Creating Driving Scenarios from Recorded Vehicle Data for Validating Lane Centering System in Highway Traffic

Gabriel Choi, General Motors LLC
Seo-Wook Park, MathWorks
Contents

- Super Cruise – History and Future
- Lane Centering with Super Cruise
- Motivation
- Create virtual driving scenario from recorded data
- Simulate closed-loop model for lane centering system
- Conclusion
- Q & A
Super Cruise – History and Future

- Debuted in 2017 with CT6 Sedan
  - Lane Centering in addition to Full Speed Range Adaptive Cruise Control
  - Uses High-Definition Map and Front Camera to detect Lane Marks
- Automated Lane Change for 2021 Cadillac CT4 / CT5 / Escalade
  - Lane Change following Driver Request
  - Able to accelerate and decelerate slightly to search gap to change in
- Eventually will be expanded to many name plates
Lane Centering with Super Cruise on Cadillac CT6

- **Sensors**
  - Pre-Scanned High Definition Map
  - Map matching with GPS
  - Camera
  - Long Range Radar
  - Short Range Radars

- **Actuation**
  - Electric Power Steering

- **Driver Monitoring System for Safety**
  - Infra-red Face Recognition
  - Steering Wheel Touch Sensor
  - Chime and Vibration Seat
System validation for driving automation system

- **Pains**
  - Big data size from “tens of thousands of miles” test drive
    - Time consuming for data analysis
  - Not easy to reproduce a real-world traffic situation with closed-loop simulation
    - Hazardous test scenarios
    - Unwanted system behavior
- **Virtual driving scenario from recorded data**
  - Reduce development time
  - Enable closed-loop simulation to identify the root causes for unwanted system behavior

➔ How to create driving scenario from recorded data?
Create virtual driving scenario from recorded data

- Record and select data
- Reconstruct road network
- Localize ego trajectory
- Reconstruct target vehicles
- Compare with recorded video

% Create an object to access the MF4 file.
mdfObj = mdf('RecordedData.mf4');

MDF (Measurement Data Format)

mp4

GPS

Vehicle Data

On-board sensors
Create virtual driving scenario from recorded data

- Record and select data
- Reconstruct road network
- Localize ego trajectory
- Reconstruct target vehicles
- Compare with recorded video

HDLMreader = hereHDLMReader(latitude,longitude);
topologyGeometry = HDLMreader.read("TopologyGeometry");
plot(topologyGeometry);
Create virtual driving scenario from recorded data

- Record and select data
- Reconstruct road network
- Localize ego trajectory
- Reconstruct target vehicles
- Compare with recorded video

HDLMreader = hereHDLMReader(latitude,longitude);
topologyGeometry = HDLMreader.read("TopologyGeometry");
plot(topologyGeometry);
Create virtual driving scenario from recorded data

- Record and select data
- Reconstruct road network
- Localize ego trajectory
- Reconstruct target vehicles
- Compare with recorded video

GPS
Vehicle Speed

trajectory(egoCar, waypoints, speeds)
Lateral deviation of ego vehicle from lane center

![Graph showing lateral deviation from lane center]

- Lateral deviation of ego vehicle from lane center
- Lateral deviation from lane center
- GPS based trajectory
- Camera lane sensor
- Right lane boundary
- Left lane boundary

Diagram showing the lateral deviation of the ego vehicle from the lane center, with GPS-based trajectory and camera lane sensor data.
Localized ego trajectory by lane sensor data

Lateral Deviation from lane center

Camera lane sensor

Localized ego trajectory

GPS based trajectory

left lane boundary

right lane boundary

lane center

Localized

Lateral deviation

travel distance (m)

lateral deviation (m)
Visualize data

Map with GPS position `geoplayer`

Video with time stamp, ego speed & yaw rate `VideoReader imshow`

Driving scenario view `drivingScenario`

Bird’s-eye plot with sensor detections `birdsEyePlot`
Create virtual driving scenario from recorded data

Record and select data → Reconstruct road network → Localize ego trajectory → Reconstruct target vehicles → Compare with recorded video
Create virtual driving scenario from recorded data

1. Record and select data
2. Reconstruct road network
3. Localize ego trajectory
4. Reconstruct target vehicles
5. Compare with recorded video

Radar Detections
Report targets only in ego and neighbor lanes
Target vehicle trajectory

- Target vehicle trajectory is defined by a series of actor poses

- Actor poses consist of
  - Position \((X_t, Y_t)\)
  - Velocity \(v_t\)
  - Orientation \(\Psi_t\)

  in world coordinate \(X, Y\)
Estimate heading angle of target vehicle

- Radar detections provides target position and velocity in ego coordinate
  - $x_t, y_t, v_x, v_y, (\phi_t?) \leftarrow \text{radar detections}$

- Estimate heading angle of target vehicle
  - Heading angle in ego coordinate
    - $\phi_t = \tan^{-1} \left( \frac{v_y}{v_x} \right)$
    - Heading angle in world coordinate
      - $\Psi_{target} = \phi_t + \Psi_{ego}$

- Target position in world coordinate
  - $(X_t, Y_t) = (X_{ego}, Y_{ego}) + \mathbb{R}(\Psi_{ego}) \cdot (x_t, y_t)^T$
Extract target trajectory from radar detections
Example of estimated yaw angle

```matlab
filteredYaw = smoothdata(yaw,'sgolay');
% smooth yaw using Savitzky-Golay filter
```
Create virtual driving scenario from recorded data

- Record and select data
- Reconstruct road network
- Localize ego trajectory
- Reconstruct target vehicles
- Compare with recorded video

Report radar detections in ego and neighbor lanes
Synthesize sensors with virtual driving scenario

Record and select data → Reconstruct road network → Localize ego trajectory → Reconstruct target vehicles → Compare with recorded video
Integrate driving scenario + sensors with a close-loop system

Record and select data ➔ Reconstruct road network ➔ Localize ego trajectory ➔ Reconstruct target vehicles ➔ Compare with recorded video

Integrate driving scenario + sensor models with a close-loop system

Synthesize sensors
Closed-loop system for lane centering

Lane Centering System Under Test

- Radar
- Vision
- Lane

Sensor Fusion and Tracking
Estimate Lane Center
Find Lead Car
Preview Curvature
Model Predictive Control

Ego Vehicle Dynamics

- Accel
- Steering
- Speed
- Ego pose

Virtual driving scenario from recorded data

Sensor Models

Vision Detection Generator
Radar Detection Generator

Use surrogate algorithm
Surrogate closed-loop system for lane centering

Lane Following Control with Sensor Fusion and Lane Detection

*Automated Driving Toolbox™*
Surrogate closed-loop system for lane centering

- Lane Following with Spacing Control Test Bench
- Model Buttons: Edit Setup Script
- Info

- Lane following controller
- Vehicle and environment

- Ego vehicle dynamics
- Radar detection sensor model
- Vision detection sensor model

- Driving scenario reader (target vehicle poses)
Lane centering test bench with recorded target vehicles
Closed-loop simulation using reconstructed virtual driving scenario
Simulation result (longitudinal & lateral control performance)

longitudinal control performance

lateral control performance

a, b, c: problem cases where headway distance drops below the safe distance.
Driving case (b): cut-in vehicle at low speed

longitudinal control performance

Velocity

Distance between two cars

Acceleration

Collision status: 0 or 1

apply brake

cut-in from right

a,b,c : problem cases where headway distance drops below the safe distance.
Driving case (c): cut-in vehicle with too close distance

longitudinal control performance

- Velocity
- Distance between two cars
- Acceleration
- Collision status: 0 or 1

a,b,c : problem cases where headway distance drops below the safe distance.
Conclusion

- Created virtual driving scenario from recorded data
  - Reproduced real-world driving scenario in the virtual simulation environment
    - Assess functional behavior and identify root cause for problem cases
    - Reduce development time with limited resources
    - Enable repetitive tests for hazardous scenarios
Remark

- Collaborative effort between GM and MathWorks.
- This study has been published in the SAE paper.

Creating Driving Scenarios from Recorded Vehicle Data for Validating Lane Centering System in Highway Traffic

Seo-Wook Park, Kunal Patil, Will Wilson, and Mark Corless  The MathWorks, Inc.

Gabriel Choi and Paul Adam  General Motors LLC

Presenter contact info and poll questions

Please contact us with questions

- Gabriel Choi, General Motors LLC (gabriel.choi@gm.com)
- Seo-Wook Park, MathWorks (spark@mathworks.com)

- Poll questions: I found this technique the most interesting
  1. Access mdf data
  2. Road network creation from HD map
  3. Ego vehicle localization
  4. Reconstruct target vehicles
  5. Data visualization
  6. Close-loop system integration for lane centering with Simulink

- If you would like to an individual follow-up, please provide your name and email address in the WebEx poll area.