MATLAB EXPO 2019

MATLAB® 및 Simulink® 를 이용한 자율주행 시스템 설계 및 시뮬레이션

김종헌



Develop Automated Driving Control Systems with MATLAB and Simulink







Develop Automated Driving Perception Systems with MATLAB and Simulink







Develop Automated Driving Planning Systems with MATLAB and Simulink







Develop Automated Driving Systems with MATLAB and Simulink







Develop Automated Driving Control Systems with MATLAB and Simulink



Some common control tasks

- Connect to recorded and live CAN data
- Synthesize scenarios and sensor detections
- Model vehicle dynamics
- Design model-predictive controllers
- <u>Design reinforcement learning networks</u>
- Automate regression testing
- Prototype on real-time hardware
- Generate production C/C++ code
- Certify for ISO26262



Synthesize Driving Scenarios to Test Sensor Fusion Algorithms

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



📣 MathWorks[,]

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[™] Driving Scenario Designer Example

Synthesize Driving Scenarios from Recorded Data

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

Automated Driving ToolboxTM

Scenario Generation from Recorded Vehicle Data Example

R2018b Integrate 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 ToolboxTM Stateflow[®] Embedded Coder[®]

Automatic Emergency Braking (AEB) with Sensor Fusion

Design Lateral and Longitudinal Model Predictive Controllers

Longitudinal Control

Lateral Control

Adaptive Cruise Control with Sensor Fusion

Automated Driving ToolboxTM Model Predictive Control ToolboxTM Embedded Coder[®]

R2017b

Lane Keeping Assist with Lane Detection Automated Driving ToolboxTM Model Predictive Control ToolboxTM Embedded Coder[®] R2018a

Longitudinal + Lateral

Lane Following Control with Sensor Fusion and Lane Detection Automated Driving ToolboxTM Model Predictive Control ToolboxTM Embedded Coder[®] R2018b

📣 MathWorks

R2018b

Automate simulation tests

- Testing a lane following controller with Simulink Test
 - Specify driving scenario
 - Design AEB logic
 - Integrate sensor fusion
 - Simulate system
 - Generate C/C++ code
 - Test with Software-In-the-Loop (SIL) simulation

Simulink Test[™]

Testing a Lane Following Controller with Simulink Test

MathWorks[.]

Automate simulation tests

Develop Automated Driving Perception Systems with MATLAB and Simulink

Some common control tasks

- Visualize images, detections, and point clouds
- Label sensor data
- Synthesize scenarios and sensors
- Design fusion and tracking algorithms
- Design vision algorithms
- Design lidar algorithms
- Generate C/C++ code
- Design deep learning networks
- Generate GPU code

Develop Automated Driving Perception Systems with MATLAB and Simulink

MathWorks[®] R2017a

Interactively Label Sensor Data

- Get started with the Ground Truth Labeler
 - Label scenes
 - Label regions of interest
 - Label lanes

Automated Driving ToolboxTM

Get Started with the Ground Truth Labeler Example

Interactively Label Sensor Data

- Get started with the
 Ground Truth Labeler
 - Label scenes
 - Label regions of interest
 - Label lanes
 - Label pixels
 - Add label attributes
 - Create sub-labels
 - Group labels

Automated Driving ToolboxTM

Get Started with the Ground Truth Labeler Example

R2018b

Automate Attributes of Labeled Objects

- Detect vehicles from monocular camera
 - Temporal Interpolation
 - ACF Detector
 - Optical Flow
 - Import automation algorithm into Ground Truth Labeling App
- Run automation algorithm and interactively validate labels

Automated Driving ToolboxTM Automate Attributes of Labeled Objects Example

Evaluate and Compare Metrics for Fusion and Tracking Algorithms

- Design multi-object trackers
 - GNN + Kalman Filter (КF, ЕКF, UKF)
 R2017а
 - MHT, IMM, JPDA **R2018b**
- Evaluate tracking metrics
- Evaluate desktop execution time R2019a

Sensor Fusion and Tracking ToolboxTM Automated Driving ToolboxTM

- 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 ToolboxTM Computer Vision ToolboxTM

Track Vehicles Using Lidar: From Point Cloud to Track List Example

- 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 ToolboxTM Computer Vision ToolboxTM **Track Vehicles Using Lidar: From Point Cloud to 1**

MATLAB EXPO 2019

MathWorks^{*}

R2019a

- 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 ToolboxTM Computer Vision ToolboxTM

Track Vehicles Using Lidar: From Point Cloud to Track List Example

- 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 ToolboxTM Computer Vision ToolboxTM

Track Vehicles Using Lidar: From Point Cloud to Track List Example

A MATLAB R20	96	σ×				
HOME	PLOTS APPS EDITOR PLOLISH VEW	🤹 Sign In				
Area Copen 1	Image: State of Files Image: State of Files <td< td=""><td>ž</td></td<>	ž				
	2 - C + work + userpath + Examples + R2019a + driving_fusion_vision + TrackVehiclesUsingLidarExample	• A				
TrackVehicl	esUsingLiderExample.m × +					
134	Interacting Multiple Model	~				
135	% IMM filter initialization function (INAN) filter					
136 -	filterInitFcn = @helperInitIMMFilter;					
137						
138	A joint probabilistic data association tracker with IMM filter	10				
139 🗨	<pre>tracker = trackerJPDA('FilterInitializationFcn', filterInitFcn,)</pre>					
140	'TrackLogic', 'History',	-				
141	'AssignmentThreshold',assignmentGate,					
142	'ClutterDensity', Kc,					
143	'ConfirmationThreshold', confThreshold, Joint Probability Data Association	Joint Probability Data Association				
144	'DeletionThreshold', delThreshold, (JPDA) tracker					
145	'HasDetectableTrackIDsInput', true,					
146	'InitializationThreshold',0);	_				
147						
148	18 · · · · · · · · · · · · · · · · · · ·	*				
Command Wind	low -	۲				
>> Tr. fx K>>	ackVehiclesUsingLidarExample					
IIII+ 3 usages o	"tracker" found In 1	139 Col 1				

- 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 ToolboxTM Computer Vision ToolboxTM

Track Vehicles Using Lidar: From Point Cloud to Track List Example

- 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 ToolboxTM Computer Vision ToolboxTM

Track Vehicles Using Lidar: From Point Cloud to Track List Example

- 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

MATLAB Coder Report Viewer - Chwork	\userpath\Examples\R2019a\d	iving_fusion_vision\TrackVehiclesUsingLic	larExample\codegen\lib\mesLidarTracker\htm	nl\report.midab:	- a x
REPORT					
Sack Forward Q Find	A B Code + Export Report LAE Code + Export Report				
 MATLAB SOURCE 	Code				0
mexLidarTracker mexLidarTracker fx mexLidarTracker fx mexLidarTracker fx HelperBoundingBoxDetect fx assembleDetections fx cropPointCloud fx getBoundingBoxes fx removeGroundPlane fx stapImpl helperCalcDetectability m fx helperCalcDetectability m fx helperCalcDetectability	mestidarTrackerm persistent derec 5 6 7 1f 1sempty(detec 8 9 X A bounding 10 11 12 13 14 15 16 17	<pre>media correcter tracker vertectagierr torModel) isempty(tracker) box detector model. 1 - HelperBoundingBoxDetector 'XLimits',[-5 75], 'YLimits',[-5 5], 'YLimits',[-2 5], 'SegmentationMinDistance' 'HinDetectionsPerCluster' 'HeasurementNoise',eye(6) 'GroundMaxDistance',0.3);</pre>	terTracker.m (10-17) → mexLidarTracki •) acksamptic Current () X min X min X min 1 X min 1 X min X max	Image: Construction Image: Construction 356 // % 357 // mextidantion 358 // mextidantion 359 // mextidantion 368 // mextidantion 364 // mextidantion 361 // mextidantion 362 // mextidantion 363 // mextidantion 364 // mextidantion 365 detectorModet 366 366 detectorModet 367 368 detectorModet 368	medurement noise in detectior - rTracker:10' detectorModel - He rTracker:12' rTracker:12' rTracker:13' rTracker:16' rTracker:16' rTracker:17' el.GroundReferenceVector[8] = (el.GroundReferenceVector[2] - (el.GroundRefere
GENERATED CODE Source Files ** AssignmentCostCalculator ** ExtendedKalmanFiler.cpp DExtendedKalmanFiler.cpp DExtendedKalmanFiler.cpp	18 19 20 20 21 21 21 21 21 21 21 21 21 21	te = [10 100]; % Assignment t d = [7 10]; % Confirmation = [8 10]; % Deletion the % False-alarm ALL MEDSAGES (0) Cration successful 24-Mar-2019 14 59:22 Static Library Microsoft Visual C++ 2017 v15 0 nma Faster Builds	hreshold; threshold for hesto rate ner unit sur- BUILD LOGE ke (64-bit Windows)	370 detectorMod 371 detectorMod 372 detectorMod 373 374 (CODE %/Br0HTS IN	DI.MINZDIStanceCluster3.8; DI.EgoVehicleRadius - 3.8; DI.IsInitialized - 0; Ctrl
14					ž

Sensor Fusion and Tracking Toolbox[™] Computer Vision Toolbox[™]

Track Vehicles Using Lidar: From Point Cloud to Track List Example

Develop Automated Driving Planning Systems with MATLAB and Simulink

Some common planning tasks

- Visualize street maps
- Connect to HERE HD Live Map
- Connect to recorded and live ROS data
- Design path planners
- Generate C/C++ code

Read Lane and Speed Information 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 ToolboxTM

Use HERE HD Live Map Data to Verify Lane Configurations

Read Lane and Speed Information 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™

<u>Use HERE HD Live Map Data to Verify Lane Configurations Example</u>

MathWorks[.] R2018a

Automated Parking Valet

- Design automated parking valet path planner
 - Inflate cost map for collision checking
 - Specify goal poses
 - Plan path using Rapidly exploring Random Tree (RRT*)

Automated Driving Toolbox[™] <u>Automated Parking Valet Example</u> MATLAB EXPO 2019

MathWorks[.] R2018a

Automated Parking Valet

- Design automated parking valet path planner
 - Inflate cost map for collision checking
 - Specify goal poses
 - Plan path using Rapidly exploring Random Tree (RRT*)

Automated Driving Toolbox[™] <u>Automated Parking Valet Example</u> MATLAB EXPO 2019

Automated Parking Valet with Simulink

- Design automated parking valet path planner and controller
 - Design lateral controller (based on vehicle kinematics)
 - Design longitudinal controller (PID)
 - Simulate closed loop with vehicle dynamics

Automated Driving ToolboxTM

Automated Parking Valet with Simulink Example

Develop Automated Driving Systems with MATLAB and Simulink

Some common integration tasks

- Synthesize scenes and sensors
- Call C/C++ code
- Call Python code
- Co-simulate through FMI/FMU
- Co-simulate through ROS
- Co-simulate with Unreal Engine
- Co-simulate with third party tools
- Automate regression testing

Integrate with ROS

Communicate via ROS to integrate with externally authored ROS components *Robotics System Toolbox*TM

MATLAB EXPO 2019

自動運動同けソフトウェアAutowareとMATLAB/Simulakの運用 Ce 16:10-16:40

#課題では、一般直自転展転向はオープンソースソフトウェアである Autoware LMATLAB[®] Simulate の事業について能介する。Autowareは、信葉 程定、時時回道、信算計画、軌道主体など自動運転システムで必要な機能を算 ただいる。Autowareのそれぞれの特徴はROS(Robot Operating System のノードムして実装されており、ROSノードとMATLAB Simulatioのインタフム 出作 MATLAB Simulatic で作成されたすディタを用いたシミュルの学校制度になる。 Autoware では成されたすディタを用いたシミュルータックが利用で ある。原始機能を得いることで、MATLAB Simulatioのギデルに対しても自動車 私の実データを用いた少くスレーションが利用になる。

大臣大平 大平臣 基础工业研究社 助者 MATLAB EXPO Japan 2017

More Information about ROS-MATLAB Interface :

https://www.mathworks.com/academia/student-competitions/tutorials-videos.html

Simulate Integration of Controls and Perception Systems

helperManaSensor.m 🗏 🕂 🖯 classdef helperMonoSensor < handle properties Sensitivity for the lane segmentation is LangSedmentationSensitivity = 0.25 48 -51 -

Visual Perception Using Monocular Camera Automated Driving ToolboxTM

46 47

49 50

Law Rodoning with Morie Camera Delector

Lane-Following Control with **Monocular Camera Perception** Model Predictive Control Toolbox[™] Automated Driving ToolboxTM Vehicle Dynamics BlocksetTM

Lane Following Control with **Sensor Fusion**

Model Predictive Control ToolboxTM Automated Driving ToolboxTM Embedded Coder[®]

Simulate Lane Following Controller with Vision based Perception

- Lane-following control with camera perception
 - MATLAB perception
 - Lane boundary detector
 - Vehicle detector
 - Simulink controller
 - Lane follower
 - Spacing control
 - Add Simulink vehicle model
 - Synthesize ideal camera image from Unreal Engine

AL A ID 1 00 · - · · · Model Button Lane Following with Mono Camera Detector Edit Setup Script Test Bench Outputer Chalase envels detection Non-Manufamorfus/da -0-Visitar Passagilian Algorithm 1 100 Evantervence/Ceasion # on Bud Velouity Perception Vehicle Model (MATLAB) lars, Incideras (Simulink) 1234 Sensor Model Control Adaptement **Unreal Interface** (Simulink) Controller And in case of the local state o (Simulink)

Automated Driving Toolbox[™] Vehicle Dynamics Blockset[™]

Lane-Following Control with Monocular Camera Perception Example

MathWorks Can Help You Customize MATLAB and **Simulink for Your Automated Driving Application**

Voyage develops MPC controller and integrates with ROS

2018 MathWorks Automotive Conference

MATLAB EXPO 2019

Autoliv labels ground truth lidar data

- Joint presentation with Autoliv
- SAE Paper 2018-01-0043
- 2018 MathWorks Automotive Conference

Ford tests algorithms with synthetic Lidar data from Unreal Engine

- Joint paper with Ford
- SAE Paper 2017-01-0107

Develop Automated Driving Systems with MATLAB and Simulink

Discuss your application with a MathWorks field engineer to help you structure your evaluation

- Understand your goals
- Recommend tasks
- Answer questions

MATLAB EXPO 2019

데모 부스와 상담부스로 질문 하시기 바랍니다.

감사합니다

