Model-based Development of Low-speed Vehicle Motion Control Algorithms for Automated Driving Functions

Dr. Marco Wegener, Hemanth Chakravarthy
Agenda

1. ZF Introduction
2. Overview Automated Driving Systems
3. Model-based Development of Vehicle Motion Control Algorithms
4. Development Results
5. Conclusion & Outlook
01
ZF Introduction
Overview Automated Driving Systems
Evolution of Automated Driving Systems

- Assisted (Level 0)
- Feet off (Level 1)
- Hands off (Level 2)
- Eyes off (Level 3)
- Brain off (Level 4-5)
Scalable System Architecture
Example: Parking Functions

• Basic Park Aid
  • Front & Rear Ultrasonic Sensors
  • Rear Camera

• Semi and Fully Automated Park Assist (Object-based)
  • Front & Rear Ultrasonic Sensors
  • Rear Camera
  • Corner Radars

• Fully Automated Park Assist (Object & Marking-based)
  Remote & Valet Parking
  • Front & Rear Ultrasonic Sensors
  • Corner Radars
  • Surround Cameras
Model-based Development of Vehicle Motion Control Algorithms
Target Functional Architecture

Environment Perception

- Object Representation
- Surrounding Objects
- Road Model
- Lane Topology Representation

Decision Making & Trajectory Planner

- Path Representation
- Path
- Target Velocity Profile
- Target Velocity Representation

Vehicle Motion Control

- Steering Angle
- Vehicle Speed
- Gear Position

Function Logic & HMI Control

Model-Based development
Non-Model-based (Manual Coding)

Ego Vehicle Localization

Model-based Development of Automated Driving Functions
Development Goals

• **Trajectory Planning**
  ▪ Flexible to any potential perception layer interface
  ▪ Capable of all kind of low-speed maneuvers (traffic jam, urban driving, parking)

• **Vehicle Motion Control**
  ▪ Combined lateral and longitudinal control
  ▪ Able to handle actuator constraints and time delays
  ▪ High precision

• **Tooling**
  ▪ Time-efficient implementation & testing with small agile development teams
  ▪ Setup of a simulation framework
  ▪ Establish toolchain to derive Key Performance Indicator driven function performance reports to track progress
Challenges of Algorithm Design

**Trajectory Planning**
- Indefinite number of potential planning algorithms
- Many concepts fail to consider:
  - Passenger car dynamic constraints
  - Multi-stroke path planning
  - Reproducibility requirement
- Re-use of well-established planning algorithm desired
- Need to pick an appropriate concept right from the beginning to meet mass-production requirements

**Vehicle Motion Control**
- High precision within cm range required
- Odometry of mass production vehicles is inaccurate close to standstill
- Actuators delay algorithm requests by low-pass filter or dead time behavior (e.g. braking down to standstill)
Benefits of Model-based Development for Automated Parking

Trajectory Planning:

- **Tools:** Automated Driving & Navigation Toolboxes provide a comprehensive set of different path planning algorithms including associated functionalities of:
  - Path Interpolation
  - Optimization
  - Visualization
  - Path Metrics

- **Support:** MathWorks engineering team support to figure out the pros & cons of each planning concept and to choose a concept that fits best to the intended application

- **Collaboration:** Engineering teams from ZF and Mathworks have mutually benefitted from monthly workshops to understand each other’s need and therefore to improve their products
Benefits of Model-based Development for Automated Parking

Vehicle Motion Control:
• Ready-to-use toolboxes as the Control System & Model Predictive Toolboxes provide advanced control algorithms capable of:
  ▪ Predicting control errors by taking into account planned trajectory and future vehicle motion
  ▪ Consideration of actuators limitations & time delays
  ▪ Comfort & safety requirements can be fulfilled already by design resulting in less tuning effort
• Simple integration into rapid-prototyping ECUs by using Simulink Coder and calibration tools for early vehicle tuning sessions

Tooling:
• Simulink enables function simulation at an early development stage and low-effort integration in simulation tools
• Built-in functions allow quick development of function-specific data analysis and performance validation frameworks
04 Development Results
Established Scenario-driven Development & Evaluation Process

Use-case Specification
- Scenario Catalogue

Design & Implementation
- Closed-loop Simulation Framework

Vehicle-level Validation
- KPI-based Performance Evaluation

Model-based Development of Automated Driving Functions

© ZF Friedrichshafen AG
Example Maneuver: Two-stroke Urban Maneuvering

Sensed Occupancy Grid & Planned Trajectory

Highly-accurate Vehicle Motion Control

---

Model-based Development of Automated Driving Functions

© ZF Friedrichshafen AG
Realized Workflow for Continuous Performance Evaluation

**Initialization**
1. Playlist Configuration
2. Run Evaluation Tool

**Automated Trimming**
1. Function-wise Use Cases and Scenario Identification
2. Log File Trimming

**Performance Evaluation**
1. Scenario-wise KPI Generation
2. Function-wise KPI Generation

**Report Generation**
1. Scenario Reports
2. Function-based Trend Report

**Result**
1. Mgt Dashboards
2. Detailed Performance Reports

**Data Logging**
- Upload Process
- File Format Conversion

**Data Management**
- Upload Process
- File Format Conversion

**Validation Team**
- Database
- Server

**MATLAB-based Performance Evaluation Tool**
Conclusion & Outlook
Conclusion

Summary:

• ZF’s ADAS function development is constantly evolving with new functions, each starting with a rapid-prototyping approach in the early project phases and then smoothly transitioning them into mass-production ready solutions.

• Model-based software development is seen as an important building block for time and cost-efficient development processes.

• Ready-to-use toolboxes by Mathworks enabled the development team to accelerate the development of Automated Driving functions.

Outlook:

• Automated Driving functions will be further developed to cover extended use-cases and increased autonomy.

• ZF will continue to benefit from rapid-prototyping capabilities enabled by additional toolboxes for localization, navigation and V2X communication.
SEE. THINK. ACT.