MATLAB EXPO 2019
Automated Driving System Design and Simulation

Dr. Amod Anandkumar
MathWorks India
Capabilities of an Autonomous Vehicle
Capabilities of an Autonomous Vehicle
Capabilities of an Autonomous Vehicle

1. Sense
2. Perceive
3. Decide 
4. Plan
Capabilities of an Autonomous Vehicle

Sense
Perceive
Decide & Plan
Act
Evolution of ADAS and Autonomous Driving Car Technologies
Reference examples using Automated Driving Toolbox™

- **L5** Full Automation
  - Self-Driving Car
- **L4** High Automation
  - Self-Driving & Human-Driven Car
- **L3** Conditional Automation
- **L2** Partial Automation
- **L1** Driver Assistance
- **L0** No Automation

- **2010**
  - ACC
  - Lane Keep Assist/Lateral Support
- **2014**
  - FCW
- **2016**
  - AEB-Vehicle (City/Inter-Urban)
  - Lane Keep Assist/Lateral Support
  - AEB-VRU (Cyclist)
  - AEB-VRU (Pedestrian)
  - Junction Assist
- **2018a**
  - Auto Pilot: Highway
  - Auto Pilot: Parking
  - Auto Pilot: Road Train
  - Lane Following
  - R2018b
- **2018b**
  - Lane Following

- **2020**
  - R2017b

- **2025**

- **2030**
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?

Simulation Integration

Perception

Planning

Control

ROS
C/C++
Python
CAN
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?

Control
Planning
Perception

Simulation Integration
- ROS
- CAN
- C/C++
- Python
- Cross Release
- Third Party

Cross
Release
Third
Party
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
Simulate driving scenarios into closed loop simulations

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

**Automated Driving Toolbox™**

**Stateflow®**

**Embedded Coder®**

---

*Develop and Test Vehicle Controllers for ADAS and Automated Driving Applications Through System Simulation*

15:00–15:30
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
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><strong>Driving Scenarios (cuboid)</strong></th>
</tr>
</thead>
</table>
| Testing | Controls  
| | Controls + sensor fusion  |
| Authoring | Driving Scenario Designer App  
| | drivingScenario programmatic API  |
| Sensing | Probabilistic radar detections  
| | Probabilistic vision detections  
| | Probabilistic lane detections  |
### How can I design with virtual scenarios?

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

### Testing
- Controls
- Controls + sensor fusion
- Controls + vision

### Authoring
- Driving Scenario Designer App
- drivingScenario programmatic API
- Unreal Editor

### Sensing
- Probabilistic radar detections
- Probabilistic vision detections
- Probabilistic lane detections
- Ideal camera (viewer)
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?

Control
Planning
Perception

Simulation Integration

ROS
CAN
C/C++
Python
Cross Release
Third Party
Interactively label sensor data

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
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
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™
Design multi-object trackers

Multi-Object Tracker

Association & Track Management

Tracking Filter

From various sensors at various update rates

- 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

Sensor Fusion and Tracking Toolbox™
Automated Driving 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

![Graphs showing position and velocity errors for different trackers](image)

- **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**
Design tracker for lidar point cloud data

Track Vehicles Using Lidar: From Point Cloud to Track List
- Design 3-D bounding box detector
- Design JPDA tracker (target state and measurement models)
- Generate C/C++ code for detector and tracker

Sensor Fusion and Tracking Toolbox™
Computer Vision Toolbox™

LiDAR Processing for Automated Driving
12:45–13:15
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?
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

```c
186 // model step function
187 void step0();
189
190 // model step function
191 void step1();
192
193 // model terminate function
194 void terminate();
195
196 // Constructor
197 AutomatedParkingValetModelClass();
198
199 // Destructor
200 ~AutomatedParkingValetModelClass();
201
202 // Root import: '<Root>/Costmap' set method
203 void setCostmap(costmapBus localArgInput);
204
205 // Root import: '<Root>/GoalPose' set method
206 void setGoalPose(real_T localArgInput[3]);
```
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 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 and Test Vehicle Controllers for ADAS and Automated Driving Applications Through System Simulation**

15:00–15:30
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™

Deep Learning and Reinforcement Learning Workflows in AI
16:15–16:45
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

Connect to live ROS data

Generate standalone ROS node

Work with rosbag Logfiles
Robotic System Toolbox™

Exchange Data with ROS Publishers and Subscribers
Robotic System Toolbox™

Generate a Standalone ROS Node from Simulink
Robotic System Toolbox™
Simulink Coder™
Call C++, Python, and OpenCV from MATLAB

Call C++

Call Python

Call OpenCV & OpenCV GPU

Import C++ Library Functionality into MATLAB

MATLAB®

R2019a

Call Python from MATLAB

MATLAB®

R2014a

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
- Create buses from C structs
- Test and verify C code

- src mean_filter dst
  - C Caller

Bring Custom Image Filter Algorithms as Reusable Blocks in Simulink

Simulink®

R2017b

Import Structure and Enumerated Types

Simulink®

R2017a

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)
Some common questions from automated driving engineers

- Synthesize scenarios to test my designs
- Discover and design in multiple domains
- Integrate with other environments

Perception

Planning

Control

Simulation Integration

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

ROS

C/C++

Python

Cross Release

Third Party
MathWorks can help you customize MATLAB and Simulink for your automated driving application

Voyage develops MPC controller and integrates with ROS
- Developed & deployed in 3 months
- 2018 MathWorks Automotive Conference

Autoliv labels ground truth lidar data
- > 4x reduction in labeling effort
- Joint paper in SAE (2018-01-0043)
- 2018 MathWorks Automotive Conference

Ford synthesizes lidar data to test autonomous driving & active safety systems
- Joint paper with Ford
- SAE Paper 2017-01-0107
Automated Driving with MATLAB
This one-day course provides hands-on experience with developing and verifying automated driving perception algorithms

Topics include:
- Labeling of ground truth data
- Visualizing sensor data
- Detecting lanes and vehicles
- Fusing sensor detections
- Generating driving scenarios and modeling sensors
Develop Automated Driving Systems with MATLAB and Simulink

Simulation Integration

- Perception
- Planning
- Control

Discuss your application with a MathWorks field engineer to help you structure your evaluation
- Understand your goals
- Recommend tasks
- Answer questions

Visit us at demo booths
- Automated Driving
- Deep Learning