Case Study: Highway Lane Following + Lane Change
Design and test decision making, path planning, and control modules in traffic scenarios

Mark Corless
Industry Marketing, MathWorks

Seo-Wook Park
Application Engineering, MathWorks

Marco Roggero
Application Engineering, MathWorks
Evolution of ADAS and Autonomous Driving Car Technologies
Application examples in Automated Driving Toolbox™

- **L0**: No Automation
  - FCW

- **L1**: Driver Assistance
  - ACC
  - Lane Keep Assist/Lateral Support

- **L2**: Partial Automation
  - Traffic Jam Assist (LF)
  - AEB-Vehicle (City/Inter-Urban)
  - AEB-VRU (Pedestrian)
  - Lane Keep Assist/Lateral Support

- **L3**: Conditional Automation
  - Auto Pilot (LF+LC)
  - Junction Assist
  - AEB-VRU (Cyclist)

- **L4**: High Automation
  - Auto Pilot: Traffic Jam Assist
  - Auto Pilot: Road Train
  - Auto Pilot: Parking

- **L5**: Full Automation
  - Self-Driving Car
  - Self-Driving & Human-Driven Car

Timeline:
- 2010
- 2014
- 2016
- 2018
- 2020
- 2025
- 2030

Application examples in Automated Driving Toolbox™

- Auto Pilot (LF+LC) R2018b
- Lane Change
- Junction Assist R2018b
- AEB-VRU (Cyclist) R2018b
- ACC R2017b
- AEB-VRU (Pedestrian) R2018a
- Lane Keep Assist/Lateral Support R2018a
- AEB-Vehicle (City/Inter-Urban) R2017a
- FCW R2017a
Traffic Jam Assist with ACC and Lane Following Control

Automated Driving Toolbox™

**ACC** (Longitudinal Control)

**Lane Following** (Lateral Control)

**Traffic Jam Assist** (Longitudinal + Lateral Control)
Auto Pilot: Lane Following plus Lane Change

Automated Driving Toolbox™

Traffic Jam Assist
(Longitudinal + Lateral Control)

Auto Lane Change
(LC Decision Logic + Planning)

Auto Pilot
(Lane Following + Lane Change)

Combined lane following and lane change control
Single Lane Change Example
Case Study for Lane Following plus Lane Change

**Design lane following + lane change controller**
- Review baseline LF example
- Design sensor configuration
- Design additional MIO detectors
- Design safety zone calculation
- Design lane change logic
- Design trajectory planner

**Automate regression testing**
- Define assessment metrics
- Add predefined scenarios
- Run Simulink test

**Test robustness with traffic agents**
- Specify driver logic for traffic agents
- Randomize scenarios using traffic agents
- Identify and assess unexpected behavior
Lane Following Control with Sensor Fusion

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

Model Predictive Control Toolbox™
Automated Driving Toolbox™
Embedded Coder®
Review lane following test bench model architecture

Lane following controller

Vehicle and environment

Collision detection
Review lane following test bench model architecture
Review vehicle and environment components

- **Ego vehicle dynamics**
- **Radar detection sensor model**
- **Vision detection sensor model**
- **Other vehicle poses (driving scenario)**
Review lane following test bench model architecture
Review lane following controller components

Estimate lane center
- Detect lane center
- Preview curvature

Model predictive controller
- Lateral: Follow lane center
- Longitudinal: Adaptive cruise control

Tracking and sensor fusion
- Cluster and track detections
- Identify most important object (MIO) ahead
Partition design to enable inserting lane change functionality

- Estimate lane center
- Detect lane center
- Preview curvature
Partition design to enable inserting lane change functionality

Detect lane center

Preview curvature
Partition design to enable inserting lane change functionality

- Detect lane center
- Select signals based on mode
- Preview curvature
Add lane change functionality to lane following controller

1. Add more sensors

2. Identify most important objects in adjacent lanes
Add lane change functionality to lane following controller

1. Add more sensors
2. Identify most important objects in adjacent lanes
3. Identify safety zones
4. Decide when to change lanes
5. Plan trajectory
Add lane change functionality to lane following controller

1. Add more sensors
2. Identify most important objects in adjacent lanes
3. Identify safety zones
4. Decide when to change lanes
5. Plan trajectory
System requirements for lane change
Intelligent transport systems - Lane change decision aid systems (LCDAS)

Adjacent zones for Blind Spot Detection

Typically implemented with
Short Range Radar

Rear zones for closing vehicle warning

Typically implemented with
Mid Range Radar

1. Left adjacent zone
2. Left rear zone
3. Right adjacent zone
4. Right rear zone
Explore sensor placement with Driving Scenario Designer
Review sensor configuration for **lane following** example

- **Camera**
  - FoV = 43.6° x 150m

- **FrontMRR**
  - FoV = 90° x 60m

- **FrontLRR**
  - FoV = 20° x 175m

- **SRR**: Short-Range Radar
- **MRR**: Mid-Range Radar
- **LRR**: Long-Range Radar
Add rear looking sensors to support **left lane change**

- **FrontLRR**
  - FoV = 20° x 175m

- **Camera**
  - FoV = 43.6° x 150m

- **FrontMRR**
  - FoV = 90° x 60m

- **LeftRearSRR**
  - FoV = 140° x 40m

- **LeftRearMRR**
  - FoV = 30° x 80m

- **SRR**: Short-Range Radar
- **MRR**: Mid-Range Radar
- **LRR**: Long-Range Radar
Overall sensor configuration for lane following plus lane change

- **Camera**: FoV = 43.6° x 150m
- **FrontMRR**: FoV = 90° x 60m
- **FrontLRR**: FoV = 20° x 175m
- **LeftFrontSRR**: FoV = 140° x 40m
- **LeftRearSRR**: FoV = 140° x 40m
- **LeftRearMRR**: FoV = 30° x 80m
- **RightFrontSRR**: FoV = 140° x 40m
- **RightRearSRR**: FoV = 140° x 40m
- **RightRearMRR**: FoV = 30° x 80m

** ×1 front camera**
** ×8 radars**

- SRR: Short-Range Radar
- MRR: Mid-Range Radar
- LRR: Long-Range Radar
Review sensor models for traffic jam assist

Visualize with Birds Eye Scope
Add sensor models for lane change

Visualize with Birds Eye Scope

×8 radar
×1 front camera
Add lane change functionality to lane following controller

1. Add more sensors
2. Identify most important objects in adjacent lanes
3. Identify safety zones
4. Decide when to change lanes
5. Plan trajectory
Identify Most Important Objects (MIO) to detect

- **Lane following** – one EgoFront MIO is enough
- **Lane change** – needs more MIOs surrounding ego car
Review baseline MIO detector architecture for traffic jam assist

One MIO (EgoFront)

Ego lane
Add MIO detectors for lane change

Sensor Fusion

- Vision
- Radar

In1: Detection
In2: Concatenation
Out: Detection

System Clock

- In
- Detection Clustering

Ego lane

Adjacent lane

6 MIOs
Add lane change functionality to lane following controller

1. Add more sensors

2. Identify most important objects in adjacent lanes

3. Identify safety zones

4. Decide when to change lanes

5. Plan trajectory
Identify safety zones to calculate

Safety Zone for lane following

Lane following is unsafe if ego front MIO is detected within the safety zone

\[ v_f \delta + \frac{v_f^2}{2a_{\text{brake}}} \]

or,

\[ v_f \text{TTC}_{\text{FCW}} \]
Identify safety zones to calculate

Left lane change is **safe** if no MIOs are detected within the safety zone.

\[ v_{lf} \delta_f + \frac{v_{lf}^2}{2a_{brake}} \text{ or } v_{lf}TTC_{LC} \]

\[ v_{lr} \delta_r + \frac{v_{lr}^2}{2a_{brake}} \text{ or } v_{lr}TTC_{LC} \]
Identify safety zones to calculate

- Safe Zone for lane following
- Safe Zone for left lane change
- Safe Zone for right lane change

Right lane change is **unsafe** if MIOs are detected within the safety zone.
Identify safety zones to calculate

- Safe Zone for lane following
- Safe Zone for left lane change
- Trigger lane change to safe side
- Safe Zone for right lane change
Visualize safety zones

- Safe Zone for lane following
- Safe Zone for left lane change
- Safe Zone for right lane change
- Trigger lane change to safe side
Add lane change functionality to lane following controller

1. Add more sensors
2. Identify most important objects in adjacent lanes
3. Identify safety zones
4. Decide when to change lanes
5. Plan trajectory
Lane change decision logic and planning

MIOs

Ego & Adjacent lanes

Safety zone calculations

Lane Change Decision Logic

Lane Change Planning

Reference trajectory for lane change
Design lane change decision logic

Lane following mode

Lane change mode

MIOs

Ego & Adjacent lanes

Lane Change Decision Logic

FCW detected

Left lane change

Right lane change

Lane Change

DecisionLogic

LaneFollowing

LaneChange

Lane following mode

Lane change mode

Add lane change functionality to lane following controller

1. Add more sensors
2. Identify most important objects in adjacent lanes
3. Identify safety zones
4. Decide when to change lanes
5. Plan trajectory
Generate trajectory

- Quintic polynomial

\[
\begin{align*}
    s(t) &= a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \\
    \dot{s}(t) &= 5a_5t^4 + 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1 \\
    \ddot{s}(t) &= 20a_5t^3 + 12a_4t^2 + 6a_3t + 2a_2
\end{align*}
\]

where \( s = \) longitudinal or lateral distance

- Start boundary conditions

\[
\begin{align*}
    a_0 &= s_{\text{start}} \\
    a_1 &= \dot{s}_{\text{start}} \\
    2a_2 &= \ddot{s}_{\text{start}}
\end{align*}
\]

- End boundary conditions

\[
\begin{align*}
    a_5t_f^5 + a_4t_f^4 + a_3t_f^3 + a_2t_f^2 + a_1t_f + a_0 &= s_{\text{end}} \\
    5a_5t_f^4 + 4a_4t_f^3 + 3a_3t_f^2 + 2a_2t_f + a_1 &= \dot{s}_{\text{end}} \\
    20a_5t_f^3 + 12a_4t_f^2 + 6a_3t_f + 2a_2 &= \ddot{s}_{\text{end}}
\end{align*}
\]
Example of trajectory generation for lane change

**Longitudinal trajectory**

- **Traj Long**
  - Longitudinal Velocity (m/s)
  - Time (s)
  - Range: 0 to 4.5

- **Long Velo (m/s)**
  - Time (s)
  - Range: 0 to 4.5

- **Long Accel (m/s^2)**
  - Time (s)
  - Range: -2 to 2

**Lateral trajectory**

- **Traj Late**
  - Lateral Distance (m)
  - Time (s)
  - Range: 0 to 4.5

- **Late Velo (m/s)**
  - Time (s)
  - Range: 0 to 4.5

- **Late Accel (m/s^2)**
  - Time (s)
  - Range: -1 to 1
Generate reference point

- Find a point for the vehicle to follow

\[ \text{Reference Point} \]

% Normalized distance between current position and section starting point
\[ u = \frac{RXY \cdot \text{DeltaXY}}{\text{DeltaXY} \cdot \text{DeltaXY}}; \]

% Find section ending point
\[ \text{indexIncrement} = \text{ceil}(u-1); \]
Calculate deviations from reference point

\[ [x(i), y(i), \theta(i), \kappa(i), v(i)] \]

Reference point

Lateral deviation

Heading deviation

Reference trajectory

\[ d_{long} \]

\[ d_{lat} \]
Create custom visualization for safety zones and trajectory

```matlab
% Custom helper visualization to show status of MIOs, safety zones, and trajectory during lane change
properties(Access = private)
    Figure
    BEP
    OutlinePlotter
    LaneBoundaryPlotter
    SafeMIOPlotter
    UnsafeMIOPlotter
    ActorPatches
    ZoneFront
    ZoneLeft
    ZoneRight
```
Create birds eye plot with utilities from Automated Driving Toolbox

% create birds eye plot
obj.BEP = birdsEyePlot('Parent', hax,...
     'XLimits', [-60, 60],...
     'YLimits', [-20, 20]);

% create lane plotter
obj.LaneBoundaryPlotter = laneBoundaryPlotter(obj.BEP,...
     'DisplayName','Lane boundaries');

% create outline plotter for target actors
obj.OutlinePlotter = outlinePlotter(obj.BEP);
% create patches for safety zones
obj.ZoneFront = patch(hax,0,0,[0 0 0]);
set(obj.ZoneFront,'XData',[],'YData',[],...
    'FaceColor','green','FaceAlpha',0.1);

% create line for trajectory path
obj.LCPath = line(hax, 0, 0,...
    'Color','blue',...
    'LineWidth',2,...
    'LineStyle','-');
Visualize safety zones and trajectory
Case Study for Lane Following plus Lane Change

**Design lane following + lane change controller**
- Review baseline LF example
- Design sensor configuration
- Design additional MIO detectors
- Design safety zone calculation
- Design lane change logic
- Design trajectory planner

**Automate regression testing**
- Define assessment metrics
- Add predefined scenarios
- Run Simulink test

**Test robustness with traffic agents**
- Specify driver logic for traffic agents
- Randomize scenarios using traffic agents
- Identify and assess unexpected behavior
Manage testing against scenarios

**Diagram:**
- **Scenarios**
- **Sensors**
- **Algorithm under test**
- **Vehicles**
- **System simulation**
- **Metrics**
- **Test Manager**
Create test scenarios

<table>
<thead>
<tr>
<th>No</th>
<th>Test Name</th>
<th>Test Description</th>
<th>Host car</th>
<th>Lead car</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01_SlowMoving</td>
<td>Passing for slow moving lead car</td>
<td>initial velocity = 20 m/s</td>
<td>constant velocity = 10 m/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HWT = 6.5 sec (HW = 130 m)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>v_set = 20 m/s</td>
<td></td>
</tr>
</tbody>
</table>

HW : Headway
HWT : Headway time
v_set : set velocity for ego car
Create test scenarios

<table>
<thead>
<tr>
<th>No</th>
<th>Test Name</th>
<th>Test Description</th>
<th>Host car</th>
<th>Lead car</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01_SlowMoving</td>
<td>Passing for slow moving lead car</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>initial velocity = 20m/s</td>
<td>constant velocity = 10m/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HWT = 6.5sec (HW = 130m)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>v_set = 20m/s</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>02_SlowMoving WithPassingCar</td>
<td>Passing for slow moving Lead car With rapidly approaching car in adjacent lane</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>initial velocity = 20m/s</td>
<td>constant velocity = 10m/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HWT = 6.5sec (HW = 130m)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>v_set = 20m/s</td>
<td></td>
</tr>
</tbody>
</table>

HW : Headway
HWT : Headway time
v_set : set velocity for ego car
## Create test scenarios

<table>
<thead>
<tr>
<th>No</th>
<th>Test Name</th>
<th>Test Description</th>
<th>Host car</th>
<th>Lead car</th>
<th>Third car</th>
<th>Spec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01_SlowMoving</td>
<td>Passing for slow moving lead car</td>
<td>initial velocity = 20m/s</td>
<td>constant velocity = 10m/s</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HWT = 6.5sec (HW = 130m)</td>
<td>v_set = 20m/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="image" alt="Diagram" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>02_SlowMoving WithPassingCar</td>
<td>Passing for slow moving Lead car With rapidly approaching car in adjacent lane</td>
<td>initial velocity = 20m/s</td>
<td>constant velocity = 10m/s</td>
<td>Constant velocity = 33m/s</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HWT = 6.5sec (HW = 130m)</td>
<td>v_set = 20m/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="image" alt="Diagram" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>03_DisabledCar</td>
<td>Passing for disabled lead car</td>
<td>initial velocity = 20m/s</td>
<td>Stationary</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HWT = 12sec (HW = 240m)</td>
<td>v_set = 20m/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="image" alt="Diagram" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**HW**: Headway  
**HWT**: Headway time  
**v_set**: set velocity for ego car
## Create test scenarios

<table>
<thead>
<tr>
<th>No</th>
<th>Test Name</th>
<th>Test Description</th>
<th>Host car</th>
<th>Lead car</th>
<th>Third car</th>
<th>Spec</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>04_CutInWithBrake</td>
<td>Passing for cut-in car with brake</td>
<td>initial velocity = 20m/s</td>
<td>initial velocity = 18m/s</td>
<td>constant velocity = 10m/s</td>
<td>Spec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>v_set = 20m/s</td>
<td>v_set = 20m/s</td>
<td>Cut-in with brake @ 6m/s² (18m/s→10m/s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Diagram</a></td>
<td><a href="#">Diagram</a></td>
<td><a href="#">Diagram</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>05_SingleLaneChange</td>
<td>Single lane change with dense traffic condition</td>
<td>initial velocity = 15m/s</td>
<td>Slow moving</td>
<td>Dense traffic</td>
<td>Spec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>v_set = 15m/s</td>
<td>v_set = 15m/s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Diagram</a></td>
<td><a href="#">Diagram</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>06_DoubleLaneChange</td>
<td>Double lane change with dense traffic condition</td>
<td>initial velocity = 15m/s</td>
<td>Slow moving</td>
<td>Dense traffic</td>
<td>Spec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>v_set = 15m/s</td>
<td>v_set = 15m/s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="#">Diagram</a></td>
<td><a href="#">Diagram</a></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

HW : Headway  
HWT : Headway time  
v_set : set velocity for ego car

---

[Diagram](#)
Create test scenarios

<table>
<thead>
<tr>
<th>No</th>
<th>Test Name</th>
<th>Test Description</th>
<th>Host car</th>
<th>Lead car</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>07_RightLaneChange</td>
<td>Passing for slow moving lead car to right lane</td>
<td>initial velocity = 20m/s</td>
<td>constant velocity = 10m/s</td>
</tr>
</tbody>
</table>

(4) — (1) — (3) — (2)

Slow moving

HWT = 6.5 sec
(HW = 130m)

HW : Headway
HWT : Headway time
v_set : set velocity for ego car
Add assessments

- % Verify that no collision was detected
  \texttt{verify(\neg \text{collision});}

- % Ensure that the time gap between the ego vehicle and lead vehicle does not dip below
  % 0.8s for more than 5Ts at a time.
  \texttt{verify(duration(timeGap < 0.8, sec) < 5*Ts);}

- % Verify that the absolute value of lateral deviation from the lane centerline does not exceed 0.2m
  % for more than 5Ts at a time.
  \texttt{verify(duration(abs(lateralDeviation) > 0.5, sec) < 5*Ts);}

- safe distance against lead car
- lateral deviation
Review report generated by Test Manager test cases

Report Generated by Test Manager

Title: Lane Following + Lane Change Control Test
Author: Seo-Wook Park
Date: 04-Apr-2019 12:03:36

Test Environment
Platform: PCWIN64
MATLAB: (R2019a)

Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Outcome</th>
<th>Duration (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCTestCases</td>
<td>70</td>
<td>2059</td>
</tr>
<tr>
<td>StraightPath</td>
<td>70</td>
<td>2059</td>
</tr>
<tr>
<td>01_SlowMoving</td>
<td></td>
<td>304</td>
</tr>
<tr>
<td>02_SlowMovingWithPassingCar</td>
<td></td>
<td>224</td>
</tr>
<tr>
<td>03_DisabledCar</td>
<td></td>
<td>330</td>
</tr>
<tr>
<td>04_CutInWithBrake</td>
<td></td>
<td>235</td>
</tr>
<tr>
<td>05_SingleLaneChange</td>
<td></td>
<td>314</td>
</tr>
<tr>
<td>06_DoubleLaneChange</td>
<td></td>
<td>420</td>
</tr>
<tr>
<td>07_RightLaneChange</td>
<td></td>
<td>228</td>
</tr>
</tbody>
</table>
Case Study for Lane Following plus Lane Change

Design lane following + lane change controller
- Review baseline LF example
- Design sensor configuration
- Design additional MIO detectors
- Design safety zone calculation
- Design lane change logic
- Design trajectory planner

Automate regression testing
- Define assessment metrics
- Add predefined scenarios
- Run Simulink test

Test robustness with traffic agents
- Specify driver logic for traffic agents
- Randomize scenarios using traffic agents
- Identify and assess unexpected behavior
Simulate interaction between driver agents

Proof of Concept

- Graphically define driver decision logic
- Integrate into cuboid driving scenario
- Visualize and debug
Scenario Reader

Driving scenario is pre-defined by DSD

Ego car is controlled by the closed-loop controller including ego vehicle dynamics
Traffic agent

State machine implementing driver logic
Implement driver logic for traffic agent

Lane Keep mode

Maintain set velocity

Follow front MIO

FCW detected

Collision detected

Lane Change mode

Lane Keep mode

Follow front MIO

Collision detected

Lane change mode
Assign traffic agents to all vehicles except ego car

Traffic agent with driver logic state machine

Ego car with LF/LC controller
Simulate with traffic agents
Analyze results for near collision scenario
Recap: Case Study for Lane Following plus Lane Change

**Design lane following + lane change controller**
- Review baseline LF example
- Design sensor configuration
- Design additional MIO detectors
- Design safety zone calculation
- Design lane change logic
- Design trajectory planner

**Automate regression testing**
- Define assessment metrics
- Add predefined scenarios
- Run Simulink test

**Test robustness with traffic agents**
- Specify driver logic for traffic agents
- Randomize scenarios using traffic agents
- Identify and assess unexpected behavior