Automotive and Engine Technology (2022) 7:295–306
https://doi.org/10.1007/s41104-022-00114-8

ORIGINAL PAPER

Encapsulated trajectory tracking control for autonomous vehicles

Tobias Homolla[1] · **Hermann Winner**[1]

Received: 16 February 2022 / Accepted: 28 June 2022 / Published online: 25 July 2022

Abstract
The motion control of autonomous vehicles with a modular, service-oriented system architecture poses new challenges, as trajectory-planning and -execution are independent software functions. In this paper, requirements for an encapsulated trajectory tracking control are derived and it's shown that key differences to conventional vehicles with an integrated system architecture exist, requiring additional attention during controller design. A novel, encapsulated control architecture is presented that incorporates multiple extensions and support functions, fulfilling the derived requirements. It allows the application within the modular architecture without loss of functionality or performance. The controller considers vehicle stability and enables the yaw motion as an independent degree of freedom. The concept is applied and validated within the vehicles of the UNICAR*agil* research project, that feature the previously described system architecture to increase flexibility of application by dynamically interconnecting services based on the current use-case.

Keywords Motion control · Modularization · Autonomous driving

1 Introduction

The motion control of autonomous vehicles consists of the two sub-functions trajectory planning and its execution. Most established approaches treat the two functions as a unit and develop them in an integrated manner [1], but their separation has multiple advantages to consider. It enables the use of a modular, service-oriented system architecture and increases the reusability as well as updatability of software modules, e.g. for different operating modes employing distinct planning algorithms without having to change the control stage used as well. Furthermore, the two functions can be developed and tested independently during particular testing [2]. However, this separation introduces new challenges to achieve a satisfactory overall performance, specifically for the trajectory tracking control (TrTC) used to execute the planned behavior.

1.1 Research project UNICAR*agil*

Within the project UNICAR*agil*, a consortium of eight German universities and eight companies are working on driverless vehicles that are built strictly modular in hardware and software, maximizing the adaptability for different urban use-cases. They are built from a scalable platform that is complemented with different cabins based on the intended application [3]. The platforms possess electric single wheel actuators, enabling independent control commands for each wheel [4]. The service-oriented architecture allows different software-services to be dynamically interconnected based on criteria such as the current operating mode of the vehicle [5]. Additionally, independent testing and approval of services is researched [6]. To fully exploit the benefits of this architecture, independent trajectory planning and control services are required.

1.2 State of the art

Current TrTC frequently employ a high level stabilization approach, in which trajectory planning and control are integrated into a single module [1]. However, this approach can

Tobias Homolla and Hermann Winner have contributed equally to this work.

✉ Tobias Homolla
tobias.homolla@tu-darmstadt.de

Hermann Winner
hermann.winner@tu-darmstadt.de

[1] Fachgebiet Fahrzeugtechnik, Technische Universität Darmstadt, Otto-Berndt Str. 2, 64287 Darmstadt, Hessen, Germany

lead to higher complexity of the combined module and lacks the possibility of separate development and testing procedures. Additionally, high level stabilization approaches can face problems concerning disturbance rejection, especially in high latency systems [7].

In conventional powertrain architectures, kinematically constrained distributions between wheel steering angles are used [8], leading to coupled translation and yaw motion of the vehicle (non-holonomic system). The yaw angle therefore cannot be influenced independently from translatoric vehicle motion.

Recently, much research attention has been given to model predictive control approaches. These can lead to desirable controller performance, especially for non-holonomic vehicles using a conventional steering system, since a sequence of control commands is optimized [9]. However, results are highly dependent on model fidelity and the computational burden can impede real time capabilities [10].

To the best knowledge of the authors, no comparable investigation of the effects of encapsulation on the conception and design of a TrTC within a modular, service-oriented system architecture has been published.

2 Scope and paper structure

TrTC is a key link within the overall system architecture of autonomous vehicles, as it transforms planned behavior into real world motion. In this paper, it is investigated which requirements arise for an encapsulated TrTC within a modular, service-oriented architecture and what implications are expected for the design of a TrTC without loss of functionality compared to integrated approaches. The requirement analysis includes an examination of the effects of a separated trajectory planning and execution stage. Furthermore, it is investigated, which effect the use of single wheel actuators has on complexity and computation effort of the applied control algorithms. After the previous questions have been answered, a TrTC is proposed that incorporates multiple novel support functions to suppress negative effects resulting from the described architecture.

The paper is divided into four main parts: in Sect. 3, the system architecture of the UNICAR*agil* vehicles is presented in detail. This lays the foundation for all further considerations within this contribution. The following Sect. 4 deals with requirements that arise for application of an encapsulated TrTC within a service-oriented architecture and for the use with single wheel actuators. It will be shown that there are key differences to integrated approaches commonly applied in vehicles with a conventional powertrain. Furthermore, the relevant use-cases are described.

To the best knowledge of the authors, no controller has been published that fulfills all derived requirements. Therefore, in Sect. 5, it is investigated if and how a controller can satisfy all requirements and if the overall complexity of the controller can be reduced compared to established model predictive control approaches. A control architecture is introduced that takes the previously defined requirements into account while maintaining a low overall complexity. This TrTC is then implemented in IPG CarMaker simulation environment to verify its applicability for the relevant use-cases (Sect. 6).

It is not the scope of this paper to develop an optimal control law based on specific metrics or to span the full solution space for the defined requirements. It is shown, however, that an operational control architecture can be derived that fulfills the specific requirements within the architecture, which is by itself a novelty.

The nomenclature used within this paper is summarized in Table 2 within the appendix.

3 System architecture

The system architecture of the UNICAR*agil* vehicles is modular and consists of three main levels: “cerebrum”, “brain stem” and “spinal cord”, each containing independent hardware and software components. While the “cerebrum” contains, among other things, environment perception and behavior planning functions, the “brainstem” is tasked with the execution of the planned trajectory by generating setpoint commands for the actuators within the “spinal cord” [3].

3.1 Mechatronic architecture

The vehicles possess electric single wheel actuators, called dynamic modules, comprised out of a wheel-hub-drive, friction brake, and steering actuator with up to 90° wheel steering angle [4]. This mechatronic structure allows independent steering and traction commands for each wheel and therefore enables the vehicle yaw motion as a fully independent degree of freedom, if no steering angle limits are reached. Using this degree of freedom, unconventional maneuvers such as sideways parking or narrow turning are possible. Figure 1 shows a sideways parking maneuver which is performed by applying 90° wheel steering angle to all dynamics modules.

3.2 Software architecture

The service-oriented architecture consists of multiple software services fulfilling specific functions within the vehicle by performing calculations and exchanging data via predefined interfaces. In contrast to conventional architectures in the automotive industry, this centralization of software

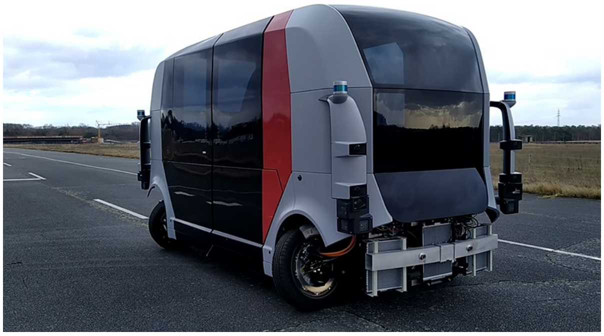

Fig. 1 Sideways parking of UNICAR*agil* vehicle "autoSHUTTLE" with 90° wheel steering angle on all wheels, utilizing the independent dynamics modules

and electronic control units aims at lowering the overall complexity, increase updatability and to prevent redundant implementations of functions (if not intended by the applied safety concept) [5].

There are different operating modes of the vehicle (i.e. autonomous driving, Safe Halt [6], or teleoperation [11]). For each mode, a different interconnection of services is necessary, which is implemented using a so called orchestrator, dynamically reconfiguring services based on current requirements. The combination of individual services make up the complete calculation pipeline for the vehicle behavior, according to the current operating mode. Communication between services is accomplished using the custom-built middleware ASOA (automotive service-oriented-software-architecture) [12].

4 Requirements

The previously described system architecture causes multiple challenges for the TrTC of such vehicles:

- One of the key benefits of the utilized service-oriented architecture is the capability to interconnect different services dynamically based on current requirements. Depending on the operating mode of the vehicle (automation, Safe Halt, or teleoperation), different services plan the target trajectory based on the intended vehicle behavior. Therefore, the TrTC needs to be independent from planning algorithms, resulting in a controller that shall be able to handle trajectories from all sources within the system without having to switch operating mode. Additionally, the resulting fundamental asynchrony of planning and control algorithms shall not have a negative effect on the TrTC performance.
- To uphold the overall modularization, information about a specific module shall only be distributed to other modules if absolutely necessary. For example, the trajectory planner shall not be customized to downstream modules such as the actuator architecture. If information sharing is unavoidable to prevent performance degradation, standardized interfaces shall be utilized, allowing to substitute modules at will.
- To avoid strict real time requirements for the computationally intensive planning services, latencies and jitter during trajectory planning shall not affect the controllers performance. This also enables using network technologies with no realtime guarantees for data transfer within the vehicle (e.g. Ethernet).
- Single wheel actuators introduce the vehicles yaw angle as a fully independent degree of freedom. The resulting three degrees of freedom (horizontal position + yaw angle) have to be controlled by the TrTC while allowing bi-directional driving. All kinematically possible locations for the vehicles center of rotation shall be enabled to be used, which includes maneuvers such as sideways parking with 90° vehicle sideslip angle (Fig. 1) and turning within narrow spaces.
- The independent dynamics modules causes the vehicle to be over-actuated. The resulting degrees of freedom for the wheel force distribution should be used for secondary goals (e.g. optimizing safety or energy demand) in a control allocation. To fully utilize the over-actuation, no predefined distribution strategy for the wheel steering angles shall be used but all necessary information shall be deducted from the current target trajectory.
- While ensuring the main function of trajectory tracking, the vehicle's stability has to be guaranteed under external disturbances and varying road conditions. This is equal to limiting the tire slip of each wheel to the stable part of its μ-slip-curve. The UNICAR*agil* vehicles are intended to be used within an urban environment (e.g. taxi or shuttle services). Operation close to the vehicle dynamic limits is hence not intended. However, during unexpected or emergency situations highly dynamic maneuvers are within the operational design domain.
- The execution of planned trajectories is subject to various constraints such as actuator-, friction- or kinematic-limits. Therefore, the TrTC needs to ensure that the independent planning stage only calculates trajectories that are physically feasible. Additionally, it has to be guaranteed that the controller will neither destabilize nor show unexpected behavior in case the execution limits were estimated wrongly and a non-feasible trajectory was planned.
- Trajectory planning and control both rely on localization data. In the contemplated service-oriented architecture, it is possible that planning and control each utilize a different and independent localization service. If these services estimate deviant poses of the vehicle, e.g. because of spe-

Table 1 Derived requirements for encapsulated TrTC within the system architecture described in Sect. 3

Req.	Description
R1	TrTC shall not need knowledge about or synchronization with the source of received trajectory (independence from planning stage). Additionally a dynamic reconfiguration of service interconnections shall be possible
R2	Only information shall be shared between modules via standardized interfaces that will lead to loss of performance if they are omitted
R3	Delay and jitter of the trajectory planning stage shall not influence TrTC performance and stability
R4	All three horizontal degrees of freedom shall be controlled independently and bi-directional driving shall be possible
R5	TrTC shall guarantee vehicle stability by limiting wheel slip to stable part of μ-slip-curve
R6	TrTC shall not rely on predefined distribution strategies for forces and wheel steering angles, but deduct all necessary information from the received target trajectory
R7	TrTC shall ensure that the trajectory planner only provides physically feasible trajectories and shall include an escalation mechanism in case of errors during estimation of execution limits
R8	No consideration of state history within the TrTC (holonomic system)
R9	Independent testing of planning and execution stage shall be possible
R10	Unwanted control action by inconsistent estimated vehicle poses shall be avoided
R11	Tracking error caused by earth curvature shall be avoided

cific sensor errors, the controller will try to compensate for this offset. It is therefore mandatory that the controller includes a mechanism that monitors the offset and guarantees that no control action will be derived from it [13]. To prevent growing tracking errors, it is also mandatory that the effects of earth curvature are compensated within the controller. Without compensation, the tracking error will increase nonlinearly with distance travelled and even comparably short distances (e. g. 50 km) can lead to unacceptable tracking errors for certain scenarios.

- Common model predictive control approaches are computationally intensive because a series of control actions is optimized, which leads to higher hardware demands and expenses on cooling and power supply. Due to the yaw motion being a fully independent degree of freedom when using single wheel actuators, the vehicle transforms into a holonomic system, hence no control input series needs to be considered to reach an intended vehicle state.[1] The controller shall therefore refrain from multi step optimization and register low computational demands, making it possible to run on microcontroller architecture and to widen the possible application field.[2]

The derived requirements are summarized in Table 1.

5 Controller architecture

After the question has been answered, which requirements arise for the TrTC within the considered system architecture, it is of interest, if and how a controller can fulfill all defined requirements.

5.1 Necessary support functions and extensions

To avoid any loss of performance and functionality, multiple extensions compared to conventional controllers need to be made.

5.1.1 Trajectory description

All planning services utilize an identical trajectory definition. The jth target trajectory $\mathcal{T}_j$ consists out of n time-stamped setpoint values for the pose $\mathcal{P}_{j,i}$ of the vehicle (horizontal position + yaw angle) and their first two time derivatives. The positions are given in the geodetic ETRS89 coordinate system, which considers earth curvature.

$$\mathcal{T}_j = \left\{ (t_{j,i}, \mathcal{P}_{j,i}, \dot{\mathcal{P}}_{j,i}, \ddot{\mathcal{P}}_{j,i})_{i=1..n} \right\} \tag{1}$$

The first point of a new trajectory $\mathcal{T}_{j+1,1}$ is located at the predicted position of the vehicle on the last target trajectory $\mathcal{T}_j$ to ensure smooth transitions. The trajectory additionally includes information about the expected slope of the road (longitudinal and lateral), that are utilized within feedforward control.

New target trajectories are cyclically sent to TrTC by planning services. By providing a significantly longer trajectory than the cycle time of the planner, it is ensured that the TrTC does not run out of target values in case of unexpected latency and jitter. In combination with an absolute

[1] The trajectory planner may utilize different strategies for the intended yaw motion of the vehicle, such as preventing a vehicle side-slip angle or constant yaw angle, and encode this behavior into the target trajectory.

[2] In contrast, computational demands for the trajectory planning might increase slightly, but as there are no hard realtime requirements for the planning service, different and more potent hardware may be used.

time reference for the containing target states, requirement R3 is fulfilled.

To enable independent development of planning and control algorithms, the planned trajectories are generally not changed due to control deviations (low-level-stabilization). As a fallback level for wrongly assumed acceleration limits of the vehicle, thresholds for the biggest acceptable control deviation are defined, triggering the re-planning of the target trajectory if exceeded (bi-level-stabilization [7]).

5.1.2 Pose offset correction

To resolve the problem caused by deviating vehicle poses, estimated by different localization functions, a pose offset correction mechanism is proposed [13]. The system monitors the offset between the pose from which a target trajectory was planned and the pose a second localization function estimated at the same time. It is assumed that any offset between the two functions is due to specific sensor errors and shall be neglected. The estimated offset is removed from all poses sent to the TrTC. Therefore, no control action will occur due to sensor errors between both functions, fulfilling requirement R10.

5.1.3 Estimation of execution limits

Due to the modular separation of planning and execution of trajectories, it has to be ensured that only physically feasible trajectories are planned. To do so, an independent support service is introduced that estimates dynamic and kinematic limits of the vehicle and communicates them to any service planning target trajectories (requirement R7) via a standardized interface. Hence, the planning stage can deduct all necessary information from this interface and does not need to be customized to a specific vehicle during development (requirement R2).

The estimation can be assisted by the single wheel actuators that allow active excitation of the powertrain. By applying differential torques between the wheels, a friction coefficient estimation is supported [14], but not subject of this publication.

5.1.4 Trajectory preprocessing

As described in Sect. 5.1.1, vehicle target positions are provided in geodetic coordinate system ETRS89. To be used for the calculation of control deviations within the TrTC, the positions in each new target trajectory are transformed into a local, Cartesian coordinate system in an additional support service. The origin of this coordinate system is located in the pose, each individual trajectory was planned on and the estimated actual vehicle position is transformed into the same system to ensure consistency. The cyclic coordinate transformation ensures that earth curvature will have no effect on controller performance, fulfilling requirement R11.

5.1.5 Stability control

According to requirement R5, the vehicle stability has to be ensured at all times. In conventional vehicles this requirement is fulfilled by three systems: traction control (TC), anti-lock braking system (ABS) and electronic stability control (ESC). Using single wheel actuators, these systems can be simplified and are therefore replaced by an integrated concept.

The vehicle yaw motion is an independent degree of freedom and directly controlled by the proposed control architecture, therefore there is no demand for a dedicated ESC algorithm.

Both TC and ABS keep the wheel slip within stable limits of the μ-slip-curve. In the proposed architecture they are replaced by dynamic wheel rotational limits that are calculated within the TrTC. These limits are then considered within the underlying electric drive control included in the single wheel actuators, reducing setpoint torques until the rotational speed limits are kept. Hence, requirement R5 is fulfilled.

5.2 Overall service architecture

The entire motion control service architecture is shown in Fig. 2. The displayed switches mark the possible reconfigurations by the ASOA orchestrator, based on the current vehicle operating mode.

During the autonomous driving operating mode of the vehicle, trajectories are planned by the trajectory planning service inside the "cerebrum" layer. The trajectories are planned based on estimated vehicle pose, provided by an independent localization function. Afterwards, the trajectories are sent to the "brainstem" layer and a coordinate transformation of the geodetic positions into local navigational coordinate system is performed during trajectory preprocessing, enabling controller design in a Cartesian system (Sect. 5.1.4).

The TrTC receives the trajectory and calculates setpoint values for the dynamic modules within the "spinal cord" layer, using the estimated vehicle state from the vehicle dynamics state estimation. The estimated state values undergo the previously described offset correction to prevent control deviation based on differences in pose estimation by the two localization functions. To calculate the pose offset, the current trajectory is needed, as it includes the pose on which it was planned on [13].

Trajectories can also be sent to TrTC from the control center during teleoperation, and from the Safe Halt function. The latter function transforms an emergency path, provided

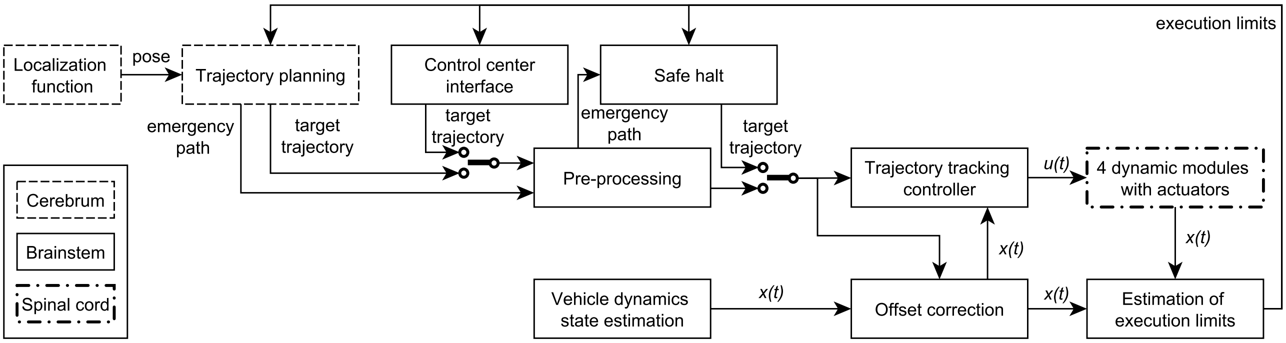

Fig. 2 Proposed motion control service architecture

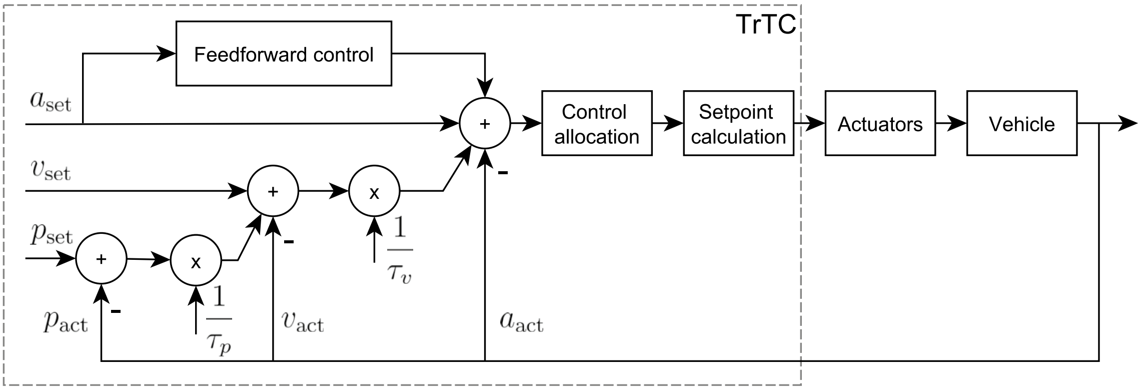

Fig. 3 Proposed controller architecture consisting out of a feedforward control and state feedback for each of the vehicle's three independent degrees of freedom

by the trajectory planner, into a trajectory after verifying it's unoccupied. Emergency paths are successively planned and in case of an error within the "cerebrum" layer, the last received path is used for the Safe Halt function. This ensures the vehicles ability to be transferred into a risk minimal state in case of degradations [6].

The TrTC needs no knowledge about the source of the trajectory as encoding doesn't change and interconnection is done solely by the orchestrator. Hence, the same service is used for all operating modes of the vehicle requiring a trajectory tracking functionality, fulfilling requirement R1.

The dynamic and kinematic execution limits are estimated within the separate service and then sent to all services planning target trajectories via a standardized interface. This ensures that every planning services can adapt the planned trajectory according to the vehicle's current capabilities.

5.3 Controller

According to requirements R4 and R6, the controller has to calculate wheel individual setpoint values for the single wheel actuators solely based on the received target trajectory and the current vehicle dynamics state, to guarantee the maximum application range for different trajectory planning algorithms. Since single wheel actuators transform the vehicle into a holonomic system, no consideration of past vehicle states is needed which allows immediate and independent correction of all three degrees of freedom. This fact is subsequently exploited to derive a control algorithm with significantly reduced complexity and computation effort compared to model predictive control approaches. Linear control techniques are applied because nonlinearities within the powertrain are compensated by underlying low-level actuator controllers within a cascaded structure.

5.3.1 Overall architecture

The TrTC is implemented with a two degree of freedom structure consisting out of a feedforward term and state feedback (Fig. 3) for each independent degree-of-freedom of the vehicle. The individual amounts are then summed for the total force demand $\boldsymbol{F}_{\mathrm{d}}$.

The feedforward control is based on setpoint acceleration values in the target trajectory to ensure high system dynamics and an open-loop realization of the target transient behavior. Linear state feedback accounts for model

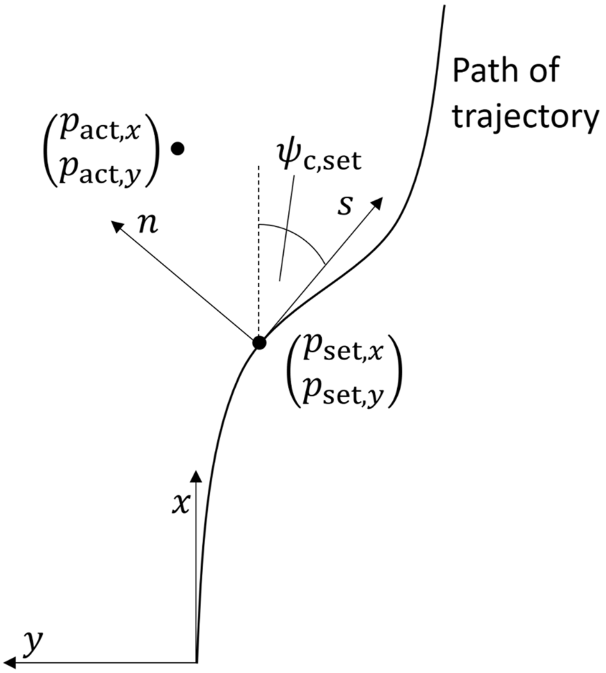
Path of trajectory
$\begin{pmatrix} p_{\mathrm{act},x} \\ p_{\mathrm{act},y} \end{pmatrix}$
$\psi_{\mathrm{c,set}}$
s
n
$\begin{pmatrix} p_{\mathrm{set},x} \\ p_{\mathrm{set},y} \end{pmatrix}$
x
y

Fig. 4 Frenet coordinate frame used in this paper. $p_{\mathrm{set},x}$ and $p_{\mathrm{set},y}$ are interpolated from the discrete target trajectory. $p_{\mathrm{act},x}$ and $p_{\mathrm{act},y}$ are not on the path of the trajectory, indicating a tracking error

uncertainties and external disturbances. Using different time constants on the feedback terms, the dynamics of each state can be manipulated.

The benefit of this architecture is the continuation of the modular approach. Feedforward, state feedback, control allocation and setpoint calculation can each be developed independently and replaced by different algorithms, without the need to change other parts of the controller (while keeping overall stability of the closed loop under consideration). Requirement R4 and R8 are thus fulfilled.

5.3.2 Force demand

The regarded vehicle is over-actuated due to the single wheel actuators. This property poses a challenge during the distribution of the overall force demand to the individual actuators (control allocation). To solve this issue, the setpoint values for position, velocity and acceleration are transformed into Frenet coordinate system (denoted by F) defined by setpoint velocity along the path of the planned trajectory (see Fig. 4). This transformation allows to separate subsets of the overall force demand that are applied by a specific actuator (e.g. steering or electric drive). The longitudinal force demand in Frenet coordinates is provided by the drive or braking actuators, while the lateral force demand is provided by steering actuators. The translational force demand ${}_{\mathrm{F}}\boldsymbol{F}_{\mathrm{d,t}}$ is defined by:

$$
\begin{aligned}
{}_{\mathrm{F}}\boldsymbol{F}_{\mathrm{d,t}} &= \begin{bmatrix} F_{\mathrm{d},s,\mathrm{t}} \\ F_{\mathrm{d},n,\mathrm{t}} \end{bmatrix} \\
&= \begin{bmatrix} \left(\frac{\frac{p_{\mathrm{set},s} - p_{\mathrm{act},s}}{\tau_p} + v_{\mathrm{set},s} - v_{\mathrm{act},s}}{\tau_v}) + a_{\mathrm{set},s} - a_{\mathrm{act},s} \right) \cdot m_{\mathrm{v}} + a_{\mathrm{set},s} \cdot m_{\mathrm{v}} \\ \left(\frac{\frac{p_{\mathrm{set},n} - p_{\mathrm{act},n}}{\tau_p} + v_{\mathrm{set},n} - v_{\mathrm{act},n}}{\tau_v}) + a_{\mathrm{set},n} - a_{\mathrm{act},n} \right) \cdot m_{\mathrm{v}} + a_{\mathrm{set},n} \cdot m_{\mathrm{v}} \end{bmatrix}
\end{aligned}
\tag{2}
$$

where set stands for setpoint values extracted from the target trajectory and s respectively n are the Frenet coordinate axes. It is important to note, that $a_{\mathrm{set},n}$ is not defined by $\dot{v}_{\mathrm{set},n}$ but calculated independently within the trajectory planner.

A rotation matrix $\boldsymbol{R}$ is calculated for the transformation of positions, velocities and accelerations from Cartesian to Frenet coordinate system:

$$
\boldsymbol{R} = \begin{bmatrix} \cos(\psi_{\mathrm{c,set}}) & -\sin(\psi_{\mathrm{c,set}}) \\ \sin(\psi_{\mathrm{c,set}}) & \cos(\psi_{\mathrm{c,set}}) \end{bmatrix} \tag{3}
$$

Using $\boldsymbol{R}$, the necessary transformations from Cartesian to Frenet coordinate frame are given by:

$$
\begin{bmatrix} p_{\mathrm{act},s} \\ p_{\mathrm{act},n} \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} p_{\mathrm{act},x} - p_{\mathrm{set},x} \\ p_{\mathrm{act},y} - p_{\mathrm{set},y} \end{bmatrix} \tag{4}
$$

$$
\begin{bmatrix} v_{\mathrm{act},s} \\ v_{\mathrm{act},n} \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} v_{\mathrm{act},x} \\ v_{\mathrm{act},y} \end{bmatrix} \tag{5}
$$

$$
\begin{bmatrix} a_{\mathrm{act},s} \\ a_{\mathrm{act},n} \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} a_{\mathrm{act},x} \\ a_{\mathrm{act},y} \end{bmatrix} \tag{6}
$$

$$
\begin{bmatrix} p_{\mathrm{set},s} \\ p_{\mathrm{set},n} \end{bmatrix} = 0 \tag{7}
$$

$$
\begin{bmatrix} v_{\mathrm{set},s} \\ v_{\mathrm{set},n} \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} v_{\mathrm{set},x} \\ v_{\mathrm{set},y} \end{bmatrix} = \begin{bmatrix} \cos(\psi_{\mathrm{c,set}}) \cdot v_{\mathrm{set},x} - \sin(\psi_{\mathrm{c,set}}) \cdot v_{\mathrm{set},y} \\ 0 \end{bmatrix} \tag{8}
$$

$$
\begin{bmatrix} a_{\mathrm{set},s} \\ a_{\mathrm{set},n} \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} a_{\mathrm{set},x} \\ a_{\mathrm{set},y} \end{bmatrix} \tag{9}
$$

Based on the estimated translational force demand on the vehicle, the over-actuated vehicle structure demands a resolution strategy, calculating a force demand for each wheel (control allocation). Different strategies can be applied, e.g. minimization of used friction potential or energy demand

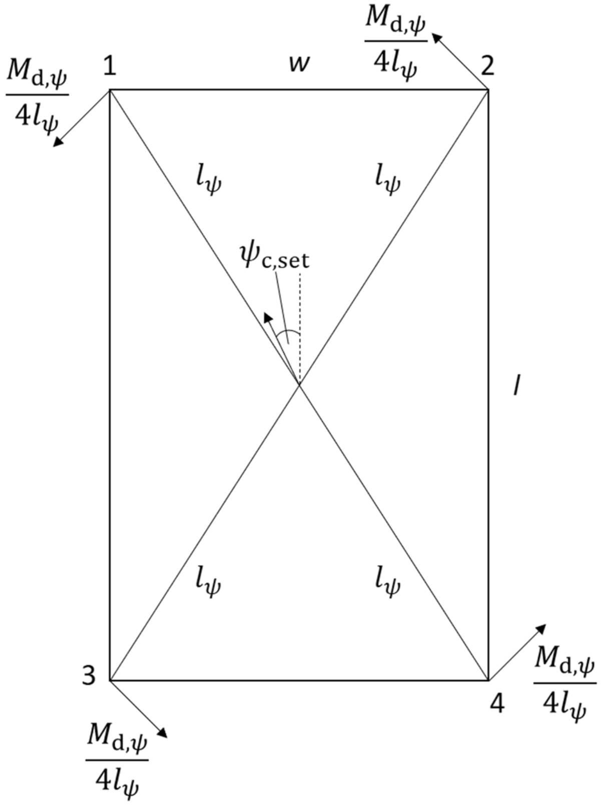
$\frac{M_{d,\psi}}{4l_\psi}$
1
w
$\frac{M_{d,\psi}}{4l_\psi}$
2
l_ψ
l_ψ
$\psi_{c,set}$
l
l_ψ
l_ψ
3
$\frac{M_{d,\psi}}{4l_\psi}$
4
$\frac{M_{d,\psi}}{4l_\psi}$

Fig. 5 Kinematic relations for the calculation of forces needed for yaw motion. The decomposition of $M_{\mathrm{d},\psi}$ maximizes the lever arm l_ψ to the vehicle center

[15–17], resulting in the translational force demand ${}_{\mathrm{F}}\boldsymbol{F}_{\mathrm{d,t},i}$ for each wheel $i \in 1..4$.

In addition to the translational vehicle motion, the yaw motion is controlled by a third independent control loop. Yaw torque demand $M_{\mathrm{d},\psi}$ is defined by:

$$M_{\mathrm{d},\psi} = \left(\frac{\frac{p_{\mathrm{set},\psi} - p_{\mathrm{act},\psi}}{\tau_p} + v_{\mathrm{set},\psi} - v_{\mathrm{act},\psi}}{\tau_v} + a_{\mathrm{set},\psi} - a_{\mathrm{act},\psi} \right) \cdot \Theta_\psi + a_{\mathrm{set},\psi} \cdot \Theta_\Psi \tag{10}$$

The result is then transformed into an additional force demand for each wheel ${}_{\mathrm{F}}\boldsymbol{F}_{\mathrm{d},\psi,i}$ by disassembling $M_{\mathrm{d},\psi}$ into force components with maximum lever arm to vehicle center (see Fig. 5).

The total force demand for for each wheel in longitudinal and lateral direction is then given by:

$${}_{\mathrm{F}}\boldsymbol{F}_{\mathrm{d},i} = \begin{bmatrix} F_{\mathrm{d},s,i} \\ F_{\mathrm{d},n,i} \end{bmatrix} = \begin{bmatrix} F_{\mathrm{d},s,\mathrm{t},i} + F_{\mathrm{d},s,\psi,i} \\ F_{\mathrm{d},n,\mathrm{t},i} + F_{\mathrm{d},n,\psi,i} \end{bmatrix} \tag{11}$$

with

$$\begin{bmatrix} F_{\mathrm{d},s,\psi,1} \\ F_{\mathrm{d},n,\psi,1} \\ F_{\mathrm{d},s,\psi,2} \\ F_{\mathrm{d},n,\psi,2} \\ F_{\mathrm{d},s,\psi,3} \\ F_{\mathrm{d},n,\psi,3} \\ F_{\mathrm{d},s,\psi,4} \\ F_{\mathrm{d},n,\psi,4} \end{bmatrix} = \begin{bmatrix} -\cos(\arctan(\frac{l}{w}) + \psi_{\mathrm{c,set}}) \\ \sin(\arctan(\frac{l}{w}) + \psi_{\mathrm{c,set}}) \\ \sin(\arctan(\frac{w}{l}) + \psi_{\mathrm{c,set}}) \\ \cos(\arctan(\frac{w}{l}) + \psi_{\mathrm{c,set}}) \\ -\sin(\arctan(\frac{w}{l}) + \psi_{\mathrm{c,set}}) \\ -\cos(\arctan(\frac{w}{l}) + \psi_{\mathrm{c,set}}) \\ \cos(\arctan(\frac{l}{w}) + \psi_{\mathrm{c,set}}) \\ -\sin(\arctan(\frac{l}{w}) + \psi_{\mathrm{c,set}}) \end{bmatrix} \cdot \frac{M_{\mathrm{d},\psi}}{4 \cdot l_\psi} \tag{12}$$

in which l is the wheelbase, w the track width, and l_ψ the distance between vehicle center and wheel contact point. The wheels are numbered in order front-left, front-right, rear-left, rear-right. The kinematic relations are displayed in Fig. 5.

Equation 12 assumes an equal distribution of forces to all wheels for the application of the desired yaw torque. However, any distribution guaranteeing an identical sum of longitudinal and lateral forces over all wheel contact points is possible. This fact can be utilized for the compensation of actuator degradations or to ensure vehicle stability when driving close to friction limits.

5.3.3 Setpoint value calculation

The force demands are then transformed into setpoint values for each single wheel actuator $\boldsymbol{u}_i$ which consists out of the wheel steering angle $\delta_{\mathrm{d},i}$ and the target wheel torque $M_{\mathrm{d},i}$:

$$\boldsymbol{u}_i = \begin{bmatrix} M_{\mathrm{d},i} \\ \delta_{\mathrm{d},i} \end{bmatrix} = \begin{bmatrix} F_{\mathrm{d},s,i} \cdot r_{\mathrm{dyn}} \\ \frac{F_{\mathrm{d},n,i}}{c_\alpha} + \psi_{\mathrm{c,set},i} \end{bmatrix} \tag{13}$$

in which c_α is an estimated cornering stiffness of the tire. The current target speed direction at the wheel contact point $\psi_{\mathrm{c,set},i}$ is calculated from the target trajectory by:

$$\begin{bmatrix} \psi_{\mathrm{c,set},1} \\ \psi_{\mathrm{c,set},2} \\ \psi_{\mathrm{c,set},3} \\ \psi_{\mathrm{c,set},4} \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{{}_{\mathrm{V}}v_{\mathrm{set},x} - v_{\mathrm{set},\psi} \cdot \frac{w}{2}}{{}_{\mathrm{V}}v_{\mathrm{set},y} + v_{\mathrm{set},\psi} \cdot \frac{l}{2}} \right) \\ \arctan\left(\frac{{}_{\mathrm{V}}v_{\mathrm{set},x} + v_{\mathrm{set},\psi} \cdot \frac{w}{2}}{{}_{\mathrm{V}}v_{\mathrm{set},y} + v_{\mathrm{set},\psi} \cdot \frac{l}{2}} \right) \\ \arctan\left(\frac{{}_{\mathrm{V}}v_{\mathrm{set},x} - v_{\mathrm{set},\psi} \cdot \frac{w}{2}}{{}_{\mathrm{V}}v_{\mathrm{set},y} - v_{\mathrm{set},\psi} \cdot \frac{l}{2}} \right) \\ \arctan\left(\frac{{}_{\mathrm{V}}v_{\mathrm{set},x} + v_{\mathrm{set},\psi} \cdot \frac{w}{2}}{{}_{\mathrm{V}}v_{\mathrm{set},y} - v_{\mathrm{set},\psi} \cdot \frac{l}{2}} \right) \end{bmatrix} \tag{14}$$

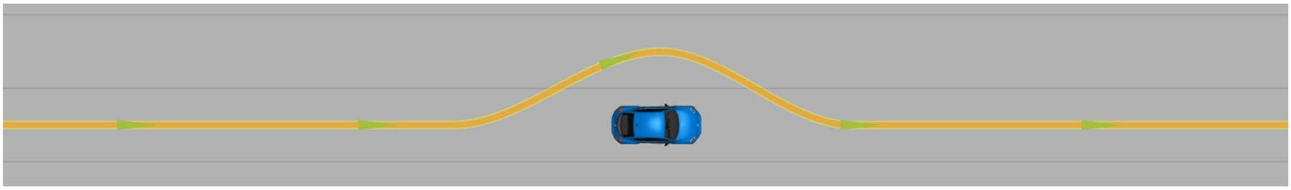

Fig. 6 Double lane change maneuver used for verification of the implemented control concept. Yellow line represents the translational path of the target trajectory used

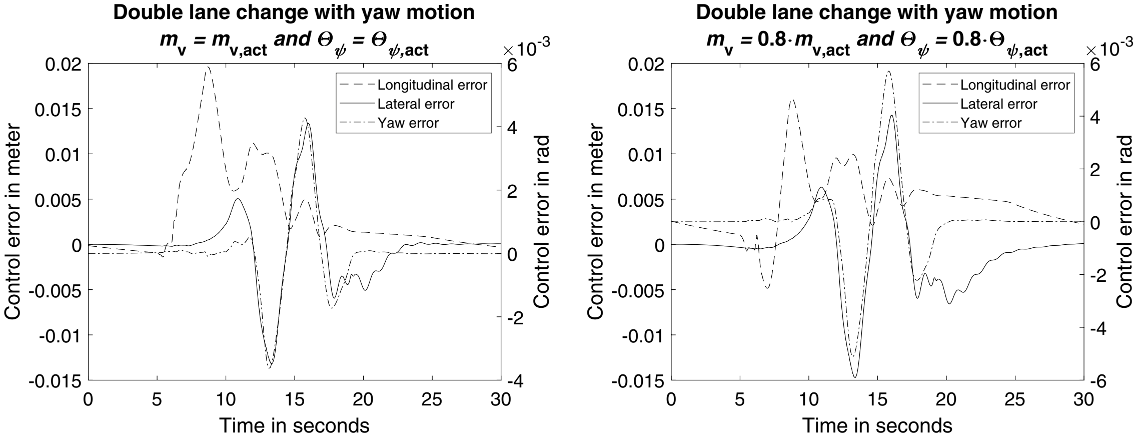

Fig. 7 Tracking error for all three independent degrees of freedom during double lane change maneuver with $\beta_{\mathrm{set}} = 0$

with ${}_{\mathrm{v}}v_{\mathrm{set},x}$ and ${}_{\mathrm{v}}v_{\mathrm{set},y}$ representing the setpoint speed in the vehicle coordinate system defined by [18]. To ensure fast correction of control errors, ${}_{\mathrm{v}}v_{\mathrm{set},y}$ and $v_{\mathrm{set},\psi}$ include a superposed term based on the respective control deviation that create a kinematic path back onto the current target trajectory.

Distribution of $M_{\mathrm{d},i}$ to drive and brake actuators are done based on current driving direction and whether specific maneuvers (e.g. sideways parking) are intended. The setpoint values $\boldsymbol{u}_i$ are executed by low-level controllers in each single wheel actuator, which are not subject of this paper.

6 Implementation

The concept is implemented in IPG CarMaker simulation environment which is extended with verified dynamic models for all single wheel actuators and therefore enables full support for the proposed control concept. For the defined use-case of urban driving, a representative test case is considered (Fig. 6). The double lane change around a parking vehicle is chosen since it represents typical maneuvering during city driving and combines longitudinal and lateral acceleration. It is performed twice, once with the goal of no vehicle sideslip angle ($\beta_{\mathrm{set}} = 0$) and once without yaw motion ($p_{\psi,\mathrm{set}} = v_{\psi,\mathrm{set}} = a_{\psi,\mathrm{set}} = 0$). The tests are not conducted at friction limits since they aren't expected during normal driving operation within the use-case, hence an initial speed of 50 $\frac{\mathrm{km}}{\mathrm{h}}$ and max. acceleration values of 1.5 $\frac{\mathrm{m}}{\mathrm{s}^2}$ (longitudinal and lateral) are chosen to ensure passenger comfort. To verify fulfillment of requirement R3, a random latency between 0 and 100 ms is artificially added to each trajectory that is sent to the TrTC. Equations 2 and 10 include an estimated vehicle mass (m_{v}) and moment of inertia (Θ_ψ), that are not always precisely known in a real world application. Hence, both lane change maneuvers are also performed with an assumed vehicle mass and moment of inertia that is 20% below the actual value.

To perform the simulation, the controller has to be parametrized by defining the different time constants for the state feedback. Since the transient behavior is executed by the independent feedforward control, the state feedback can utilize a stiff parametrization to account for model uncertainties and external disturbances. Thus, the stiff values $\tau_p = 0.28$ s and $\tau_v = 0.07$ s are chosen.

During maneuvering through city traffic, position errors are considered to be critical and are therefore primarily analyzed, as they are directly responsible for a potential collision. The tracking error for each simulation consists out of

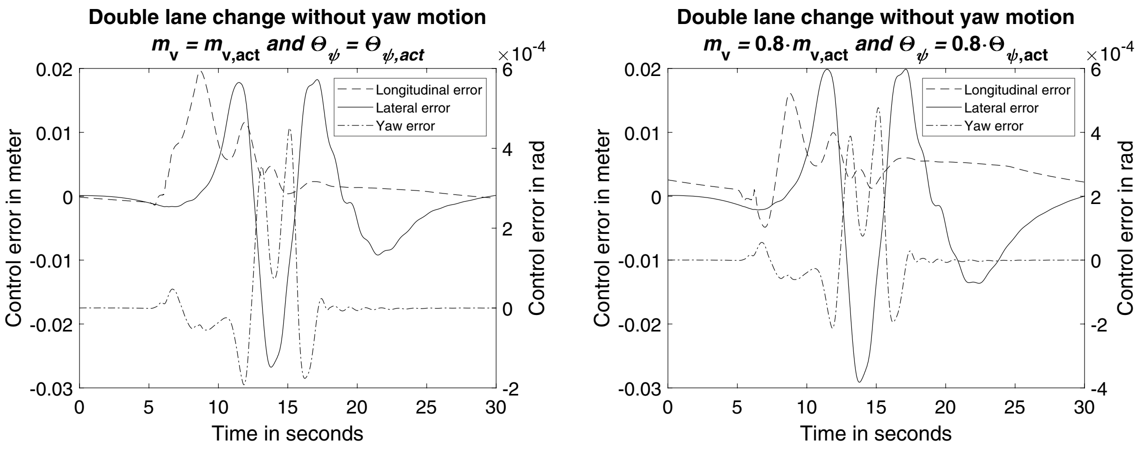

Fig. 8 Tracking error for all three independent degrees of freedom during double lane change maneuver with $p_{\psi,\text{set}} = v_{\psi,\text{set}} = a_{\psi,\text{set}} = 0$

the horizontal position error in both Frenet axis directions and a yaw angle error. The results are shown in Figs. 7 and 8.

It is shown that the vehicle is able to follow the target trajectory and that translational and yaw motion of the vehicle can be manipulated independently (requirement R4). Different steering strategies are deducted solely from the target trajectory, which fulfills requirement R6. The added artificial latency and jitter does not negatively affect the tracking accuracy (requirement R3), as identical results are obtained without it. Furthermore, the tracking error is well below the required accuracy for driving on local roads [19], which is the objective within the project UNICAR*agil*. This is also true with an underestimated vehicle mass and moment of inertia, which leads to an expected slight increase of tracking errors. This fact demonstrates the robustness of the controller towards a typical parameter uncertainty in real world application. To obtain these results, no consideration of state history in a multi step optimization was necessary, verifying requirement R8. The constants τ_p and τ_v allow the two goals of tracking accuracy and passenger comfort to be weighed against each other and an application-specific compromise to be found.

7 Conclusion

This paper offers multiple contributions to the TrTC of autonomous vehicles.

Firstly, it has been shown which requirements arise from the application of an encapsulated TrTC within a modular service-oriented system architecture and what impact single wheel actuators have on the controllers task. The analysis shows that well established control concepts cannot be adopted without significant changes and the introduction of support functions. The high complexity of these concepts (caused by the integration of vehicle state history) is not needed when using single wheel actuators due to the holonomic system property.

Secondly, the question whether a control architecture can fulfill all defined requirements has been answered by proposing and implementing a novel architecture. The previously identified required extensions and support functions are conceptually designed and implemented, to ensure that no loss of performance and functionality occurs. The controller is capable of tracking a target trajectory in all three independent degrees of freedom of the vehicle, while utilizing the single wheel actuators for secondary objectives and guaranteeing low demands on computational effort. The concept does not need a predefined distribution of torques and wheel steering angles, as all necessary information can be derived from setpoint values in the target trajectory. Trajectories from multiple sources within the service-oriented environment can be handled by the developed control architecture while not needing any information about the overall vehicle operation mode.

By presenting the aforementioned TrTC architecture, it was confirmed that a separation of trajectory-planning and execution stage can be achieved without loss of performance or functionality, allowing the use of particular testing methods for both stages, significantly increasing flexibility and updatability within autonomous vehicles. The contributions in this paper extend the applicability of TrTC to modular system architectures. Thus, cross-domain objectives such as the service-orientation and dynamic reconfiguration of functions are no longer impeded by the TrTC, which transforms planned behavior into real world motion.

The control approach is currently being integrated in all four research vehicles of the UNICAR*agil* project using microcontroller embedded hardware and therefore applied within the described system architecture. Real-world validation experiments for the control architecture are performed.

Certain aspects of the overall concept will be subject of further research, such as the control allocation strategy, the estimation of execution limits and rotational speed limits for stability control, or the compensation of actuator degradations. Additional starting points for research based on this paper may include an evaluation of the subjective passenger comfort during test drives as well as controller extensions for highly dynamic driving close to friction limits.

Appendix A: Nomenclature

See Table 2.

Table 2 Nomenclature used in this paper

Symbol	Description	Index	Description
a	Acceleration	d	Demand
c_α	Cornering stiffness	F	Frenet coordinate system
F	Force	i, j	Running variables
$\mathcal{P}$	Pose	s, n	Frenet coordinate axes
p	Position	x, y, z	Cartesian coordinate axes
r_{dyn}	Dynamic wheel radius	set	Setpoint
$\mathcal{T}$	Trajectory	act	Actual
t	Time	p	Position
u	Setpoint commands	t	Translational
v	Velocity	v	Velocity
x	State	v	Vehicle coordinate system
δ	Wheel steering angle	ψ	Yaw degree of freedom
Θ	Moment of inertia		
μ	Friction coefficient tire-road		
ψ_{c}	Course angle		
τ	Time constant		
β	Vehicle sideslip angle		
n	Total number		
$\boldsymbol{R}$	Rotation matrix		
m_{v}	Vehicle mass		
M	Torque		
l	Wheelbase		
w	Track width		
l_ψ	Distance between vehicle center and wheel contact point		

Acknowledgements This research is accomplished within the project "UNICAR*agil*" (FKZ 16EMO0286). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Rupp, A., Stolz, M.: Survey on control schemes for automated driving on highways. In: Watzenig, D., Horn, M. (eds.) Automated Driving, pp. 43–69. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-31895-0_4
2. Klamann, B., Lippert, M., Amersbach, C., Winner, H.: Defining pass-/fail-criteria for particular tests of automated driving functions. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 169–174. IEEE, Auckland (2019). https://doi.org/10.1109/ITSC.2019.8917483
3. Woopen, T., Lampe, B., Böddeker, T., Eckstein, L., Kampmann, A., Alrifaee, B., Kowalewski, S., Moormann, D., Stolte, T., Jatzkowski, I., Maurer, M., Möstl, M., Ernst, R., Ackermann, S., Amersbach, C., Leinen, S., Winner, H., Püllen, D., Katzenbeisser, S., Becker, M., Stiller, C., Furmans, K., Bengler, K., Diermeyer, F., Lienkamp, M., Keilhoff, D., Reuss, H.-C., Buchholz, M., Dietmayer, K., Lategahn, H., Siepenkötter, N., Elbs, M., v. Hinüber, E., Dupuis, M., Hecker, C.: UNICARagil—disruptive modular architectures for agile, automated vehicle concepts. In: 27th Aachen Colloquium, Aachen (2018)
4. Martens, T., Pouansi Majiade, L., Li, M., Henkel, N., Eckstein, L., Wielgos, S., Schlupek, M.: UNICARagil dynamics module. In: 29th Aachen Colloquium Sustainable Mobility, Aachen (2020)
5. Mokhtarian, A., Alrifaee, B., Kampmann, A.: The dynamic service-oriented software architecture for the UNICARagil Project. In: 29. Aachen Colloquium Sustainable Mobility 2020, Aachen. https://doi.org/10.18154/RWTH-2020-11256
6. Stolte, T., Graubohm, R., Jatzkowski, I., Maurer, M., Ackermann, S., Klamann, B., Lippert, M., Winner, H.: Towards safety concepts for automated vehicles by the example of the project UNICARagil. In:

29th Aachen Colloquium Sustainable Mobility 2020, 5.–7. Oktober 2020 (2020)

7. Werling, M.: Ein Neues Konzept Für die Trajektoriengenerierungu-Und -Stabilisierung in Zeitkritischen Verkehrsszenarien. KIT Scientific Publishing, Karlsruhe (2011). https://doi.org/10.5445/KSP/1000021738
8. Herold, M.: Redundant steering system for highly automated driving of trucks. PhD thesis, Technische Universität, Darmstadt (2019). https://doi.org/10.25534/tuprints-00009458
9. Nolte, M., Rose, M., Stolte, T., Maurer, M.: Model predictive control based trajectory generation for autonomous vehicles—an architectural approach. In: 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 798–805. IEEE, Los Angeles (2017). https://doi.org/10.1109/IVS.2017.7995814
10. Sorniotti, A., Barber, P., De Pinto, S.: Path tracking for automated driving: a tutorial on control system formulations and ongoing research. In: Watzenig, D., Horn, M. (eds.) Automated Driving, pp. 71–140. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-31895-0_5
11. Feiler, J., Hoffmann, S., Diermeyer, D.F.: Concept of a control center for an automated vehicle fleet. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. IEEE, Rhodes (2020). https://doi.org/10.1109/ITSC45102.2020.9294411
12. Kampmann, A., Alrifaee, B., Kohout, M., Wustenberg, A., Woopen, T., Nolte, M., Eckstein, L., Kowalewski, S.: A dynamic service-oriented software architecture for highly automated vehicles. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2101–2108. IEEE, Auckland (2019). https://doi.org/10.1109/ITSC.2019.8916841
13. Homolla, T., Gottschalg, G., Winner, H.: Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten in modularen technischen Systemen. In: Uni-DAS 13. Workshop Fahrerassistenz und Automatisiertes Fahren. FAS 2020, p. 12. Uni-DAS e.V., Darmstadt (2021). https://doi.org/10.26083/tuprints-00018522
14. Chen, Y., Wang, J.: Vehicle-longitudinal-motion-independent realtime tire-road friction coefficient estimation. In: 49th IEEE Conference on Decision and Control (CDC), pp. 2910–2915. IEEE, Atlanta (2010). https://doi.org/10.1109/CDC.2010.5717437
15. Johansen, T.A., Fossen, T.I.: Control allocation—a survey. Automatica **49**(5), 1087–1103 (2013). https://doi.org/10.1016/j.automatica.2013.01.035
16. Moseberg, J.-E.: Regelung der Horizontalbewegung Eines überaktuierten Fahrzeugs Unter Berücksichtigung Von Realisierungsanforderungen. FAU-Studien aus der Elektrotechnik, vol. Band 4. FAU University Press, Erlangen (2016)
17. Huang, Y., Wang, F., Li, A., Shi, Y., Chen, Y.: Development and performance enhancement of an overactuated autonomous ground vehicle. IEEE/ASME Trans. Mechatron. **26**(1), 33–44 (2021). https://doi.org/10.1109/TMECH.2020.2998454
18. DIN.: DIN ISO 8855:2013-11, Straßenfahrzeuge_- Fahrzeugdynamik und Fahrverhalten_- Begriffe (ISO_8855:2011). Technical report, Beuth Verlag GmbH. https://doi.org/10.31030/1941260
19. Reid, T.G.R., Houts, S.E., Cammarata, R., Mills, G., Agarwal, S., Vora, A., Pandey, G.: Localization requirements for autonomous vehicles. SAE Int. J. Connect. Autom. Veh. **2**(3), 12–02030012 (2019). https://doi.org/10.4271/12-02-03-0012