Level 2+ Advanced Driver Assist Algorithm Prototyping via Model Based Design

Yue Sun Et al.
AVL Company Overview – One Global Partner

**RESEARCH**
10% of turnover in-house R&D

**INNOVATION**
1,500 granted patents

**STAFF**
10,300 employees
65% engineers & scientists
300 engineers in NA

**GLOBAL FOOTPRINT**
30 engineering locations
• >220 testbeds
• Global customer support network

**GROWTH**

<table>
<thead>
<tr>
<th>Year</th>
<th>Sales (billion €)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>0.15</td>
</tr>
<tr>
<td>2017</td>
<td>1.55</td>
</tr>
<tr>
<td>2018</td>
<td>1.81</td>
</tr>
</tbody>
</table>

**SERVICES**
powertrain to vehicle integration

**STAFF**
10,300 employees
65% engineers & scientists
300 engineers in NA

**GLOBAL FOOTPRINT**
30 engineering locations
• >220 testbeds
• Global customer support network

70 years of experience
AVL Capabilities – Software & Controls
AVL’s Lv2+ ADAS/AD Function Offerings Via MathWorks Toolchain (MBD)

Rapid development & white-box libraries
- Localization
- Motion Planning & Control
- Sensor Fusion

Platform agnostic & open partnership functions
- Highway Pilot
- Urban Scenarios
- Parking Lot

MODULAR
Integration of algorithms executing at different rates

EFFICIENT
Automatic code generation of complex algorithms

TEAM SYNERGY
Team collaboration made easy across continents
ADAS/AD Function Development Platform

2018 KIA Soul EV AVL Development Platform
Lv2+ ADAS/AD MBD Process Overview

1. First-principles plant models of the system
2. Develop control algorithms
3. Co-simulate controller and system models
4. Next design iteration

Typical Feature Development Lifecycle

- Start of development
- Control functions iteration
- Timeline
- Feature Concept
- System Design
- Functional Design
- Component level Design
- Software/Hardware Development
- Functional Integration testing
- System Integration Plan
- Functional Integration Plan
- Unit device Test Plan
- Component Verification
- Feature Assessment
- Feature Sign Off
- System Validation
- System Integration Plan
- System Validation
- End of development

LEARN HERE!

NOT HERE!

Co-simulate controller and system models
1. First-principles plant models of the system

- AVL deploys various commercial vehicle dynamics tools to provide high-fidelity vehicle and powertrain dynamics within ADAS virtual environments, fully integrated to Simulink for controls development.

**AVL VSM™** provides a high fidelity attribute balancing platform which can be embedded within ADAS environments such as **VTD**.

**CarSim** and **TruckSim** provides additional sensors and environments for ADAS simulation, to close the loop for controls performance assessment via **AVL-DRIVE Autonomous™**.
1. AVL-DRIVE Autonomous
A tool for the objective assessment

AVL-DRIVE Autonomous™

- Enhances the feedback from simulations with perceived safety, safety and comfort assessment
- Provides consistent development and testing tools on road, test bench and virtual environment
- Enable the reuse of office simulation environment for continuing development phases
- Provides maneuvers for scenario variations development to maximize test coverage
1. Modeling of other key components

- **Environment Model**
  - Occupancy Grid Representation
  - Drivable Space Identification

- **Decision Making**
  - Driving Scenario Identification
  - Target Maneuver Generation

- MBD advantage:
  - **Partnership Open**
2. Develop control libraries

Motion Planning SW development

Structured Motion Planning
- Algorithms
  - Model Predictive Control Optimization & Dynamic Programming for Path & trajectory sample generation and selection
- Data Structure
  - ArrayList to store path and trajectory information

Unstructured Motion Planning
- Algorithms
  - Hybrid A* and Post Optimization
- Data Structure
  - Graphs and Priority Queues (MIN-Heap) for Forward State Generation & Search State Bookkeeping
  - Hash Tables for Motion Primitive Look-Up & Cost Association

A* associates costs with centers of cells and only visits states that correspond to grid-cell centers.

Hybrid A* associates a continuous state with each cell, and the score is the cost of its associated continuous state.
2. Develop control libraries

Motion Control SW Development

- Algorithms
  - Longitudinal Control
    - **Feedforward** and **feedback PI** control for throttle and brake
  - Lateral Control
    - **Extended Stanley** method for steering control
2. Develop control libraries

Localization
- Algorithms
  - Extended Kalman Filter
- MBD advantage
  - Rapid control prototyping

3-D positions [m] 3-D velocities [m/s] @ max. 20Hz

Output
- 2-D position [m] (North-East in ENU coordinate)
- 2-D velocity [m/s] (Vehicle XYZ Frame)
- Longitudinal/lateral accelerations [m/s²] (Vehicle XYZ Frame)
- Heading angle [rad] (referenced to North in ENU coordinate)
- Heading rate change [rad/sec]
2. Develop control libraries

Integration – Threading & Parallelization

- Modules implemented as *model reference* function-call subsystems to enable:
  - Different execution rates
  - Core partitioning on the hardware

- MBD Advantage
  - Hardware platform *agnostic*
  - Available *tools* for synchronization and memory configuration to grant successful threading and parallelization

Example of execution timeline on Speedgoat

![Diagram of execution timeline on Speedgoat](image_url)
3. Co-simulate controller and system models

MIL Simulation Results

Static Vehicle Take Over

Dynamic Vehicle Trajectory Prediction and Take Over
3. Co-simulate controller and system models

**MIL Simulation Results in Parking Lot**

**Pose = (x, y, Θ)**

- **Start Pose:** x=30, y=25, Θ=15°
- **Goal Pose:** x=83, y=46, Θ=90°

- **Start Pose:** x=10, y=80, Θ=0°
- **Goal Pose:** x=57, y=92, Θ=-90°

- **Start Pose:** x=5, y=20, Θ=90°
- **Goal Pose:** x=78, y=90, Θ=90°

*a:* A*, o:* Hybrid A*
4. Next design iteration

Simulation output used to increase feature maturity resulting in:

- Honed requirements
- Feature performance improvements
- Library iteration

Typical Feature Development Lifecycle

Next Design Iteration

End of development

Timeline

Control functions iteration

Co-simulate controller and system models
AVL’s Lv2+ ADAS/AD Function Offering Via MathWorks Toolchain (MBD)

Rapid development & white-box libraries
- Localization
- Motion Planning & Control
- Sensor Fusion

Platform agnostic & open partnership functions
- Highway Pilot
- Urban Scenarios
- Parking Lot

MODULAR
Integration of algorithms executing at different rates

EFFICIENT
Automatic code generation of complex algorithms

TEAM SYNERGY
Team collaboration made easy across continents
Additional AVL Offerings in ADAS/AD Domain

Typical Feature Development Lifecycle

- Feature Concept
- System Design
- Functional Design
- Component Level Design
- Software/Hardware Development
- Feature Assessment
- System Integration Plan
- Functional Integration Plan
- Unit/Device Test Plan
- Component Verification
- System Validation
- Functional Integration testing
- System Integration Plan
- Component level Design
- System Sign Off

Timeline:
- Start of development
- End of development

Test Case Generator
- Based on scenarios
- Parallel computing

Cloud Master
- Objective evaluation of Automated Driving (incl. perceived safety)

AVL VSM™
- Precise vehicle dynamics simulation

Sensor Simulation
- Models for Radar, Lidar, Camera and Ultrasonic

Environment Simulation
- Traffic, road and environment modelling

AVL CAMEO™
- Active E2E test optimisation

AD Controller
- e.g. ACC/LKA, AEB, Highway Pilot, Traffic Jam Pilot, Automated Parking
Key Contacts & Contributors

**Andy Roberts** - Skill Team Lead, Controls, Systems & Electrification
andrew.roberts@avl.com

**Yue Sun** – ADAS/AD Controls & SW Lead
yue.sun@avl.com

**Justin Floro** – ADAS/AD Systems Lead
justin.floro@avl.com

**Frederic Jacquelin** – Chief Engineer, Simulation
frederic.jacquelin@avl.com

**Jennifer Durfy** – Skill Team Lead, Vehicle Engineering
jennifer.durfy@avl.com

**Thomas Mueller** – Technical Specialist, Vehicle Engineering
thomas.mueller-werth@avl.com

**Ankit Goila** – Project Engineer Controls
ankit.goila@avl.com
Thank You

www.avl.com