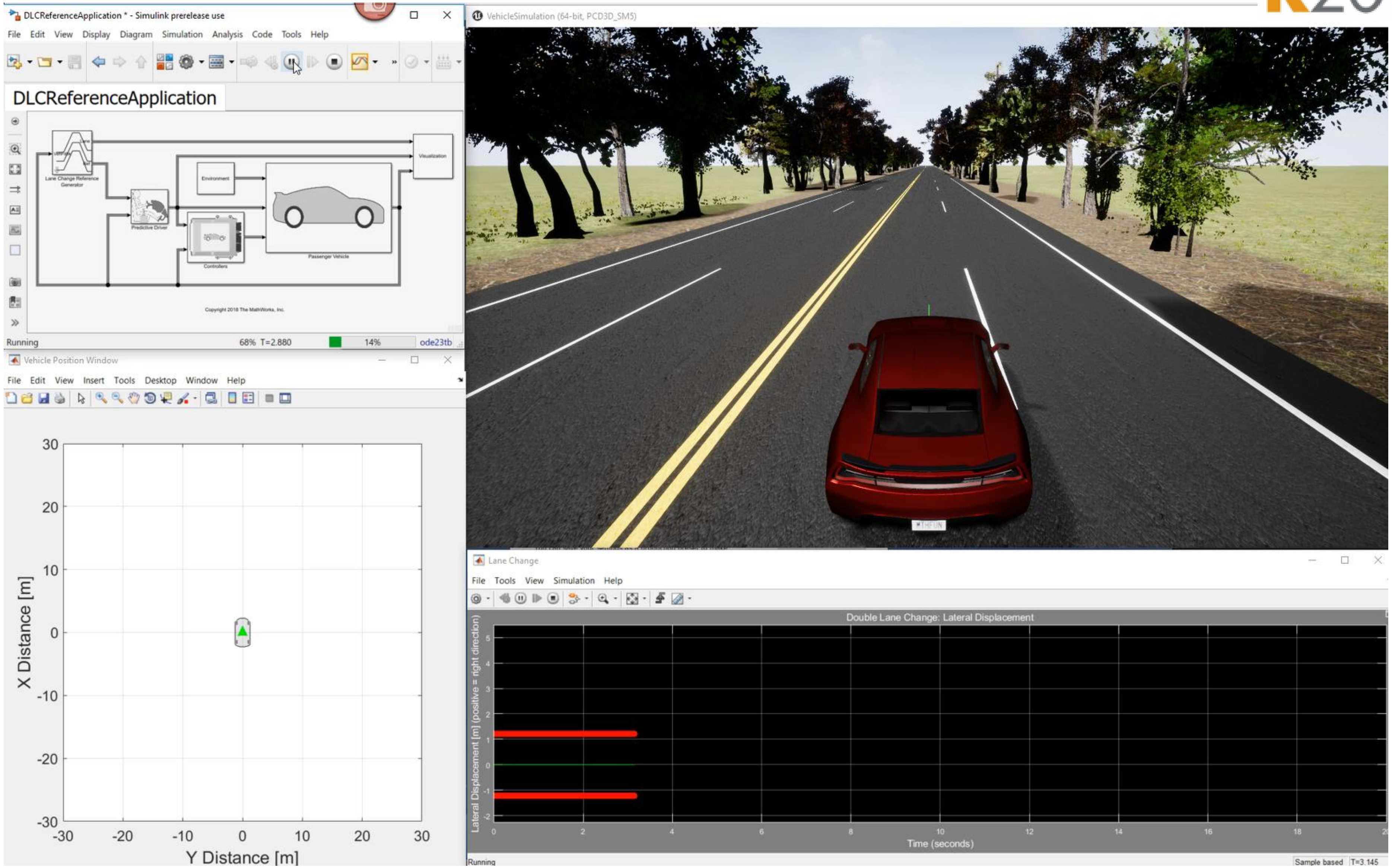


Generate Synthetic Detections from an Interactive Driving Scenario

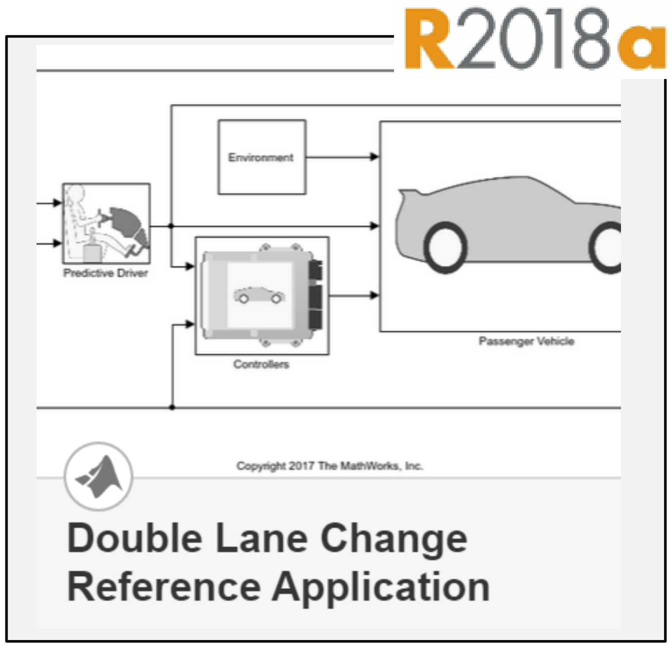
- **仿真与生成 C/C++ 代码**
  模型预测控制与传感器融合算法
- **仿真与生成 C/C++ 代码**
  模型预测控制与含车道线检测的视觉传感器
- **编辑道路、交通参与者及传感器**
  使用驾驶场景设计器
  `drivingScenarioDesigner`

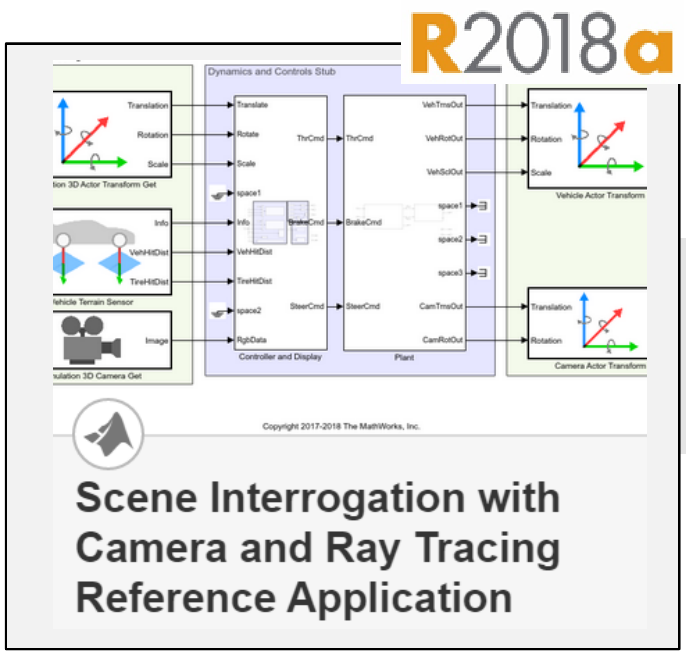
车辆动力学模块库 Vehicle Dynamics Blockset

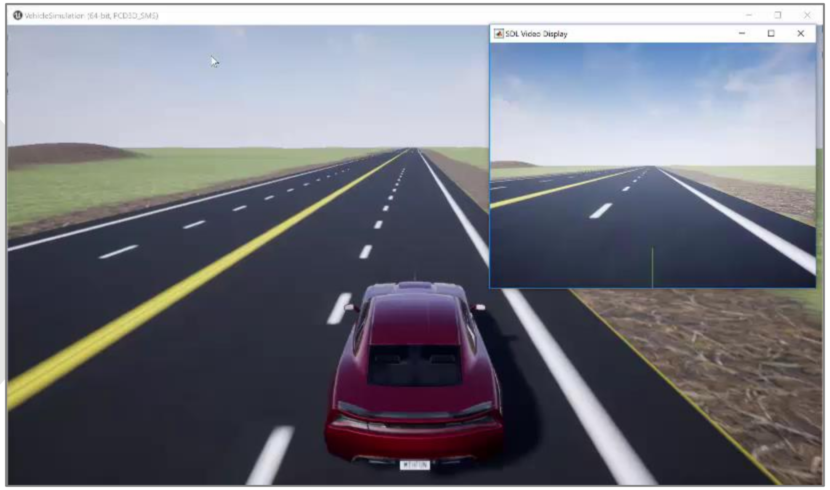
R2018a



通过这些案例了解如何对车辆动力学建模，用于开发控制算法







- **车辆动力学仿真**
  用于闭环控制设计
  Vehicle Dynamics Blockset™
- **与 Unreal 引擎协同仿真**
  设置交通参与者位置及获取摄像机图像
  Vehicle Dynamics Blockset™

使用 MATLAB 和 Simulink 开发自动驾驶算法的案例

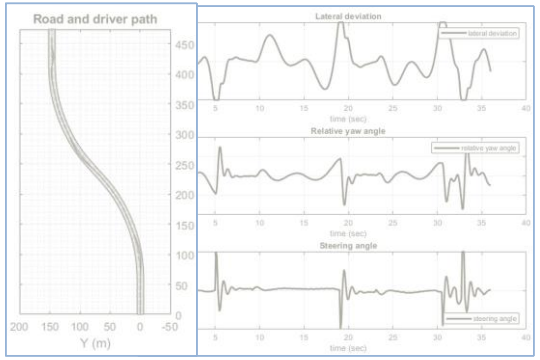
深度学习

对实时数据的
传感器融合

感知
Perception

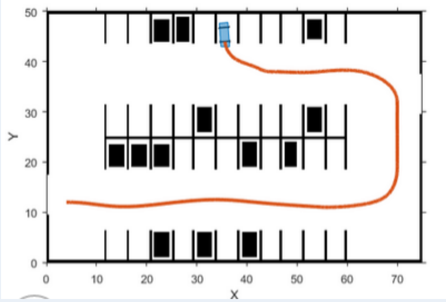
控制
Control

传感器建模 &
模型预测控制



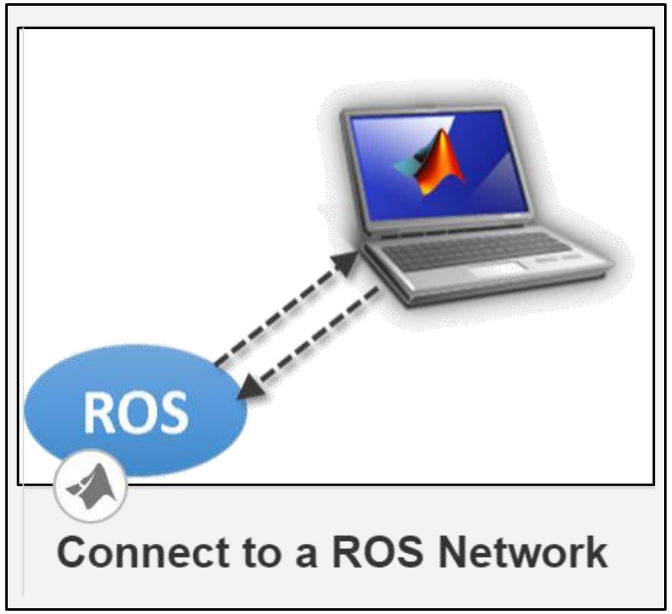
规划 Planning

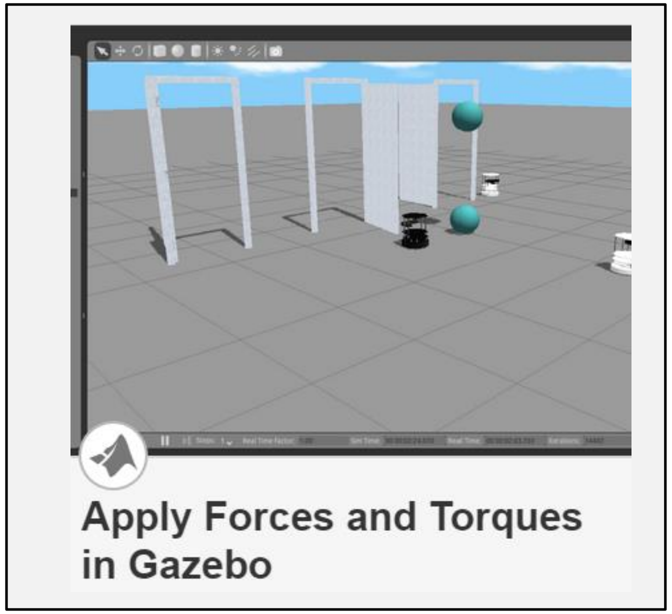
路径规划

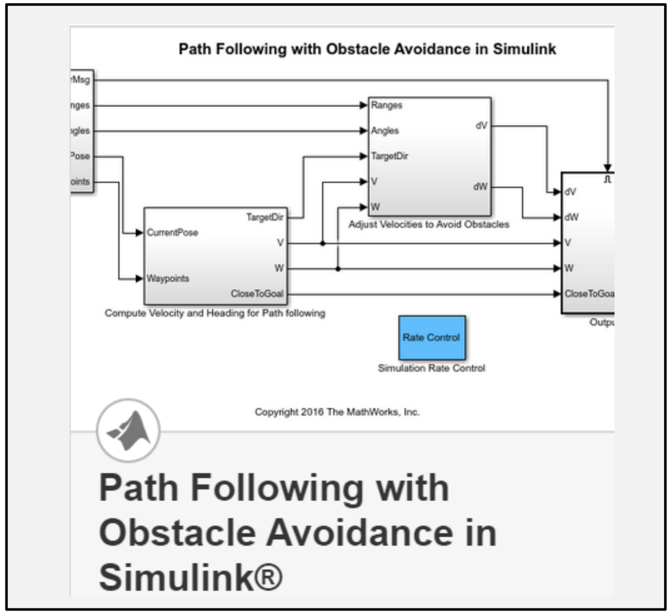


机器人系统工具箱 Robotics System Toolbox

提供与 ROS 生态系统的连接性

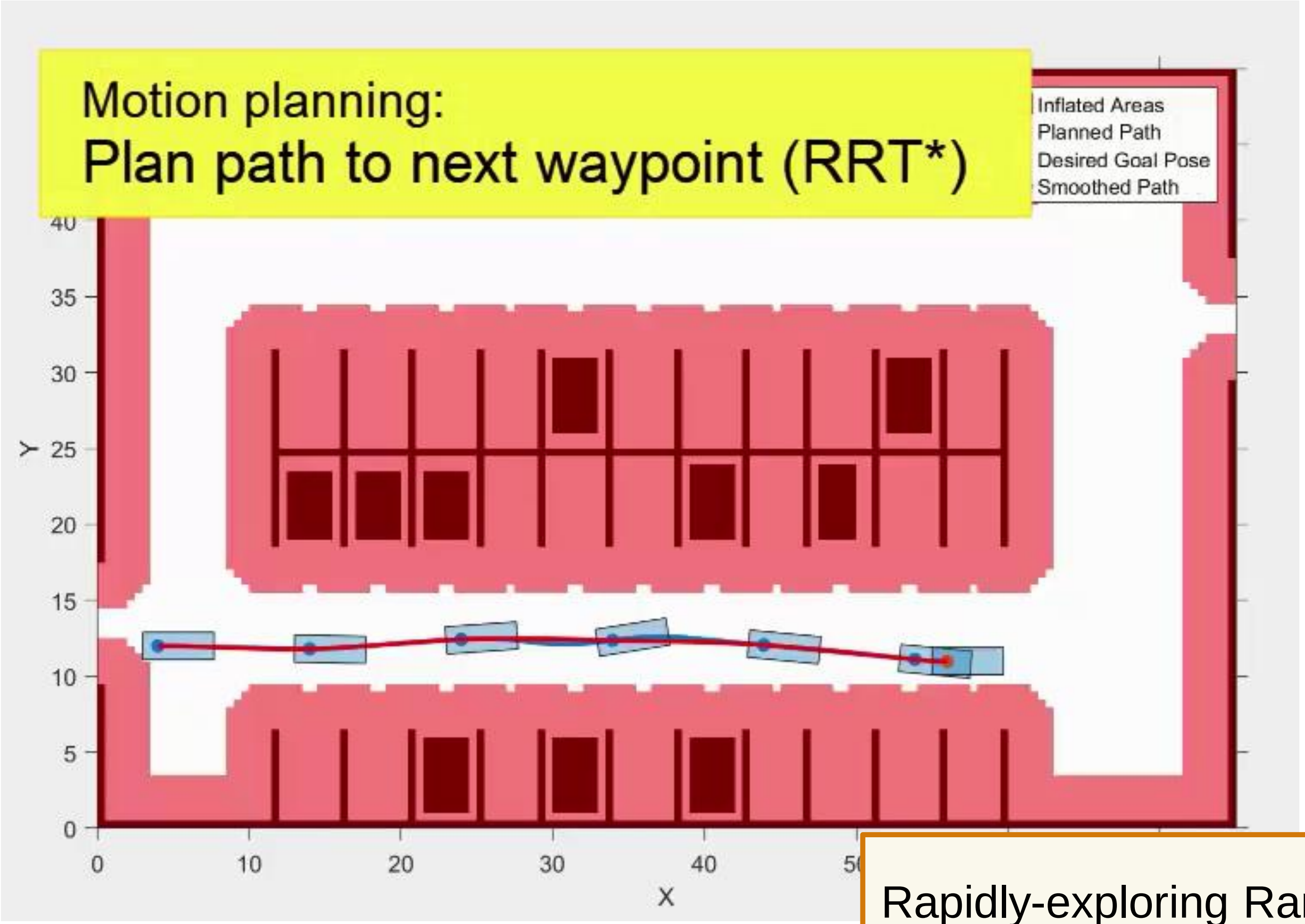






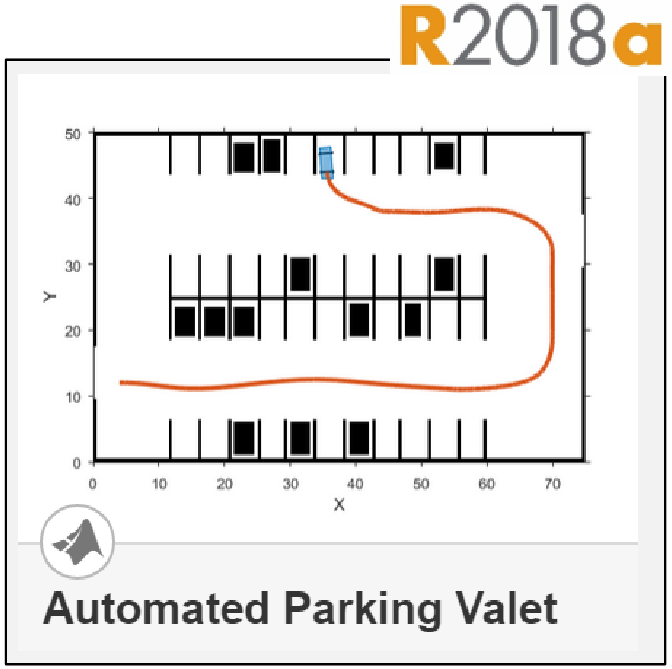
- **使用 ROS 通信**
  与外部的 ROS 组件集成
- **与 Gazebo 通信**
  机器人系统的可视化仿真
- **路径跟随**
  基于 ROS 的差分驱动机器人仿真

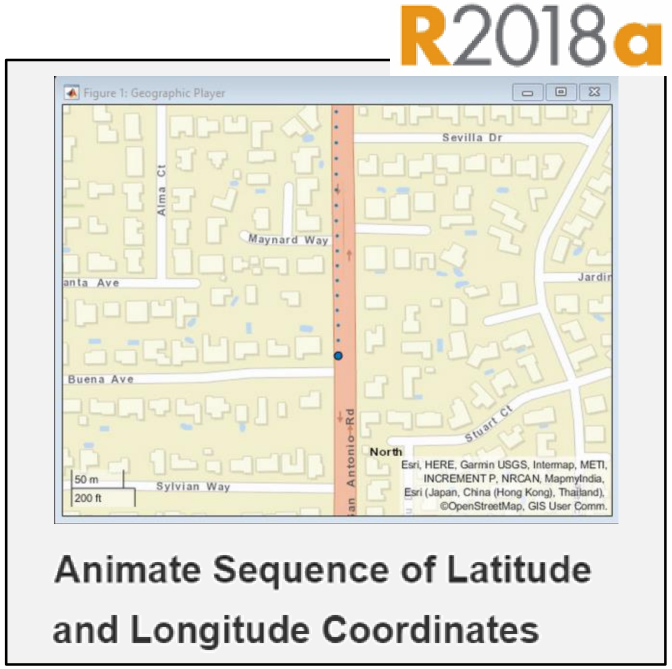
车辆路径规划算法的设计和仿真

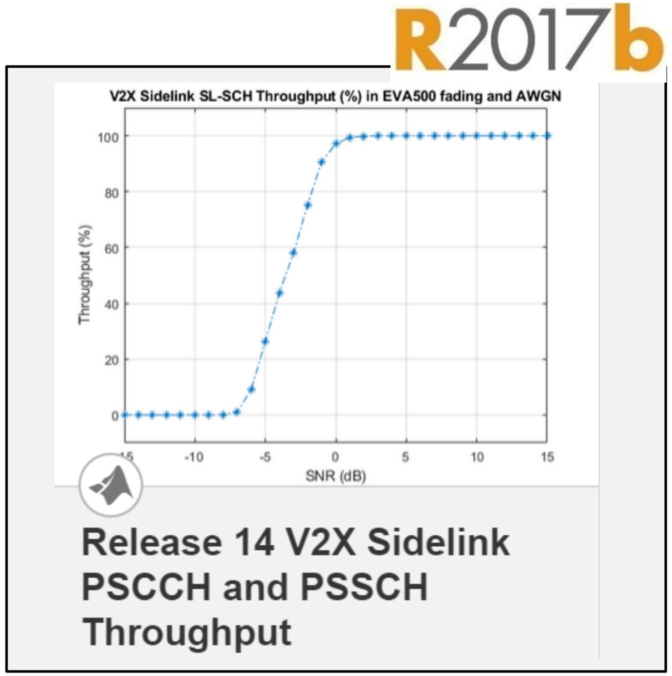


Rapidly-exploring Random Tree (RRT*)

通过这些案例了解如何开发路径规划算法







- **规划车辆路径**
  预定义地图的应用
  Automated Driving System Toolbox™

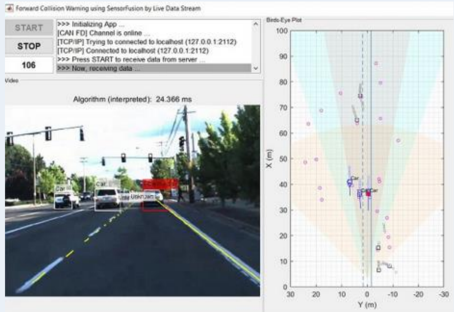
- **绘制地图瓦片**
  使用 World Street Map (ESRI)
  Automated Driving System Toolbox™

- **仿真 V2X 通信**
  评估信道吞吐量
  LTE System Toolbox™

使用 MATLAB 和 Simulink 开发自动驾驶算法的案例

深度学习

对实时数据的
传感器融合

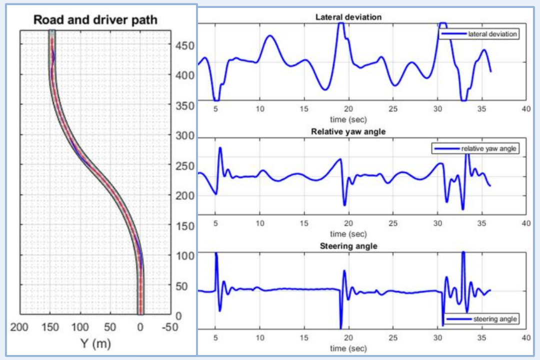


感知
Perception

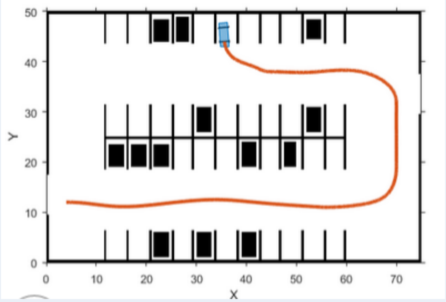
控制
Control

规划 Planning

传感器建模 &
模型预测控制

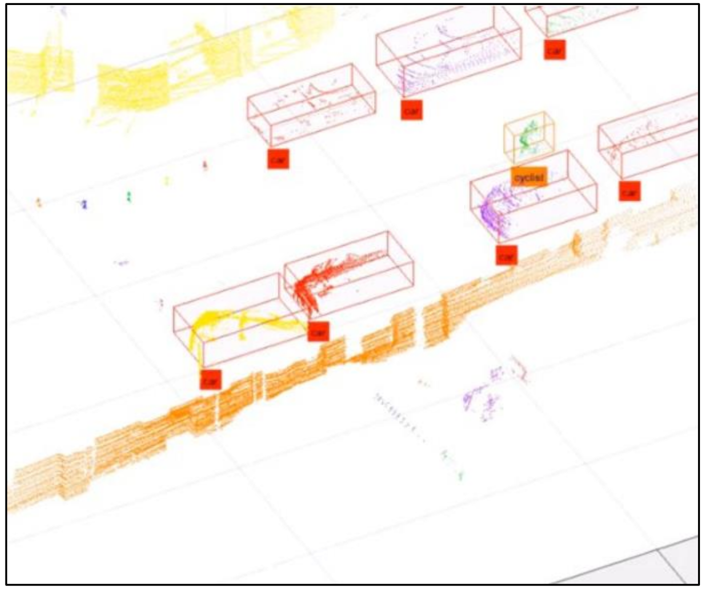


路径规划

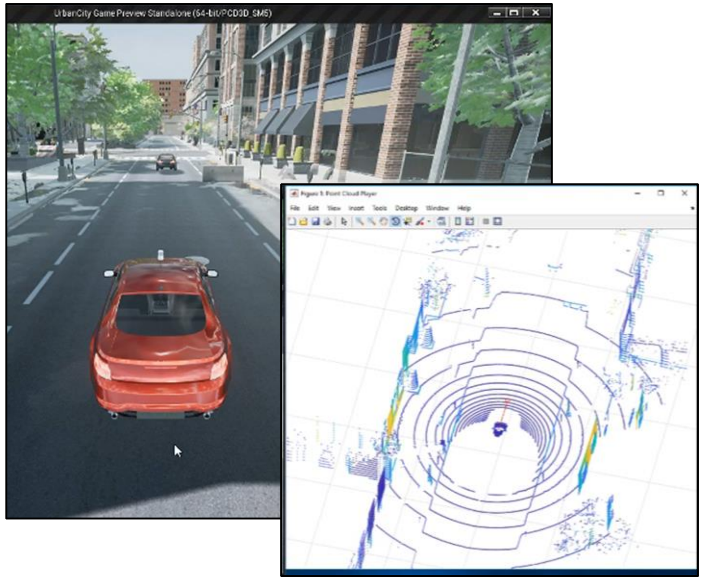


MathWorks 能够帮助您自定义 MATLAB 和 Simulink，建立专门的自动驾驶开发工具

- **基于 Web 的真实值标注**
- 与 Caterpillar 合作的咨询项目
- [2017年 MathWorks 北美汽车年会视频]



- **激光雷达真实值标注**
- 与 Autoliv 联合发表论文（SAE 论文 2018-01-0043）



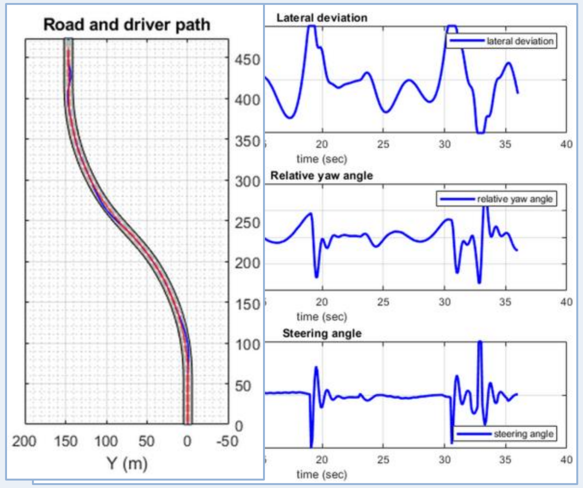
- **用于 Unreal 引擎的激光雷达传感器模型**
- 与 Ford 联合发表论文（SAE 论文 2017-01-0107）

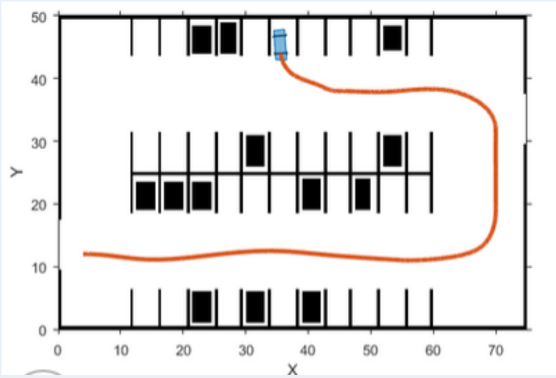
您可以使用 MATLAB 和 Simulink 开发自动驾驶算法 …

感知
Perception

控制
Control

规划 Planning

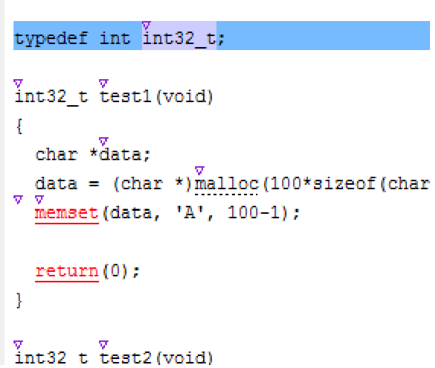




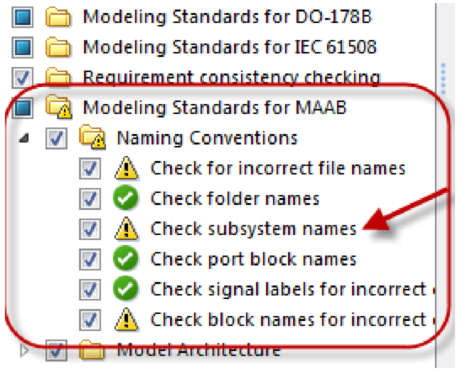
… 同时获得完善的设计验证工具支持

R2017b

Polyspace



Simulink Verification & Validation

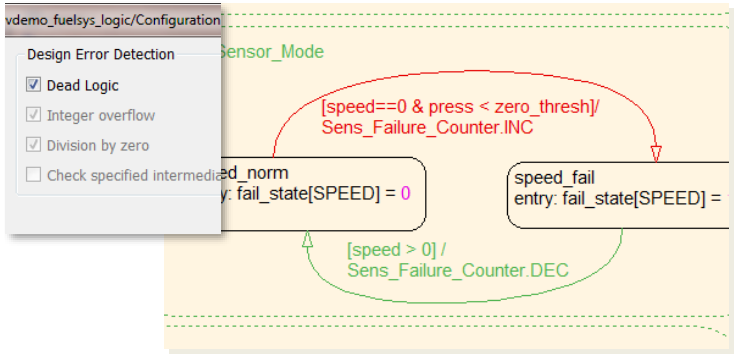


Simulink Requirements

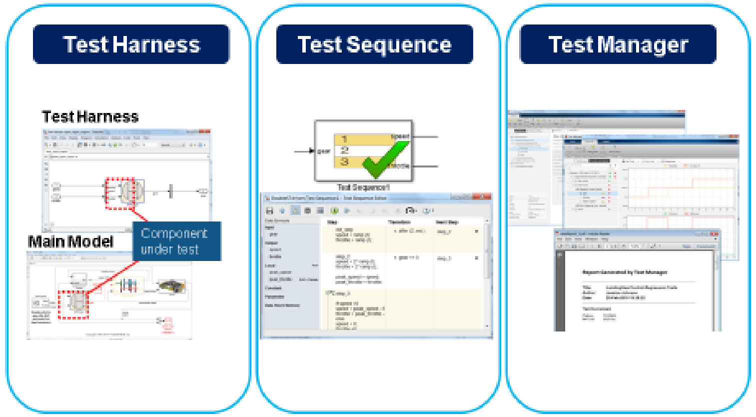
Simulink Coverage

Simulink Check

Simulink Design Verifier



Simulink Test



Q&A