What's New in Automated Driving with MATLAB and Simulink

Mark Corless
Industry Marketing
Automated Driving Segment Manager
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?

Control  Planning  Perception  Simulation Integration

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

Perception
Planning
Control

Simulation Integration
- ROS
- CAN
- C/C++
- Python
- Cross Release
- Third Party
Synthesize scenarios to test sensor fusion algorithms

**Sensor Fusion Using Synthetic Radar and Vision Data**
- Synthesize road and vehicles
- Add probabilistic vision and radar detection sensors
- Fuse and track detections
- Visualize sensor coverage areas, detections, and tracks

*Automated Driving Toolbox™ R2017a*
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
Integrate driving scenarios into Simulink simulations

Test Open-Loop ADAS Algorithm Using Driving Scenario

- Edit driving scenario
- Integrate into Simulink
- Add sensor models
- Visualize results
- Pace simulation

Automated Driving Toolbox™ R2019a
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®*
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®
Visualize sensor detections and tracks

**Lane Following Control with Sensor Fusion**

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

*Model Predictive Control Toolbox™*
*Automated Driving Toolbox™*
*Embedded Coder®*
Automate testing against driving scenarios

Testing a Lane Following Controller with Simulink Test
- Author high level requirements
- Synthesize driving scenarios
- Specify assessment criteria
- Run interactive simulation
- Automate regression testing
- Review verification status

Simulink Test™
Automated Driving Toolbox™
Model Predictive Control Toolbox™

Requirements link
Simulink Model
Define scenario ID and data initialization
Plot the results
Synthesize driving scenarios from recorded data

**Scenario Generation from Recorded Vehicle Data**
- Visualize video
- Import OpenDRIVE roads
- Import GPS
- Import object lists

*Automated Driving Toolbox™ R2019a*
## How can I design with virtual scenarios?

<table>
<thead>
<tr>
<th>Scenes</th>
<th>Driving Scenarios (cuboid)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="#" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Testing</th>
<th>Controls</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Controls + sensor fusion</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authoring</th>
<th>Driving Scenario Designer App</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>drivingScenario programmatic API</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sensing</th>
<th>Probabilistic radar detections</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Probabilistic vision detections</td>
</tr>
<tr>
<td></td>
<td>Probabilistic lane detections</td>
</tr>
</tbody>
</table>
How can I design with virtual scenarios?

<table>
<thead>
<tr>
<th>Scenes</th>
<th>Driving Scenarios (cuboid)</th>
<th>3D Simulation (Unreal Engine)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing</td>
<td>Controls</td>
<td>Controls</td>
</tr>
<tr>
<td></td>
<td>Controls + sensor fusion</td>
<td>Controls + vision</td>
</tr>
<tr>
<td>Authoring</td>
<td>Driving Scenario Designer App</td>
<td>Unreal Editor</td>
</tr>
<tr>
<td></td>
<td>drivingScenario programmatic API</td>
<td></td>
</tr>
<tr>
<td>Sensing</td>
<td>Probabilistic radar detections</td>
<td>Ideal camera (viewer)</td>
</tr>
<tr>
<td></td>
<td>Probabilistic vision detections</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Probabilistic lane detections</td>
<td></td>
</tr>
</tbody>
</table>
Simulate controls and perception systems

Lane Following Control with Sensor Fusion
Model Predictive Control Toolbox™
Automated Driving Toolbox™
Embedded Coder®

Visual Perception Using Monocular Camera
Automated Driving Toolbox™

Lane-Following Control with Monocular Camera Perception
Model Predictive Control Toolbox™
Automated Driving Toolbox™
Vehicle Dynamics Blockset™
Simulate lane controls with vision based perception

**Lane-Following Control with Monocular Camera Perception**

- Integrate Simulink controller
  - Lane follower
  - Spacing control
- Integrate MATLAB perception
  - Lane boundary detector
  - Vehicle detector
- Synthesize ideal camera image from Unreal Engine

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

Perception

Planning

Control

Simulation Integration

ROS

CAN

C/C++

Python

Cross Release

Third Party

Control Planning Perception

Simulation Integration

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

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?
Design trackers

Multi-Object Tracker

Association & Track Management

Tracking Filter

Tracks

Detections

From various sensors at various update rates

- Multi-object tracker

- Linear, extended, and unscented Kalman filters

Automated Driving Toolbox™
Design trackers

From various sensors at various update rates

- 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, and Interacting Multiple Model (IMM) filters

Automated Driving Toolbox™
Sensor Fusion and Tracking Toolbox™
Design multi-object trackers

**Extended Object Tracking**
- Design multi-object tracker
- Design extended object trackers
- Evaluate tracking metrics
- Evaluate error metrics
- Evaluate desktop execution time

**Sensor Fusion and Tracking Toolbox™**

**Automated Driving Toolbox™**

Updated **R2019a**
Design extended object trackers

**Extended Object Tracking**
- Design multi-object tracker
- Design extended object trackers
- Evaluate tracking metrics
- Evaluate error metrics
- Evaluate desktop execution time

*Sensor Fusion and Tracking Toolbox™*

*Automated Driving Toolbox™*

Updated **R2019a**
Evaluate tracking performance

**Extended Object Tracking**
- Design multi-object tracker
- Design extended object trackers
- Evaluate tracking metrics
- Evaluate error metrics
- Evaluate desktop execution time

**Sensor Fusion and Tracking Toolbox™**

**Automated Driving Toolbox™**

*Updated 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
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 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™**

MathWorks 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
Generate C/C++ code for lidar detector and tracker

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™
Create region of interest labels and groups

Get Started with the Ground Truth Labeler
- Label rectangles
- Label lane markings
- Label pixels
- Label scenes
- Create label groups
- Create sublabels
- Add label attributes

Automated Driving Toolbox™
Updated R2019a
Create sublabels and add attributes

Get Started with the Ground Truth Labeler
- Label rectangles
- Label lane markings
- Label pixels
- Label scenes
- Create label groups
- Create sublabels
- Add label attributes

Automated Driving Toolbox™
Updated R2019a
Create polyline labels and add attributes

Get Started with the Ground Truth Labeler
- Label rectangles
- Label lane markings
- Label pixels
- Label scenes
- Create label groups
- Create sublabels
- Add label attributes

Automated Driving Toolbox™
Updated R2019a
Create pixel labels

Get Started with the Ground Truth Labeler
- Label rectangles
- Label lane markings
- Label pixels
- Label scenes
- Create label groups
- Create sublabels
- Add label attributes

Automated Driving Toolbox™
Updated R2019a
Create scene labels and groups

Get Started with the Ground Truth Labeler
- Label rectangles
- Label lane markings
- Label pixels
- Label scenes
- Create label groups
- Create sublabels
- Add label attributes

Automated Driving Toolbox™
Updated R2019a
Import custom automation algorithms

Automate Attributes of Labeled Objects

- Import automation algorithm into Ground Truth Labeling app
- Detect vehicles from monocular camera
- Estimate distance to detected vehicles
- Run automation algorithm and interactively validate labels

Automated Driving Toolbox™

R2018b
Add custom visualizations for multi-sensor data

Connect Lidar Display to Ground Truth Labeler
- Sync external tool to each frame change
- Control external tool through playback controls

Automated Driving Toolbox™

R2017a
Interoperate with neural network frameworks

Open Neural Network Exchange
Design camera, lidar, and radar perception algorithms

**Detect vehicle with camera**

**Detect ground with lidar**

**Detect pedestrian with radar**

*Object Detection Using YOLO v2 Deep Learning*

*Computer Vision Toolbox™*

*Deep Learning Toolbox™*

*Segment Ground Points from Organized Lidar Data*

*Computer Vision Toolbox™*

*Introduction to Micro-Doppler Effects*

*Phased Array System Toolbox™*
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?

Perception

Planning

Control

Simulation Integration

ROS

CAN

C/C++

Python

Cross Release

Third Party
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*
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

**Automated Driving Toolbox™**

R2018b
Generate C/C++ code for path planner and controller

**Code Generation for Path Planning and Vehicle Control**
- Simulate system
- Configure for code generation
- Generate C/C++ code
- Test using Software-In-the-Loop
- Measure execution time of generated code

*Automated Driving Toolbox™*
Embedded Coder

R2019a
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?

Perception

Planning

Control

Simulation Integration

ROS

C/C++

Python

CAN

Cross Release

Third Party
Design lateral and longitudinal Model Predictive Controllers

**Longitudinal Control**

- Adaptive Cruise Control with Sensor Fusion
- Automated Driving Toolbox™
- Model Predictive Control Toolbox™
- Embedded Coder®

**Lateral Control**

- Lane Keeping Assist with Lane Detection
- Automated Driving Toolbox™
- Model Predictive Control Toolbox™
- Embedded Coder®

**Longitudinal + Lateral**

- Lane Following Control with Sensor Fusion and Lane Detection
- Automated Driving Toolbox™
- Model Predictive Control Toolbox™
- Embedded Coder®
Develop automatic emergency braking application

**Automatic Emergency Braking (AEB) with Sensor Fusion**
- Specify driving scenario
- Design AEB logic
- Integrate sensor fusion
- Visualize sensors and tracks
- Generate C/C++ code
- Test with software in the loop (SIL) simulation

Automated Driving Toolbox™
Stateflow®
Embedded Coder®

R2018b
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
Some common questions from automated driving engineers

- How can I synthesize scenarios to test my designs?
- How can I discover and design in new domains?
- How can I integrate with other environments?
Integrate with ROS

Replay logged ROS data

Exchange Data with ROS Publishers and Subscribers

Generate a Standalone ROS Node from Simulink

Work with rosbag Logfiles

Robotic System Toolbox™

Updated R2016b

Robotic System Toolbox™

Simulink Coder™

Updated R2018a

Generate standalone ROS node
Call C++, Python, and OpenCV from MATLAB

**Call C++**

- **.hpp**
- **.mlx**

**Call Python**

```python
import pytextwrap.TextWrapper
pytextwrap.TextWrapper(...
  'initial_indent', '% ','subsequent_indent', '% ','width', int32(30))
```

**Call OpenCV & OpenCV GPU**

- **cv::Rect**
- **cv::KeyPoint**
- **cv::Size**
- **cv::Mat**
- **cv::Ptr**

**Import C++ Library Functionality into MATLAB**

MATLAB®

**Call Python from MATLAB**

MATLAB®

**Install and Use Computer Vision Toolbox OpenCV Interface**

Computer Vision System Toolbox™

OpenCV Interface Support Package

**R2019a**

**R2014a**

**Updated R2018b**
Call C code from Simulink

- **Call C code**
- **Create buses from C structs**
  ```c
  typedef struct {
    double coeff;
    double init;
    fault_t fault;
  } params_t;
  ```
- **Test and verify C code**

**Bring Custom Image Filter Algorithms as Reusable Blocks in Simulink**

- **Import Structure and Enumerated Types**
- **Custom C Code Verification with Simulink Test**
  - Simulink Test™
  - Simulink Coverage™

**R2017b**

**R2017a**

**R2019a**
Connect to third party tools

152 Interfaces to 3rd Party Modeling and Simulation Tools
(as of March 2019)
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
Some common questions from automated driving engineers

Synthesize scenarios to test my designs

Discover and design in multiple domains

Integrate with other environments
Get started on your own with documented examples

- **Automated Driving Toolbox**
  - Labeling, perception, sensor fusion, path planning, synthetic sensor data

- **Model Predictive Control Toolbox** *(Section: Automated Driving Applications)*
  - Adaptive cruise control, lane keeping, lane following with spacing control

- **Simulink Test** *(Section: Systematic Testing and Reporting)*
  - Test lane following controller with sensor fusion
Gain tool experience with Training Services

MATLAB and Simulink Training

- Automated Driving with MATLAB
- Deep Learning with MATLAB
- Computer Vision with MATLAB
- Simulink for System and Algorithm Modeling
- Integrating Code with Simulink
- Code Generation for AUTOSAR Software
- Verification and Validation of Simulink Models
- Polyspace Bug Finder for C/C++ Code Analysis
- Ask about customizing training courses for your needs (contact training)
Partner on your projects with Consulting Services

MATLAB and Simulink Consulting Services

- Image Processing and Computer Vision
- MATLAB with Hadoop and Spark
- Tools Integration
- ISO 26262 Process Deployment Advisory Service
- Model-Based Design Process Establishment
- Model-Based Design Process Assessment and Maturity Framework
- Ask about extending tools for labeling or synthesizing sensor data (contact consulting)
Get started developing automated driving systems with MATLAB and Simulink

Discuss your application with me (mark.corless@mathworks.com) or a MathWorks field engineer to help you structure an evaluation

- Understand your goals
- Recommend tasks
- Answer questions