Improvement of Lane Keeping Assistance ADAS Function utilizing a Kalman Filter Prediction of Delayed Position States

virtual vehicle

Enabling future vehicle technologies

Selim Solmaz

Virtual Vehicle Research Center GmbH
Electrics, Electronics & Software Department (Dept.-E)
Control Systems (CS) Group
selim.solmaz@v2c2.at

Georg Nestlinger, Georg Stettinger

Virtual Vehicle Research Center GmbH
Electrics, Electronics & Software Department (Dept.-E)
Control Systems (CS) Group

05.11.2019

The **2019 IEEE ICCVE** – International Conference on Connected Vehicles and Expo

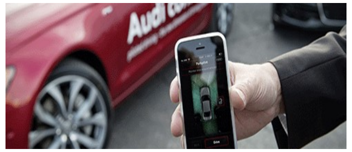
Presentation Outline

- Autonomous/Automated Driving Roadmap
- ADAS/AD Function Infrastructure
- MWC & Lane Keeping System Controller
- ADAS/AD Demonstrator Vehicle
- Description of the Implementation Problem
- Diagnosis of the Problem via Simulation
- Kalman Filter Predictor based Mitigation
- Results and Discussion

Levels of Automation (SAE Level 0-5)

Level 0: No Automation (warning of driver)
e.g. acoustic/haptic warning, display etc.

Level 1: Driver Assistance (e.g. break assistant etc.)
driver assistance system of "either steering or acceleration/deceleration"



Level 2: Partial Automation
Combined driver assistance system (e.g., Adaptive Cruise Control (ACC), Lane Keeping Assistant (LKA), or traffic jam assistant: road measurement and/or distance measurement)

Level 3: Conditional Automation
Relevant driving tasks by system, driver ready to take over

Level 4: High Automation
All driving tasks by system, limited roadway and environmental conditions (e.g. highway)

Level 5: Full Automation
All driving tasks in all situations by system ("no steering wheel/pedals"): autonomous driving

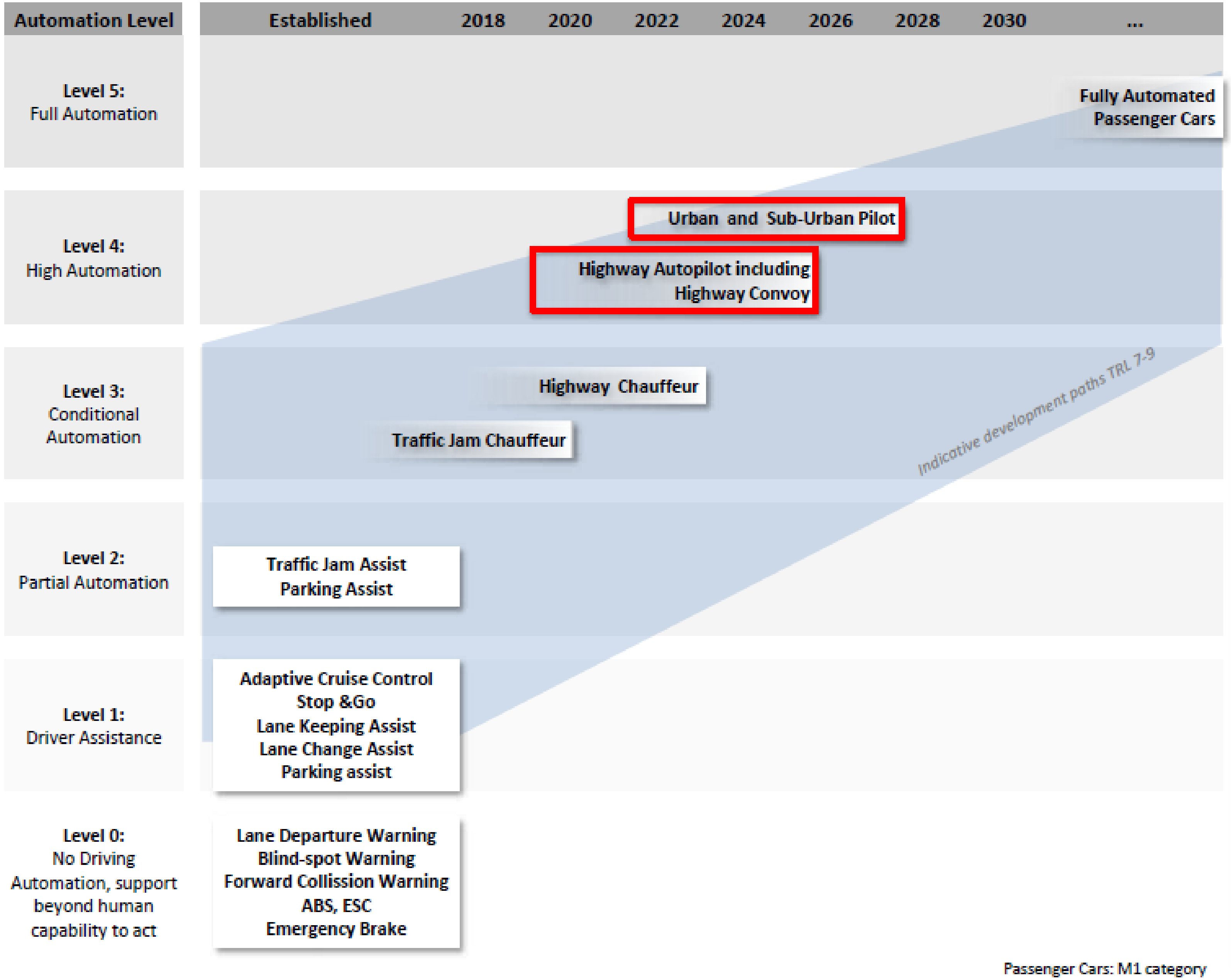
Monitoring: human driver

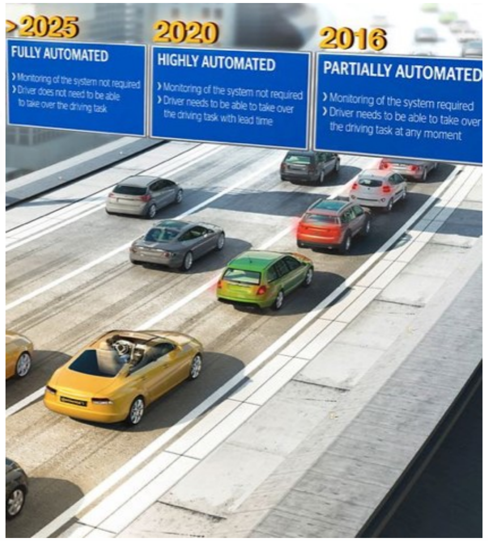
Monitoring: system

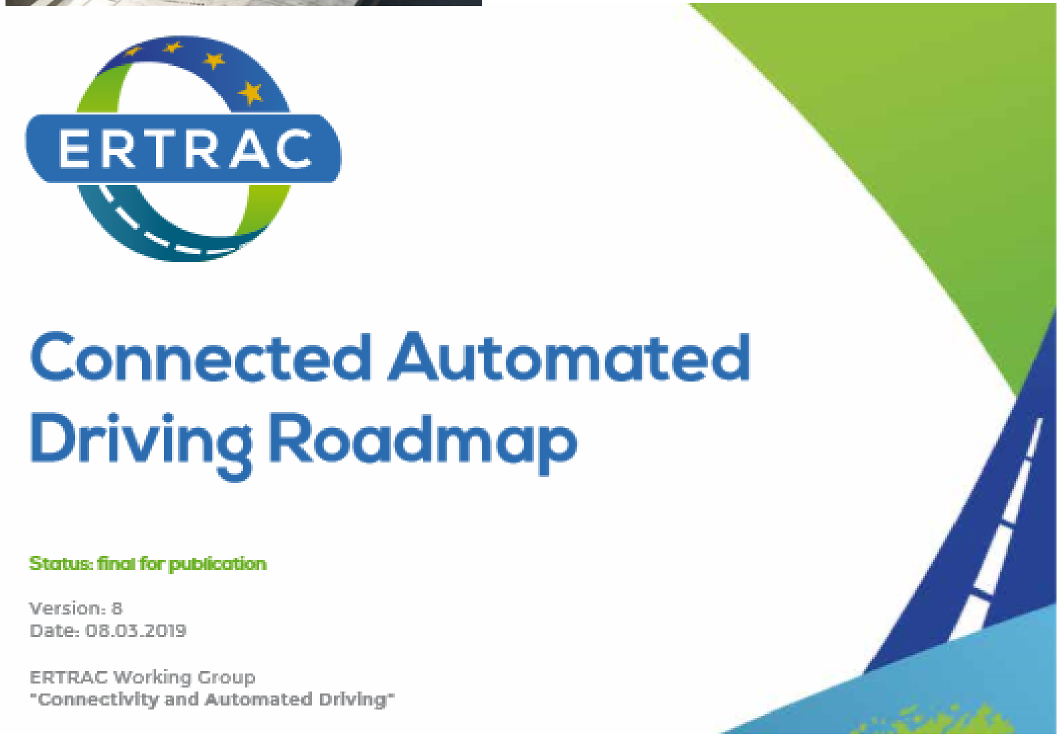
Quelle: AVL

Wording:
Automated driving ≠ Autonomous driving
Levels of automation 0 to 5

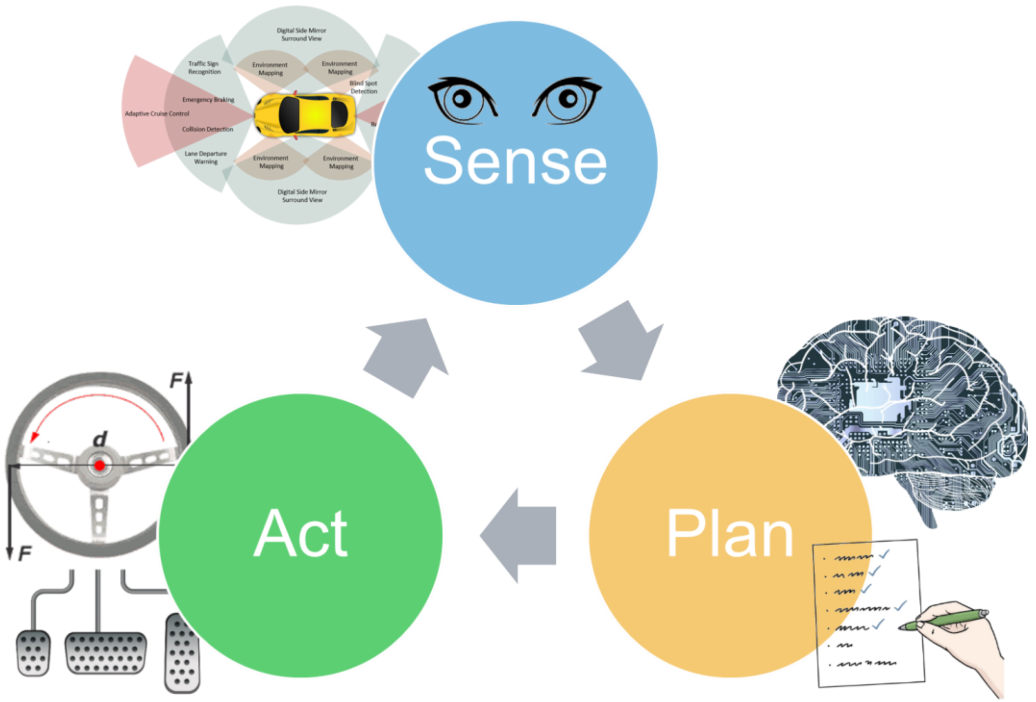
Automated/Autonomous Driving Roadmap

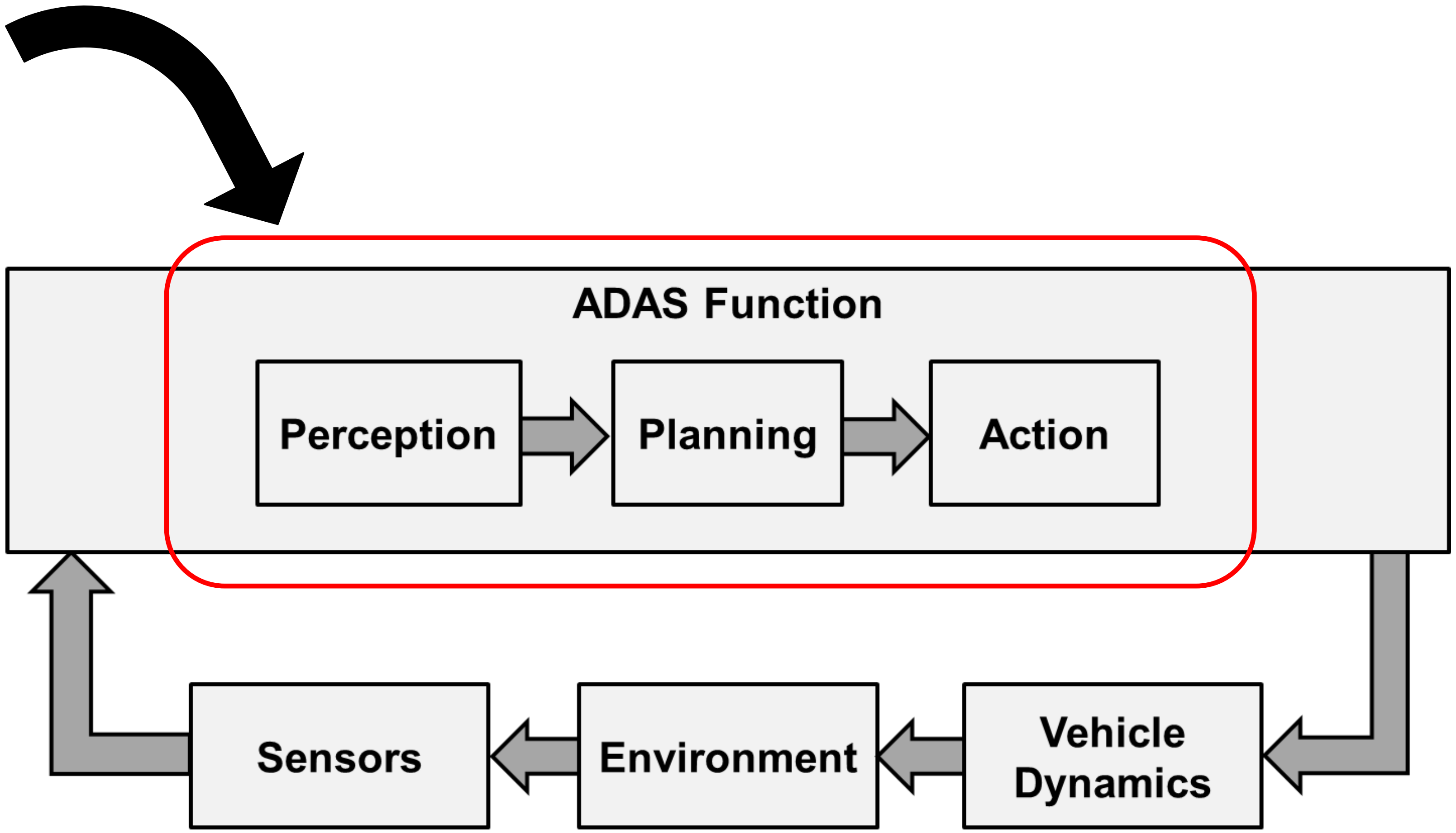




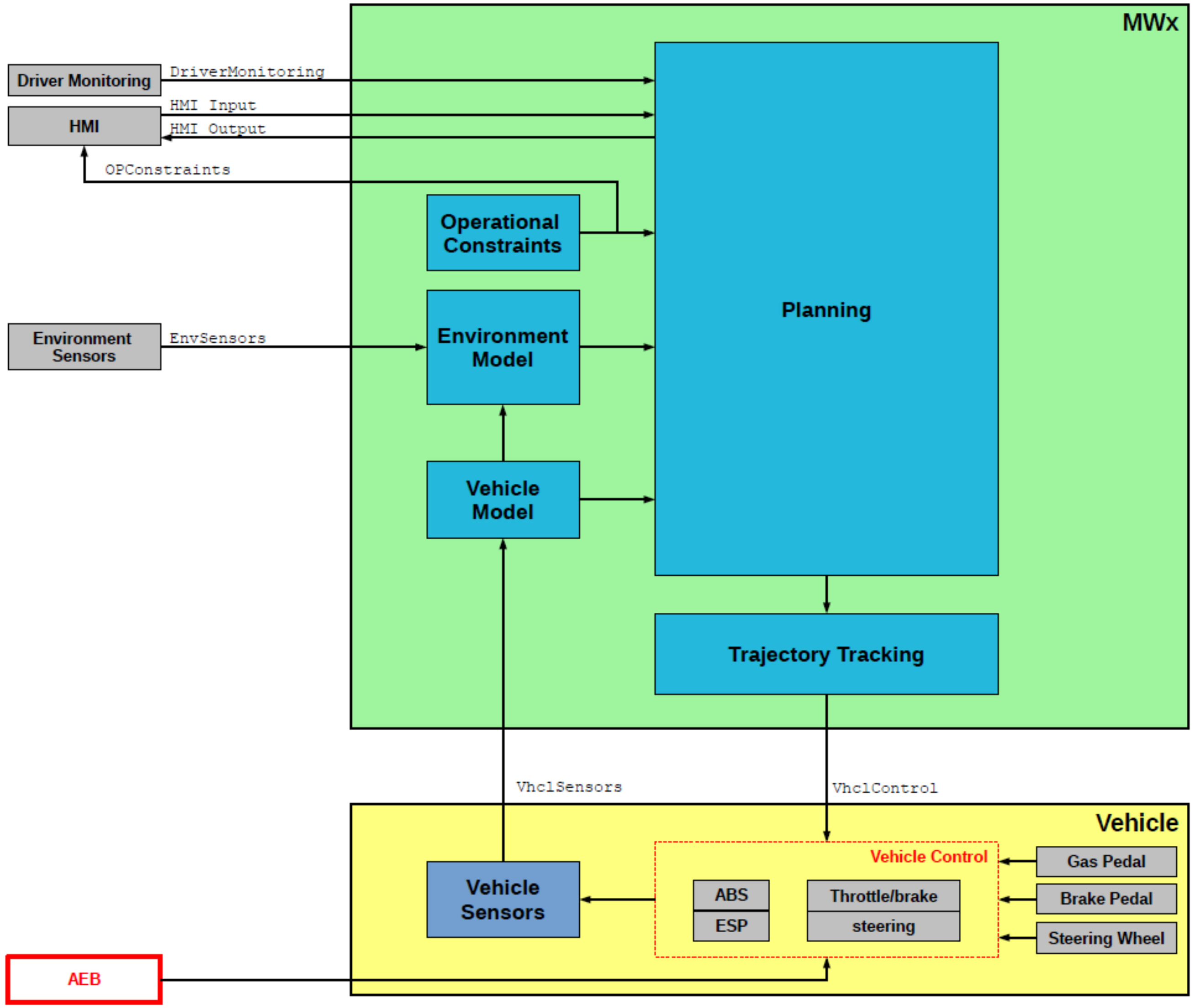


ADAS/AD Function Infrastructure

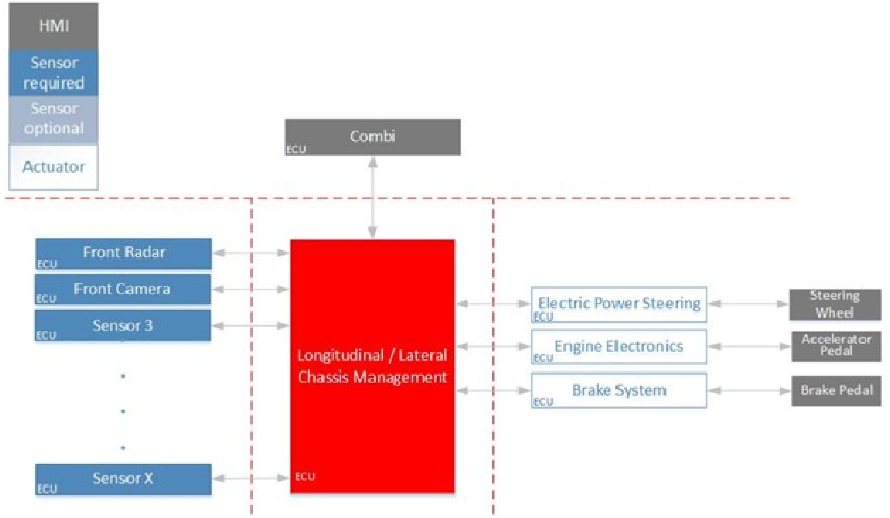




Virtual Vehicle MWC Controller Architecture

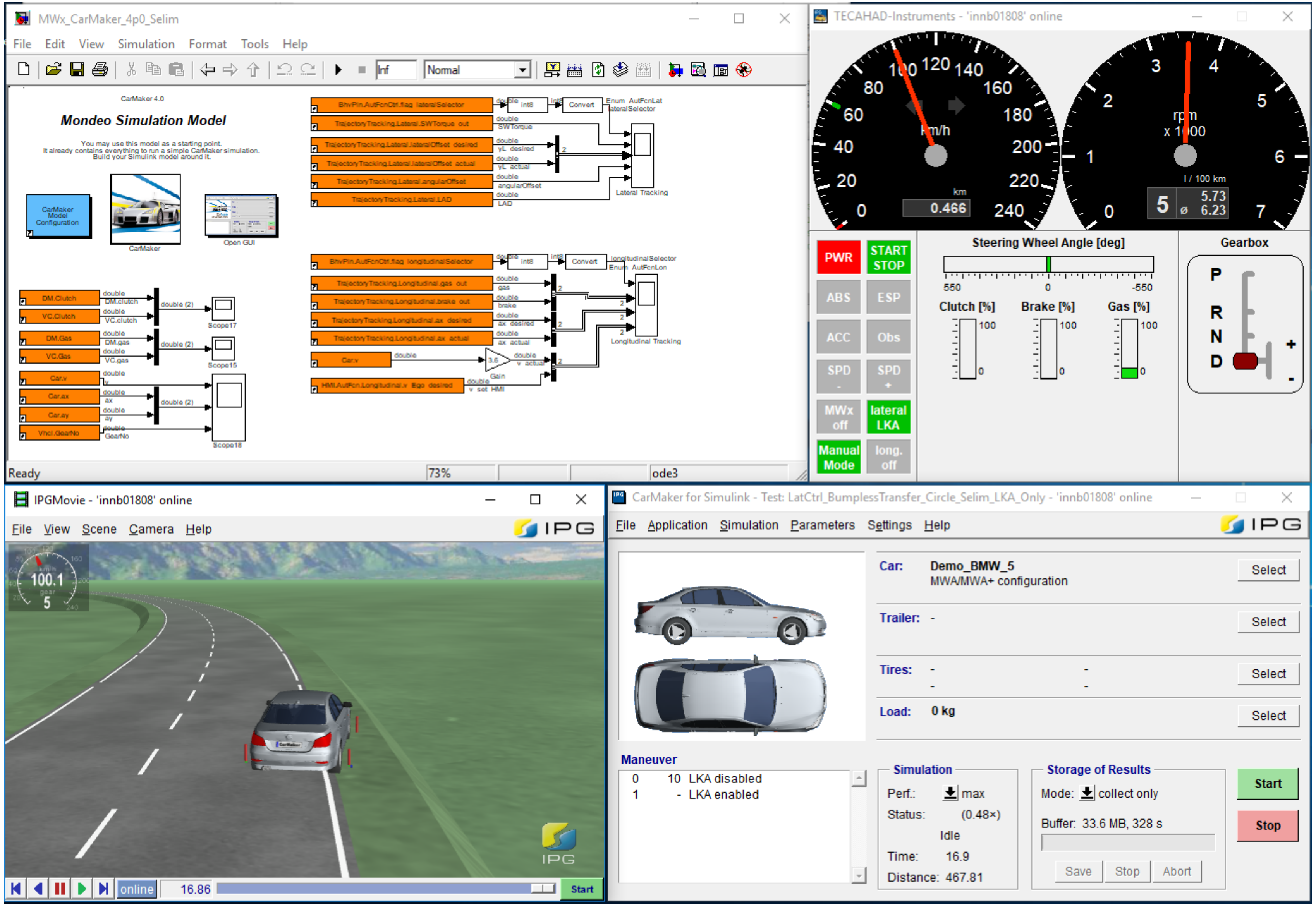


- An in-house developed Level 3+ Motorway Chauffeur Implementation
- State-based multi-controller combining
  - LKA (Lane Keeping Assistant)
  - ACC (Adaptive Cruise Controller)
  - TJA (Traffic Jam Assistant)
  - TP (Trajectory Planner)
- MATLAB/SIMULINK based development of individual MWC modules combining:
  - IPG-CarMaker based vehicle dynamics model
  - Individually designed controller logic for each MWC module in SIMULINK

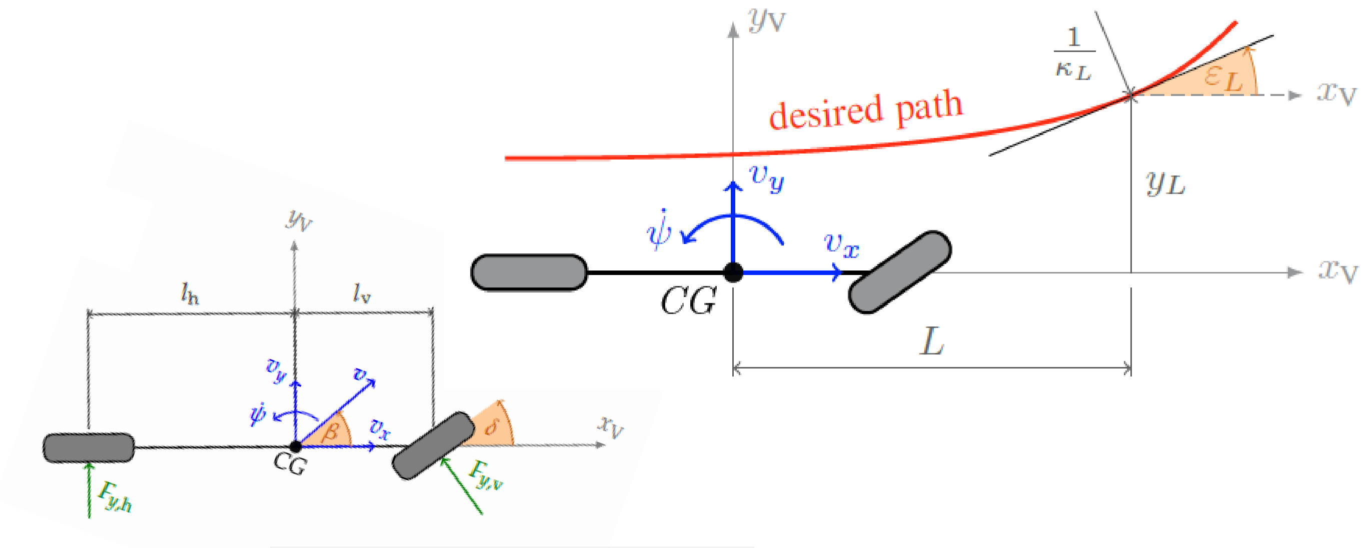


Simulation Environment

- CarMaker – Matlab/Simulink environment was developed during a former project
- Many autonomous functions (ACC, LKA, TJA, TP and their combination MWx) were developed in a realistic environment and tested in simulation



Lane Keeping (LKA) Controller Architecture*



$$\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + b\delta + b_S\kappa_L \qquad x := \begin{bmatrix} v_y & \dot{\psi} & y_L & \varepsilon_L \end{bmatrix}^\top$$

$$A = \begin{bmatrix} -\frac{c_{s,v}+c_{s,h}}{m v_x} & \frac{-c_{s,v} l_v + c_{s,h} l_h}{m v_x} - v_x & 0 & 0 \\ \frac{-c_{s,v} l_v + c_{s,h} l_h}{I_z v_x} & -\frac{c_{s,v} l_v^2 + c_{s,h} l_h^2}{I_z v_x} & 0 & 0 \\ -1 & -L & 0 & v_x \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} \frac{c_{s,v}}{m} & \frac{c_{s,v} l_v}{I_z} & 0 & 0 \end{bmatrix}^\top \qquad b_S = \begin{bmatrix} 0 & 0 & 0 & v_x \end{bmatrix}^\top$$

- Preview controller using lateral offset (y_L) and heading (ε_L) error
- Linear state feedback controller based on LQR methodology and the lane keeping model

$$\delta = -k^\top x$$

- Functional *J(t)*

$$J[\delta(t)] = \int_0^\infty \left(x^\top(t) Q x(t) + R\delta^2(t) \right) \mathrm{d}t$$

- Solve the optimization problem minimizing *J(t)*

$$\min_{k} J$$

* Nestlinger, G., & Stolz, M. (2016). *Bumpless transfer for convenient lateral car control handover*. 132-138. 9th IFAC Symposium on Intelligent Autonomous Vehicles, Leipzig, Deutschland. https://doi.org/10.1016/j.ifacol.2016.07.721

Automated Drive (AD) Demonstrator Test Vehicle

- **Drive by wire:**

  DataSpeed ADAS Kit:
  drive, brake, steer, visualize by wire

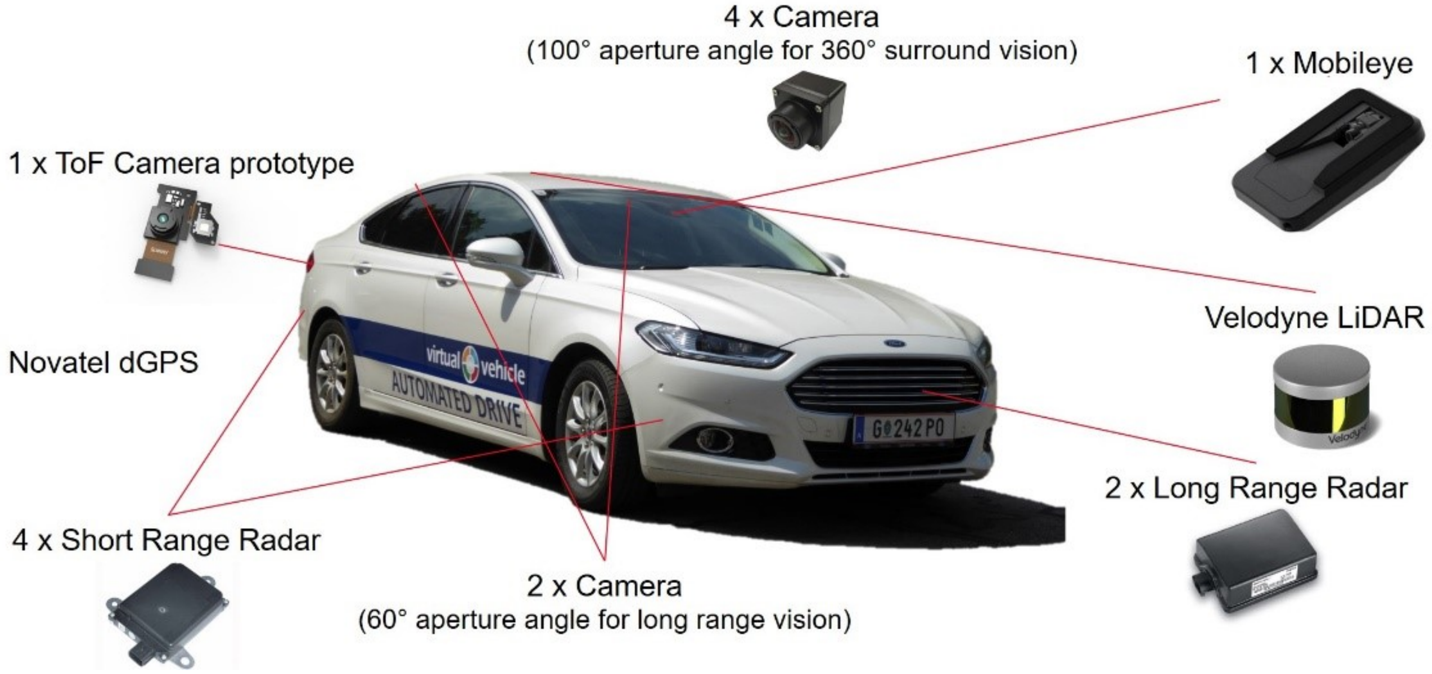
- **Sensors:**

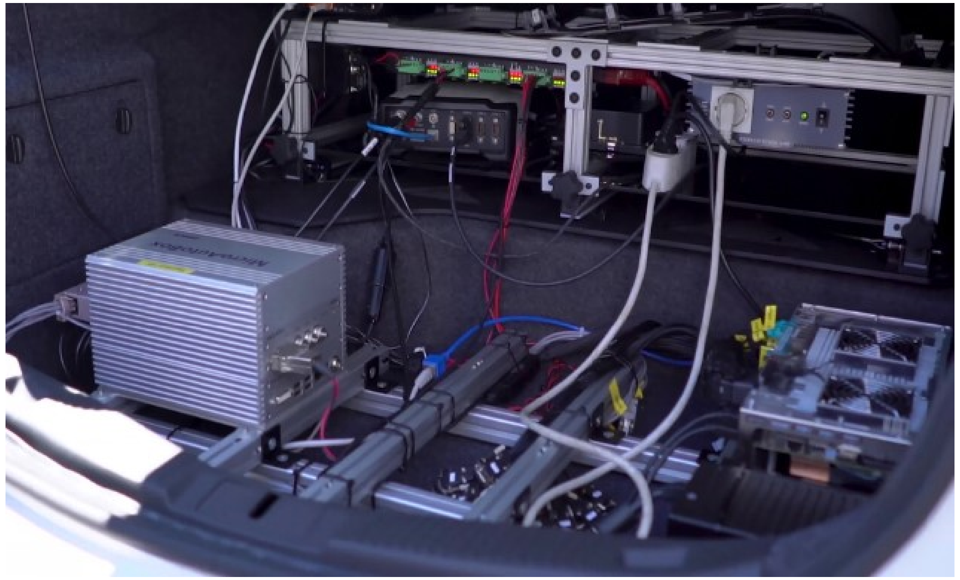
  Cameras, ultrasonic sensors, inertial sensors, RTK-GPS, Radars, Lidar(s), ToF…

- **Interfaces:**

  HMI touch display, CAN,
  ROS (Robot operating System) Kinetic
  Nvidia Drive PX/2 (Ubuntu 16.04)
  dSPACE MicroAutoBoX II
  PC (Win/Linux)

- **Applications**:
  - ❑ *Measurement (sensor data acquisition, sensor fusion)*
  - ❑ *Development and test (ADAS/AD)*
  - ❑ *Energy management (hybrid car)*
  - ❑ *Proving ground platform*





AD Demonstrator Sensor Layout

2x Long Range Radar
Continental ARS408

4x Short Range Radar
Continental ARS408

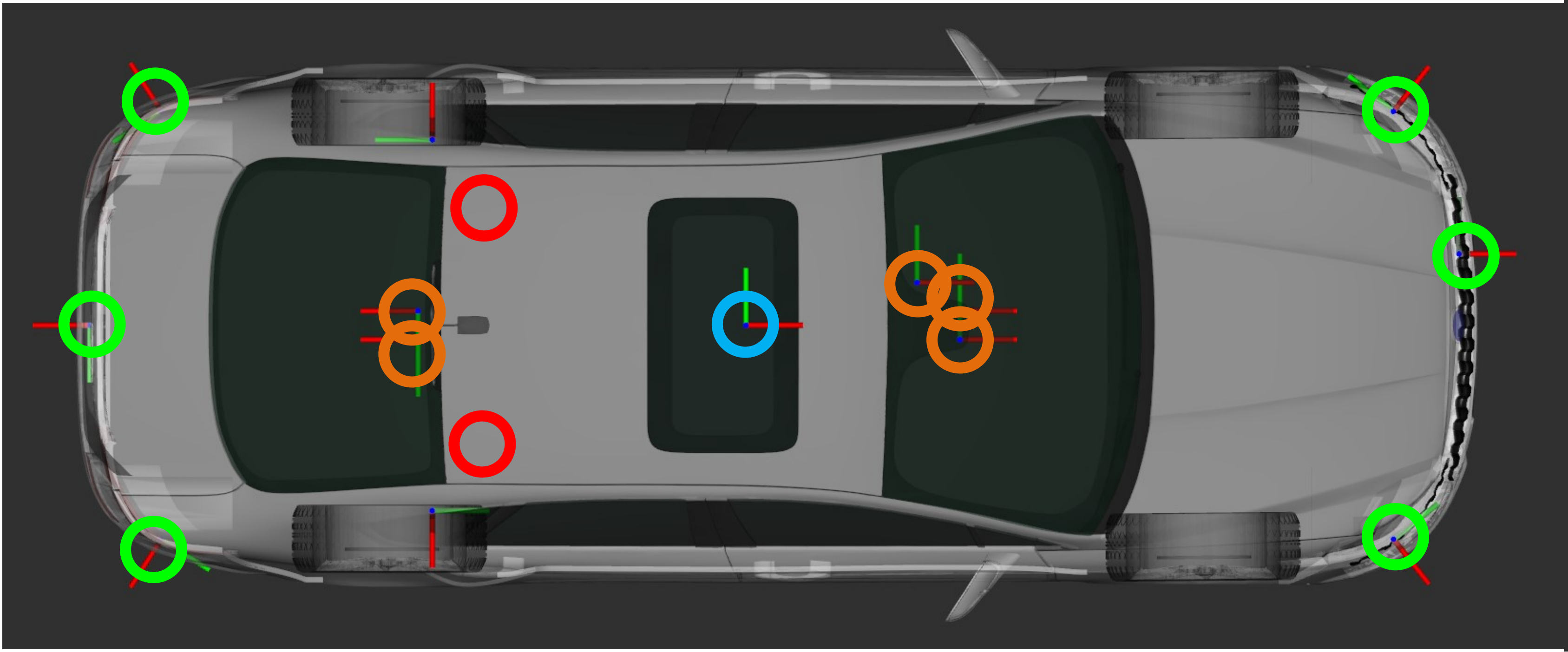
4x Camera
(100° aperture angle for 360°)

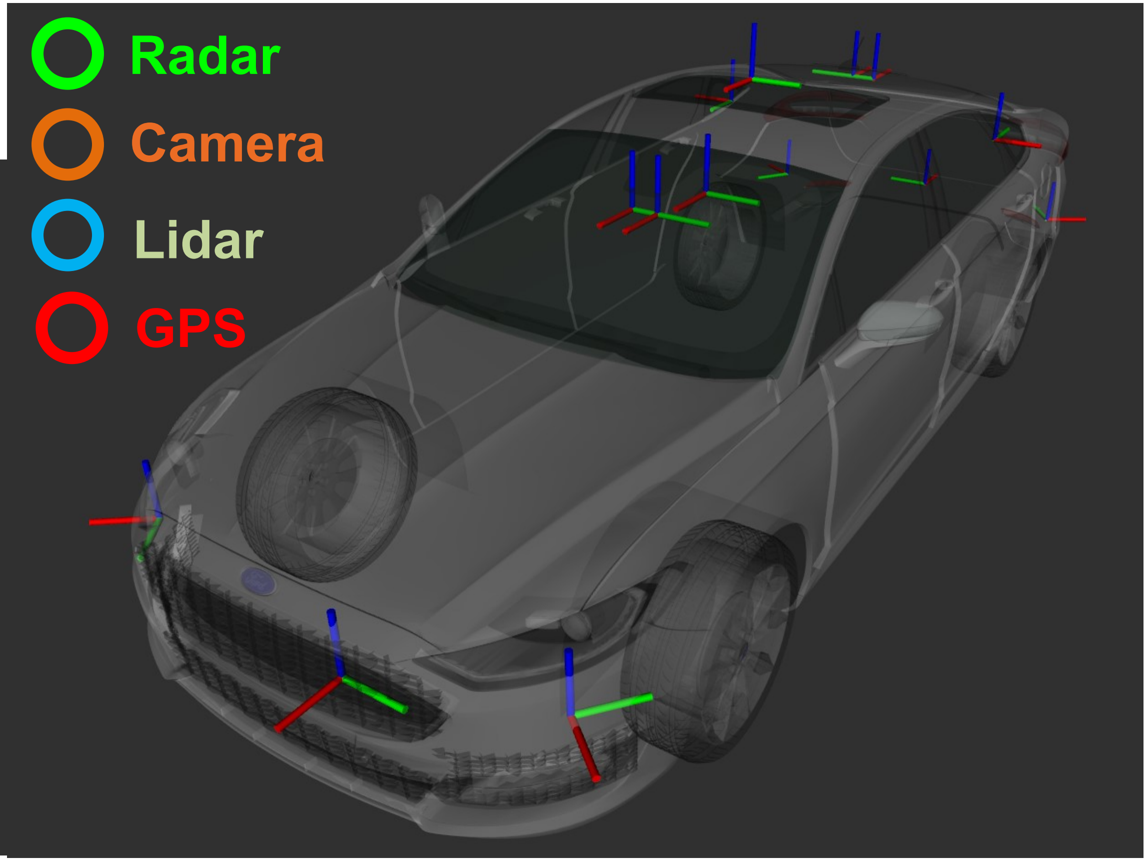
2x Camera
(60° aperture angle for long range)

Mobileye

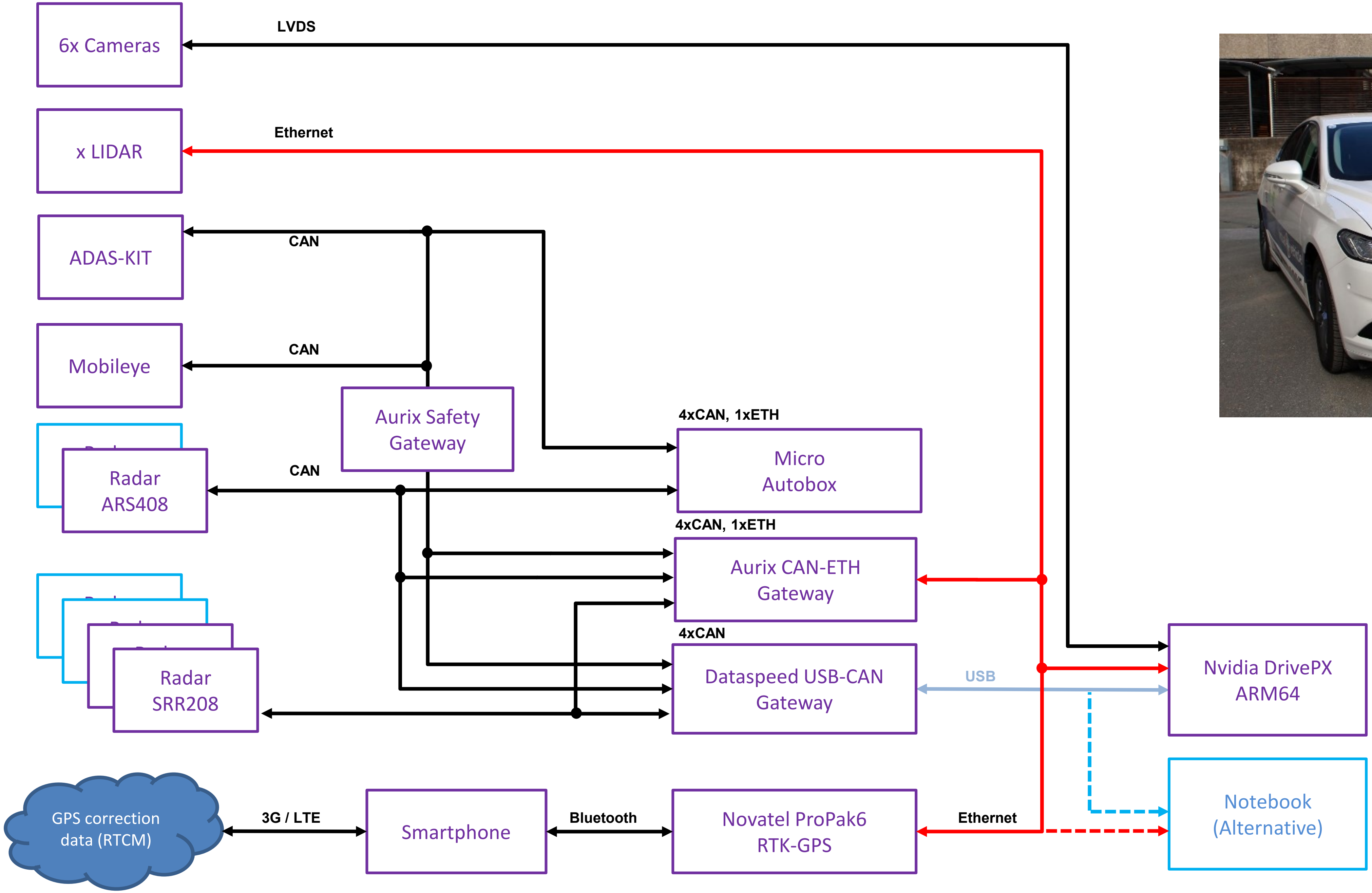
1x Lidar
Velodyne VLP-16

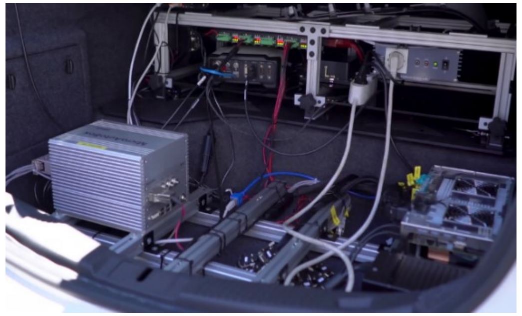
RTK-GPS
Novatel ProPak 6





AD Demonstrator-E/E Architecture

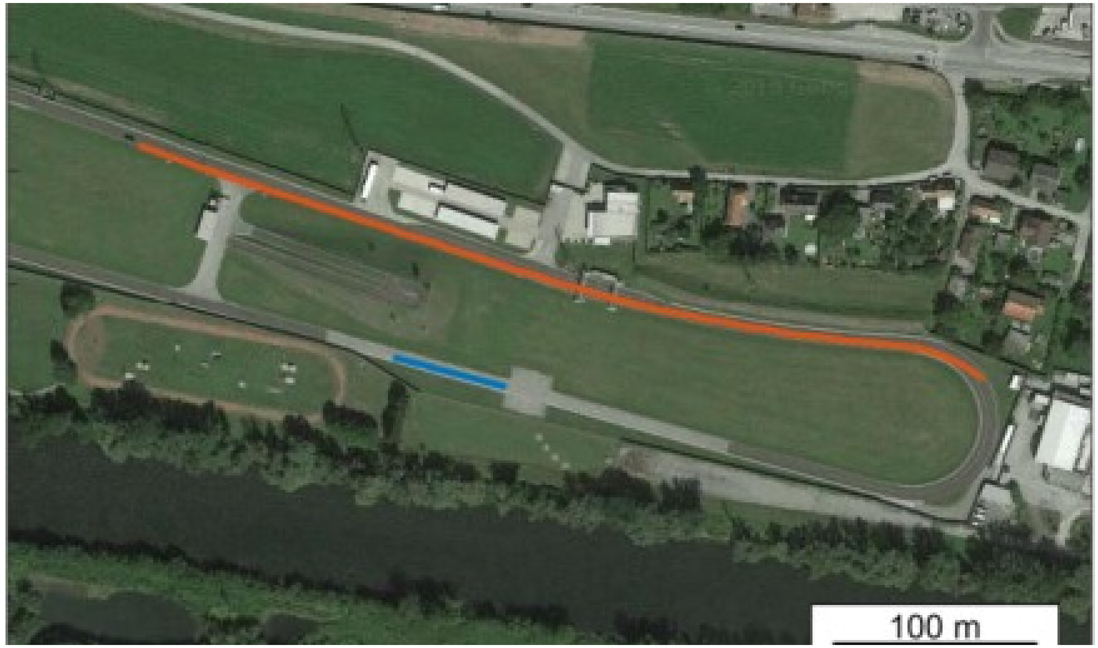


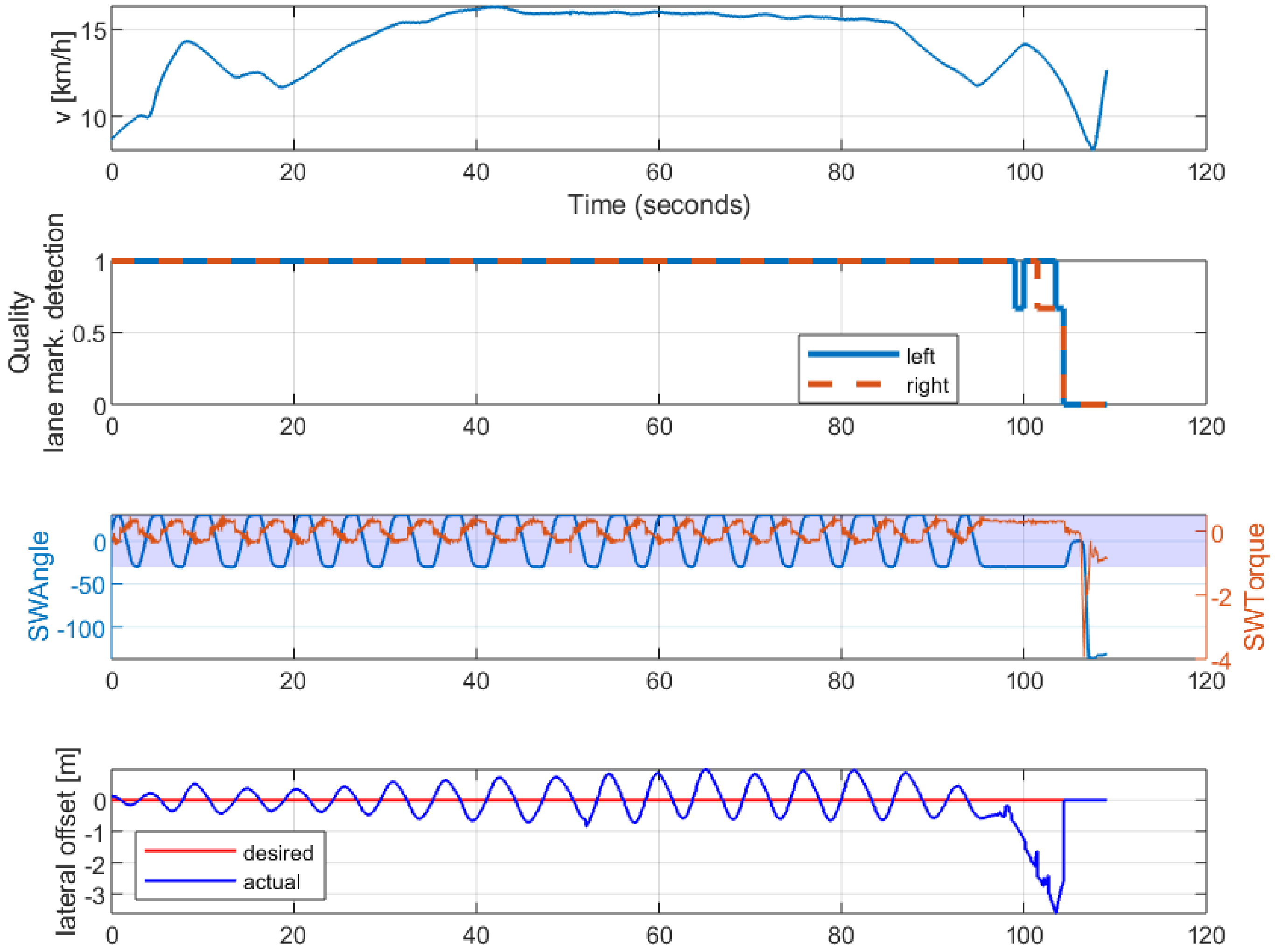


Description of the Implementation Problem

- The developed LKA controller was implemented on the AD Demonstrator using the MATLAB/SIMULINK Embedded coder to run on the dSPACE MicroAutobox-II real-time ECU
- LKA Function based on DataSpeed ADAS Kit and MobilEye Camera alone

Initial Implementation Performance Results

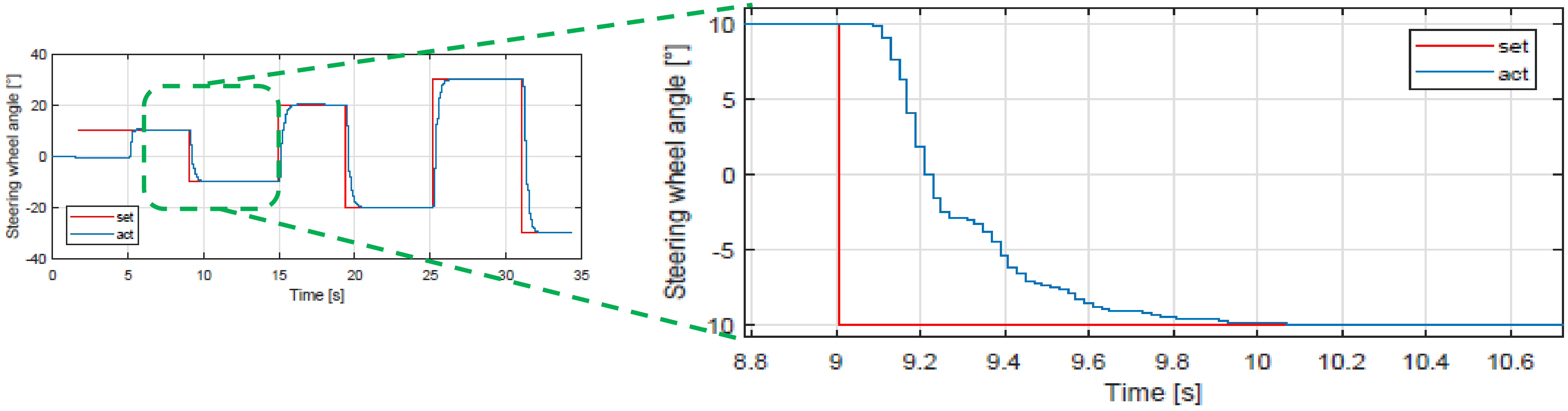




- Our first "successful" LKA test at the AVL test track in Gratkorn in late July 2018
- First trials with the same simulation algorithms had performance issues
  - Performance problems with tracking
  - Limited speed (~35 km/h)
  - Oscillations and unstable behavior at high speeds
  - Gain and parameter tuning didn't help much

Problem Details

- LKA Performance Diagnosis:
  - LKA working only at low speeds
  - Has oscillatory behavior in keeping the lane
- Both problems originate from the delays in sensors and the control actuator
- Steering actuator has a consistent delay of 80 – 100 *ms* which were measured with experiments
- MobilEye 630 sensor providing the lane information has unknown (and unfortunately unmeasurable) delay
- We are estimating that the delay in the camera is about 300~500 *ms*

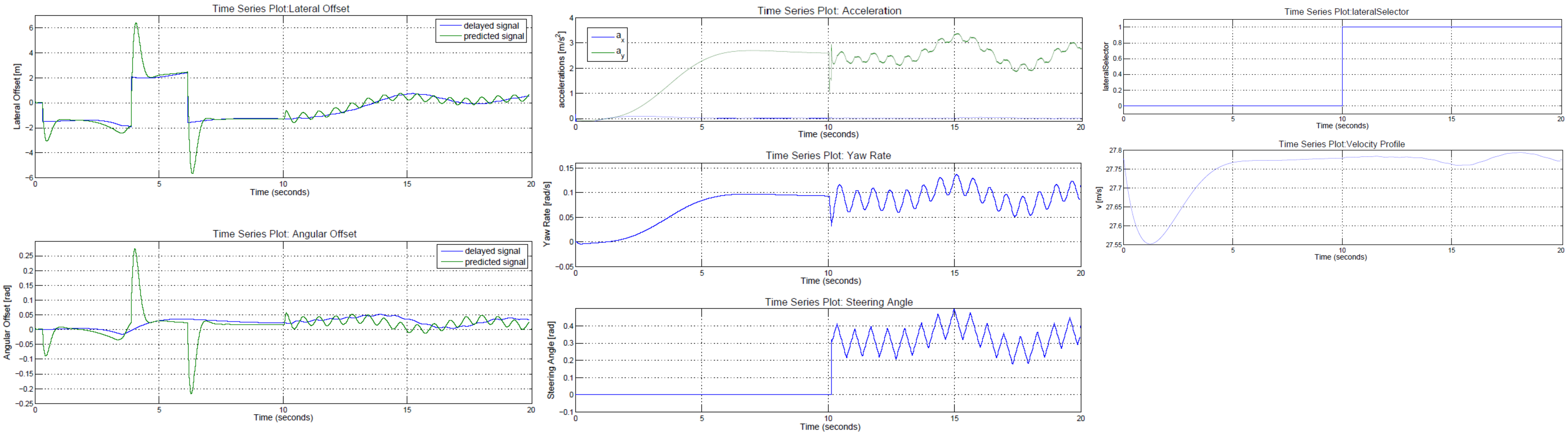


Diagnosis of the Problem via Simulation

- To verify the diagnoses, the same uncertainties can be implemented in the simulation environment
- Simulation in the CarMaker-Matlab/Simulink simulation environment with
  - 80 *ms* steering actuator delay
  - 300 *ms* position sensor (MobileEye 630) delay (providing lateral offset y_L and angular offset ε_L)

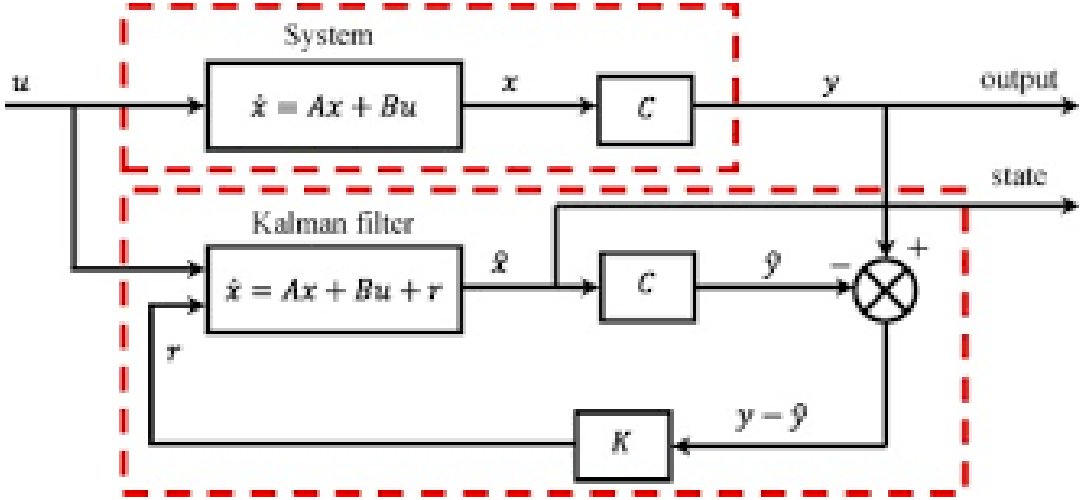
Diagnosis of the Problem via Simulation

- To verify the diagnoses, the same uncertainties can be implemented in the simulation environment
- Simulation in the CarMaker-Matlab/Simulink simulation environment with
  - 80 *ms* steering actuator delay
  - 300 *ms* position sensor (MobileEye 630) delay (providing lateral offset y_L and angular offset ε_L)



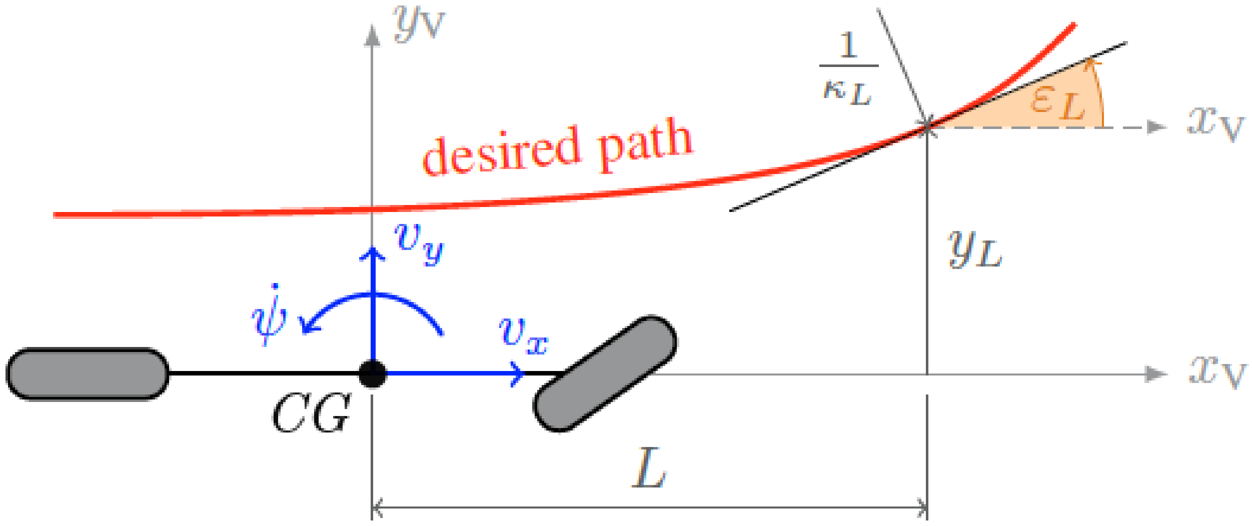
Kalman Filter Predictor based Mitigation

- In order to compensate the sensor and actuator delays, we designed a model based Kalman Filter predictor for the delayed position measurements of lateral and angular offsets: y_L, ε_L
- Kalman Filter as an optimal single step observer:



- Physical model: Kinematic Lane-keeping Model

$$\dot{y}_L = v_x \varepsilon_L - v_y - L\dot{\psi}$$
$$\dot{\varepsilon}_L = v_x \kappa_L - \dot{\psi}$$



Kalman Filter Predictor based Mitigation

virtual vehicle

- Discrete-time state space model for KF (based on the lateral tracking model)

$$\boldsymbol{x_{k+1}} = \boldsymbol{F_k x_k} + \boldsymbol{B_k u_k} \quad \text{(state Equation)}$$

$$\boldsymbol{z_k} = \boldsymbol{H_k x_k} \quad \text{(measurement Equation)}$$

$$F_k = \begin{bmatrix} 1 & v_x \Delta t \\ 0 & 1 \end{bmatrix} \qquad x_k = \begin{bmatrix} y_L \\ \varepsilon_L \end{bmatrix}$$

$$B_k = \begin{bmatrix} -\Delta t & -L\Delta t & 0 \\ 0 & -\Delta t & v_x \Delta t \end{bmatrix} \qquad u_k = \begin{bmatrix} v_y \\ \dot{\psi} \\ \kappa_L \end{bmatrix}$$

$$H_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Recursive Kalman Filter Equations

Predict

Predict	Predicted (*a priori*) state estimate	$\hat{\mathbf{x}}_{k\mid k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1\mid k-1} + \mathbf{B}_k \mathbf{u}_k$
	Predicted (*a priori*) error covariance	$\mathbf{P}_{k\mid k-1} = \mathbf{F}_k \mathbf{P}_{k-1\mid k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$
Update	Innovation or measurement pre-fit residual	$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k\mid k-1}$
	Innovation (or pre-fit residual) covariance	$\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k\mid k-1} \mathbf{H}_k^{\mathrm{T}}$
	Optimal Kalman gain	$\mathbf{K}_k = \mathbf{P}_{k\mid k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$
	Updated (*a posteriori*) state estimate	$\hat{\mathbf{x}}_{k\mid k} = \hat{\mathbf{x}}_{k\mid k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$

Kalman Filter Predictor based Mitigation

An a-priori *n*-step predictor :

- Compensates the delay of the sensor for more than 1 time step obtained from the Kalman Filter
- We assumed constant F_k **and** u_k

for $i = 1{:}n$

$$\hat{x}_k = F_k \hat{x}_{k+i-1} + B_k u_k$$

end

- The recursion is repeated *n* steps at every time instant after the Kalman Filter recursion
- *n* is dependent on the delay in the sensor data and the time step of the control algorithm

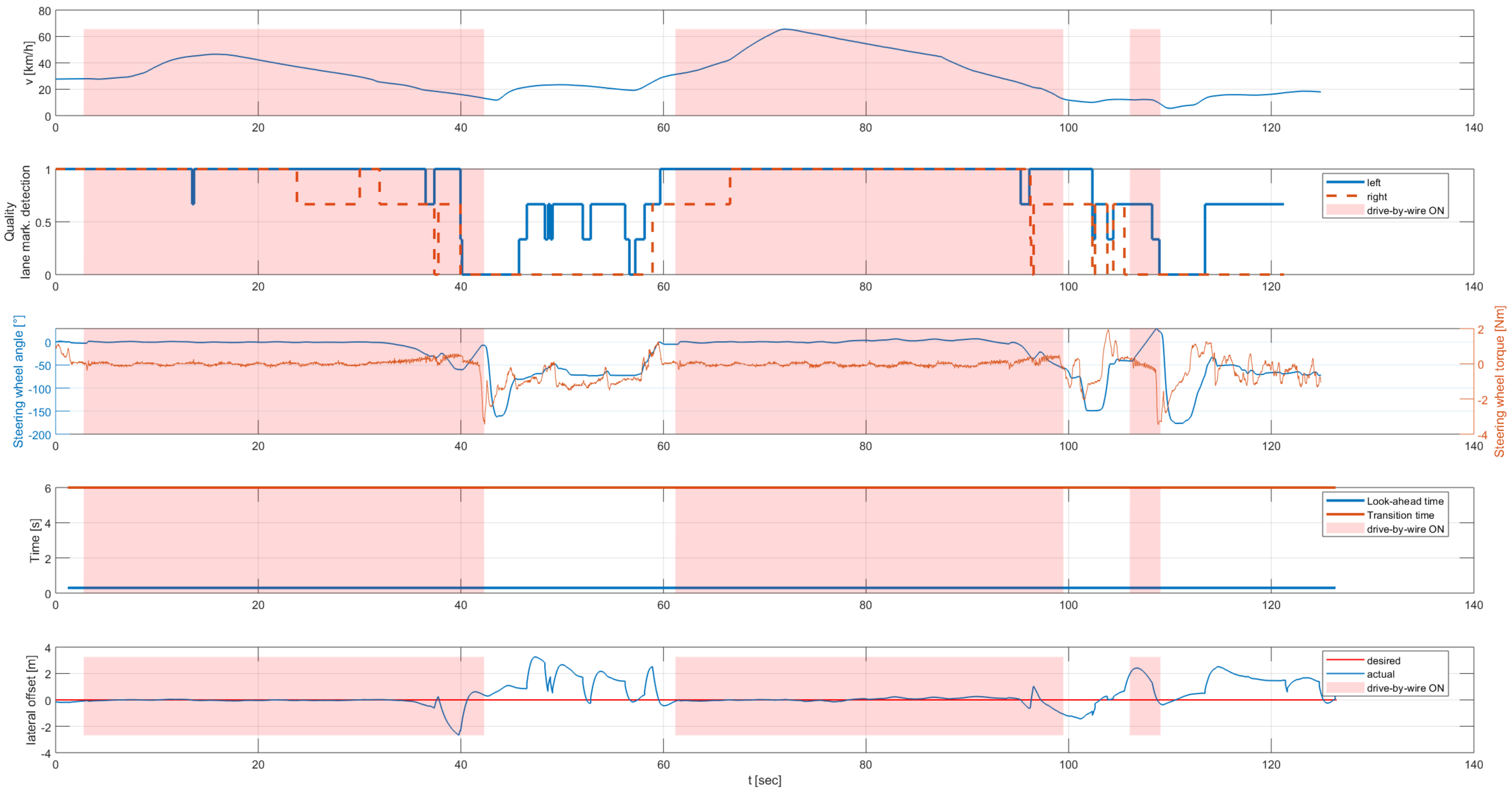
Simulative Comparison w/wo the Kalman Filter

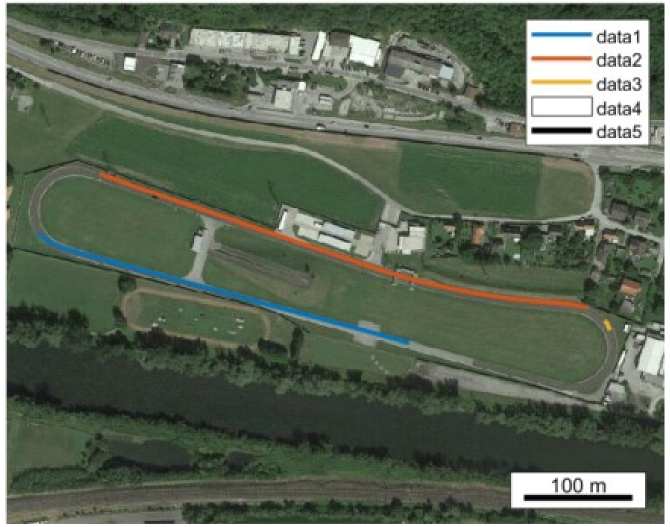
without the Kalman Filter Predictor

with the Kalman Filter Predictor

Simulative Comparison w/wo the Kalman Filter

Final Tests with the Enhanced LKA Function

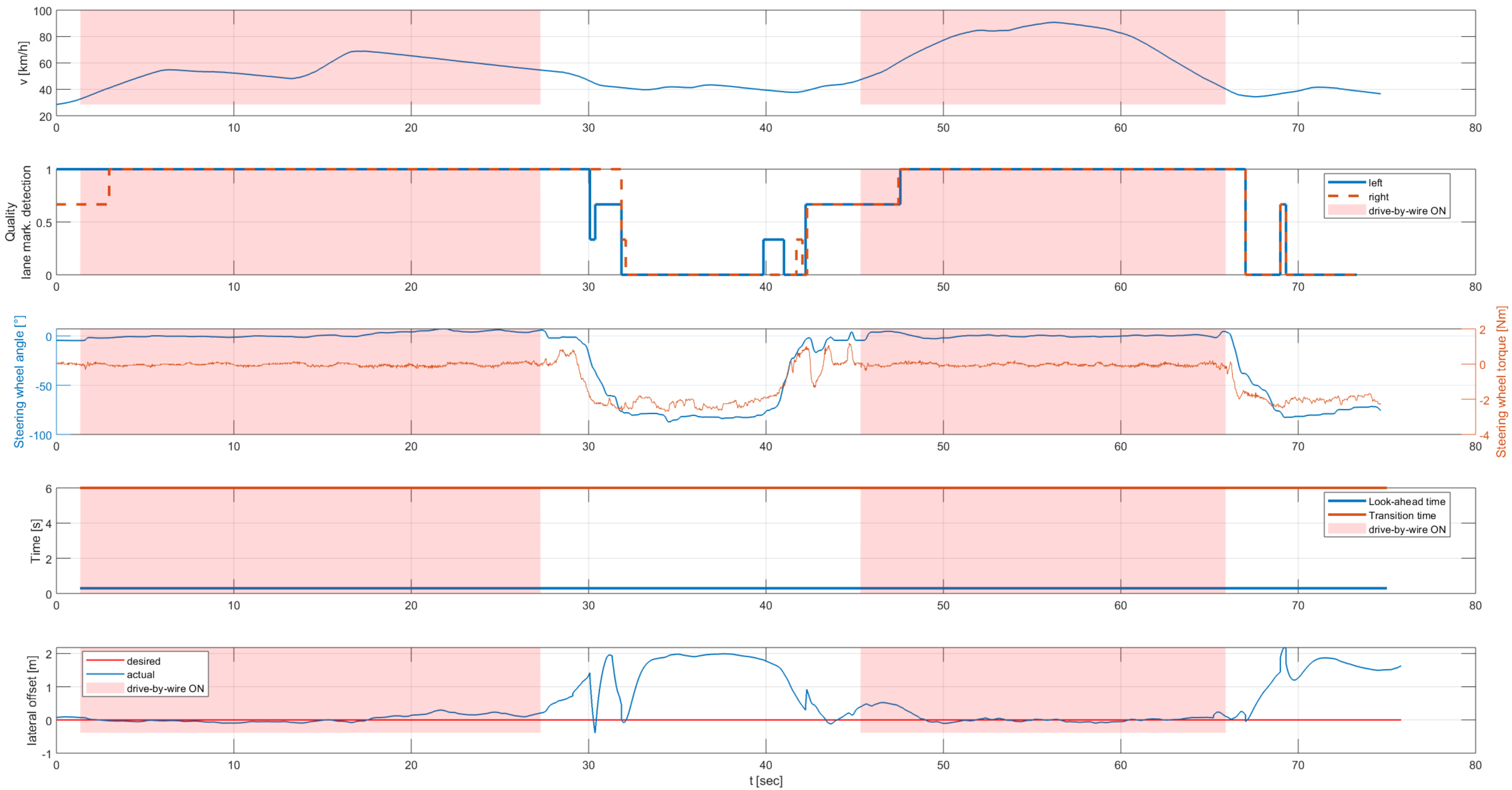


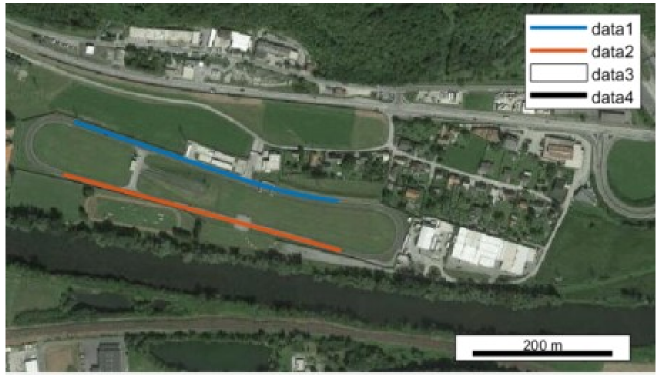


Our final LKA test in AVL test track in Gratkorn (26/11/2018)

"Test-1"

Final Tests with the Enhanced LKA Function





Our final LKA test in AVL test track in Gratkorn (26/11/2018)

“Test-2”

Final Tests with the Enhanced LKA Function

- Our final LKA test in AVL test track in Gratkorn (26/11/2018)
- "Test-2 Video"

Conclusions and Outlook

- MWC controllers were developed and tested based on a simulation environment
- Test track and test vehicle experience indicated that there are many differences between the simulation environment and the real-life implementations of the controllers
- We gained plenty of experience on how to detect the problems in the real vehicle implementations and how to address these issues
- Final state of the ADAS functions can be used for autonomous vehicle implementations for different classes of vehicles

virtual vehicle

Enabling future vehicle technologies

THANK YOU

Contact:

Dr. Selim Solmaz

Senior Researcher

Dept. E / Control Systems Group

VIRTUAL VEHICLE

selim.solmaz@v2c2.at

Tel.: +43 316 873 9730

https://www.v2c2.at/cooperation/referenzprojekte/

ICCVE2019 | November 5, 2019

VIRTUAL VEHICLE Research Center is funded within the COMET – Competence Centers for Excellent Technologies – programme by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT), the Federal Ministry for Digital and Economic Affairs (BMDW), the Austrian Research Promotion Agency (FFG), the province of Styria and the Styrian Business Promotion Agency (SFG). The COMET programme is administrated by FFG.