

Advanced Path Following Control of an Overactuated Robotic Vehicle

Peter Ritzer, Christoph Winter and Jonathan Brembeck

Abstract—This work describes an advanced path following control strategy enabling overactuated robotic vehicles like the ROboMObil (ROMO) [1] to automatically follow predefined paths while all states of the vehicle’s planar motion are controlled. This strategy is useful for autonomous vehicles which are guided along online generated paths including severe driving maneuvers caused by e.g. obstacle avoidance. The proposed approach combines path following, i.e. tracking a plane curve without a priori time parameterization of a trajectory, with feedback based vehicle dynamics stabilization. A path interpolation method is introduced which allows to perform the path following task employing a trajectory tracking controller. Furthermore a tracking controller based on I/O linearization and quadratic programming based control allocation is proposed which allows employing the vehicle’s overactuation in an optimal manner. The work concludes by a simulative evaluation of the controller performance.

I. INTRODUCTION

This paper presents a centralized control approach for the automatic driving and stabilization of an overactuated robotic vehicle. According to [2] motion control of vehicles can be classified into three generic problems. These are the stabilization of fixed postures, the stabilization of trajectories and the path following problem. The term stabilization of fixed postures refers to the control task of e.g. parking a car at a specified position with a desired orientation. Considering our aim of automatically driving a vehicle, the special case of vehicle parking is not treated in this work. Neglecting this special case, controlling a vehicle can be formulated in two ways either as a trajectory tracking or a path following problem.

Within this paper a path refers to an one-dimensional geometrical figure connecting two points in space, e.g. a road. In contrast to that signals in a trajectory are expressed as a function of time. This distinction between trajectory and path helps to understand the difference between trajectory tracking and path following. For path following the aim is keeping the vehicle position on the defined path. The path following controller does not consider the time when the vehicle is at a specific position. In contrast to that a trajectory tracking controller aims to fulfill predefined time schedules while it is valid to leave the path described by the set of positions passed by the trajectory.

Peter Ritzer is research assistant at the DLR Robotics and Mechatronics Center (RMC), Oberpfaffenhofen, Germany. (E-mail: Peter.Ritzer@dlr.de)
 Christoph Winter is research assistant at the DLR RMC (E-Mail: Christoph.Winter@dlr.de).
 Jonathan Brembeck is research assistant and PhD student at the DLR RMC. (E-mail: Jonathan.Brembeck@dlr.de).

Comparing these inherent properties of the problem formulation to the task of driving a vehicle on roads shows which formulation has to be selected. Driving on roads requires to stay on the track and to not cut corners while there is no time schedule predefining vehicle postures at specific times. These requirements indicate that vehicle control problems are more suitably formulated as path following problems. In addition to following a path, automatic driving of vehicles requires stabilizing the vehicle in case of disturbances. Therefore this work focuses on the design of a vehicle controller combining path following control with an integrated vehicle dynamics control method similar to [3].

In order to provide a general and simple approach for the combination of path following and integrated vehicle dynamics control, the first part of this work proposes a method termed *Time Independent Path Interpolation* (TIPI), which enables to employ a trajectory tracking controller for the path following task. The second part of this paper develops a tracking controller for the overactuated robotic vehicle ROMO. Finally simulation results for two maneuvers are presented. In the first maneuver the control accuracy of the proposed method is demonstrated, while in the second test a comparison of the controlled and uncontrolled vehicle reaction on a severe disturbance is discussed.

II. NOTATION

For the efficient notation of the presented algorithms, different coordinate systems are used. The different frames are denoted by the superscript I, C and P for Inertial, Car and Path frame, respectively. The affiliation of the signal to the Car or the Path is denoted by subscripts C or P.

III. MOTION DEMAND REPRESENTATION

The motion demand $\lambda(s)$ of the time independent path following control is formulated as generalized path with respect to the arc length s . The parametric curve $\lambda(s) \in \mathbb{R}^5$ encompasses the demanded path represented by path position $\mathbf{p}_P^I(s) \in \mathbb{R}^2$, orientation $\varphi_P(s)$ and curvature $\kappa(s)$. In addition the vehicle reference velocity $v_x^P(s)$ along the path tangent \mathbf{x}^P is part of the motion demand illustrated in Fig. 1. Noteworthy is that the partial derivatives $\frac{\partial}{\partial s}\mathbf{p}_P^I(s)$ and $\frac{\partial}{\partial s}\varphi_P(s)$ are implicitly part of the motion demand

$$\frac{\partial}{\partial s}\mathbf{p}_P^I(s) = (\cos \varphi_P(s) \quad \sin \varphi_P(s))^T \quad (1)$$

$$\frac{\partial}{\partial s}\varphi_P(s) = \kappa_P(s) . \quad (2)$$

These partial derivatives are relevant for the online calculation of the demanded trajectory, which is performed by the TIPI introduced in sec. IV. The definition of motion

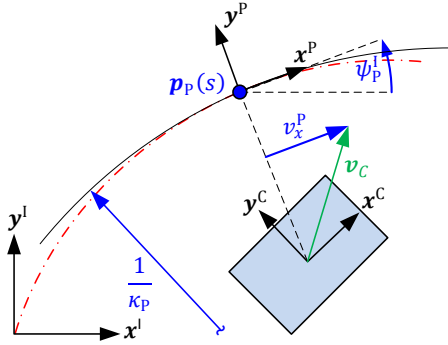


Fig. 1: Geometric quantities and reference control variables defining the motion demand $\lambda(s)$.

demands in this form has been selected, since it enables geometric path planning as proposed in [4]. This method requires the definition of position, orientation and velocity demands independent of time and therefore the motion demand is represented as function of arc length s . In order to provide a general interface to path planning tools the motion demand $\lambda(s)$ is assumed to be represented as a look-up table.

IV. TIME INDEPENDENT PATH INTERPOLATION

The motion demand representation described in the previous section is not directly used for controlling the vehicle in this paper. In order to control the vehicle, an online calculation of set points and its time derivatives is performed. Due to the need of expressing this set point function and its derivatives without utilizing time explicitly, the following approach is termed as *Time Independent Path Interpolation* (TIPI). The outcome of this approach is an online calculated trajectory, which enables performing the path following task by means of a trajectory tracking controller.

A. Online trajectory interpolation

Given the motion demand $\lambda(s)$ the calculation of the set points in time domain can be achieved by expressing the arc length s in dependence of time t . Furthermore the exact calculation of the derivatives of the motion demand can be performed by applying the chain rule. For the first and the second time derivative of $\lambda(t)$ this yields

$$\dot{\lambda}(t) = \frac{\partial}{\partial s} \lambda(s(t)) \cdot \dot{s}(t) \quad (3)$$

and

$$\ddot{\lambda}(t) = \frac{\partial^2}{\partial s^2} \lambda(s(t)) \cdot \dot{s}(t)^2 + \frac{\partial}{\partial s} \lambda(s(t)) \cdot \ddot{s}(t). \quad (4)$$

These equations show that for the calculation of time derivatives of set points, the partial derivatives of the demand functions have to be known and the time derivatives of $s(t)$ need to be calculated. We decided to apply the analytic chain rule to calculate the first-order derivatives of set points and the second-order derivatives are calculated by applying numerical differentiation using DT1-filters. The reason is that the second-order partial derivative of $\lambda(s)$ is not considered in the planning of $\lambda(s)$ and the second time derivative of s is not computed to reduce the computational effort.

B. Definition of the actual parameter value

The first step in the TIPI is to determine the characteristic of the actual parameter value s^* . This is trivial in case the vehicle position p_c coincides with a point on the path. In this case s^* is defined by the solution of equation (5)

$$p_p(s^*) \stackrel{!}{=} p_c. \quad (5)$$

In case the vehicle position does not lie on the reference path, (5) has no solution. A common approach to relax strict equality requirements is to solve the equation in a least-squares sense. For (5) this approach yields the minimization of the displacement $e(s)$ between vehicle and reference path

$$s^* = \operatorname{argmin}_s \left\| \frac{p_p(s) - p_c}{e(s)} \right\|_2, \quad (6)$$

as depicted in Fig. 2. The geometrical interpretation of this minimization problem is that $p_p(s^*)$ results by projecting p_c orthogonally on the path $p_p(s)$. For this issue it is stated in [2] that $p_p(s^*)$ exists and is unique if $\|e(s^*)\|_2$ is smaller than the lower bound of the curve radius. For the TIPI this condition implies that the inverse of the maximal curvature of the demanded vehicle path defines the maximum lateral displacement for which s^* exists.

The graphic solution of the optimization problem (6) is illustrated in Fig. 2. Investigating this figure it can be seen that $\|e(s)\|_2$ is minimal if the vectors $e(s)$ and $x^p(s)$ are orthogonal, which is the first order optimality condition of (6). This condition implies that the tangential displacement component $e_x^p(s)$ is zero. Hence, the actual parameter value s^* is defined by the root finding problem

$$e_x^p(s^*) \stackrel{!}{=} 0. \quad (7)$$

C. Dynamic root finding approach

Following the definition of the actual parameter the second step in TIPI is the calculation of $s(t) = s^*$ by solving (7). Basically this algebraic equation can be solved iteratively by e.g. the bisection method [5]. However, to avoid iterative routines in the vehicle control loop, a dynamic root finding approach depicted in Fig. 3 is proposed to find the solution of (7) in every time instant. Essentially the dynamic root finding algorithm spreads the iterations over time. It can be represented with a proportional control law which adjusts the

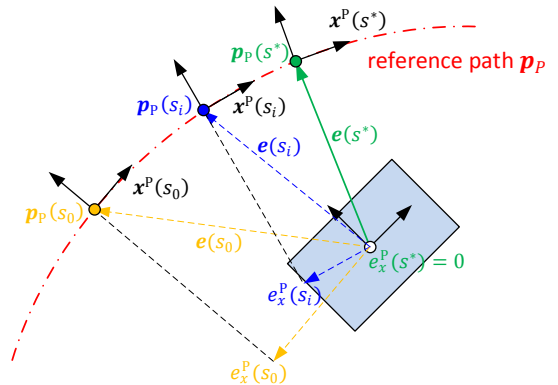


Fig. 2: Illustration of intermediate steps of the root finding problem in the time independent path interpolation. Starting from the initial guess s_0 the algorithm converges towards s^* , c.f. (6).

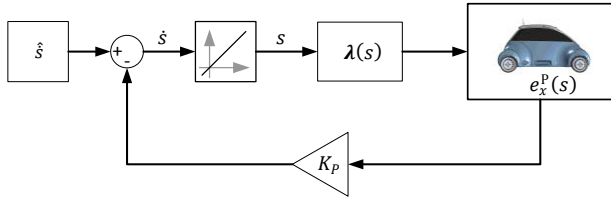


Fig. 3: Block diagram describing the schematic of the dynamic root finding approach for solving (7).

path parameter rate \dot{s} in order to force e_x^p to tend to zero

$$\dot{s} = -K_p \cdot e_x^p. \quad (8)$$

In a more sophisticated version this feedback loop is extended by an estimated curve parameter rate \hat{s} which is applied as feed forward, c.f. Fig. 3. With the help of this feed forward term it is theoretically possible to reduce the longitudinal displacement e_x^p to zero. In practice it allows decreasing the remaining longitudinal displacement and makes it independent of the vehicle velocity [6].

The estimated path parameter rate \hat{s} employed in the feed forward path in Fig. 3 is based on the calculation of \dot{s} derived on the assumption that s^* is known. This situation is depicted in Fig. 4 assuming that the motion of the reference point along the path can be regarded as a sequence of motions along the tangent of a circle with radius r_p . Similarly, the vehicle motion can be decomposed into a motion along the tangent of a circle with the same center and into a motion along radial direction. The radius of this circle is given by r_c . The ideal path parameter rate \dot{s} results from considering the tangential component of the vehicle velocity $v_{C,x}^p$.

As illustrated in Fig. 4 the center of the circles, the reference point $p_p(s^*)$ and the vehicle position p_c have to stay on a line in order to keep e_x^p equal to zero. Utilizing this geometrical relation together with the application of the intercept theorem yields the basic estimation equation

$$\dot{s} \cdot r_c = v_{C,x}^p \cdot r_p. \quad (9)$$

In this expression the instantaneous radii of rotation r_p and r_c can be expressed in dependence of the path curvature $\kappa_p(s^*)$ and lateral position error $e_y^p(s^*)$. With the help of this relation and the substitution of s^* by the actually known parameter value s the path parameter rate can be estimated as

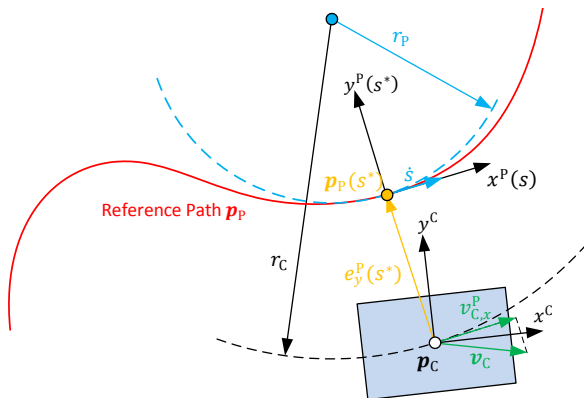


Fig. 4: Kinematic relations employed for the path parameter rate estimation

$$\hat{s}(s) = \frac{v_{C,x}^p}{1 + \kappa_p(s) \cdot e_y^p(s)}. \quad (10)$$

Implemented in discrete time the presented root finding approach calculates the path parameter value and its first time derivative depending on an initial guess value. This approach together with the *Online Trajectory Interpolation* discussed in sec. IV A. represents an iteration free method for performing the TIPI algorithm.

V. CONTROL ALGORITHM

In the previous section an algorithm is introduced, which enables to transform the path following task into a tracking task of an online calculated trajectory. With this groundwork it suffices to develop a trajectory tracking method to complete the path following controller. The design of the tracking controller is the focus of this section.

The presented approach separates the vehicle control task into two sub problems: the high level control of the vehicle via virtual control inputs and the allocation of the virtual control demands to physical control inputs.

A. Controller Architecture

An overview of the controller architecture is provided in Fig. 5. The top level control algorithm is the *Central Vehicle Dynamics Controller*, a feedback controller performing the tracking task in the motion coordinates of the vehicle at its center of gravity. The intermediate control layer consists of the *Control Allocator* (CA), which provides an interface between the tracking controller and the vehicle actuators. In the bottom layer of the control architecture lie the distributed actuator controllers which are not in the scope of this paper.

B. Central Vehicle Dynamics Controller

For the controller synthesis a vehicle model in the form of a body with planar degrees of freedom is chosen. This simplified vehicle model is represented as nonlinear state space model and therefore the design of the tracking controller is based on input/output linearization of the vehicle model. This design method, applied in robotics [7] and flight control [8], enables to exactly linearize the input to output behavior of the controlled system and thus applying a linear control law for tracking a trajectory. In addition to linearizing the system behavior the calculated transformation

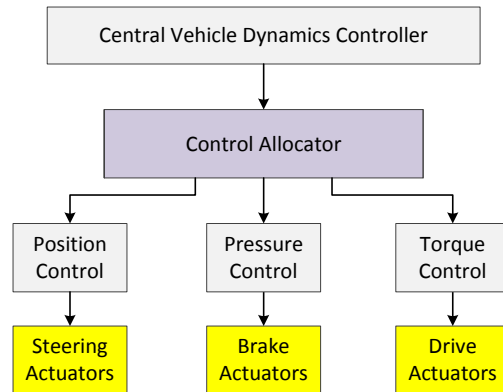


Fig. 5: Control Architecture with the Vehicle Dynamics Controller performing the tracking task and the Control Allocator as interface to the actuator level.

is used to integrate the motion demand $\lambda(s(t))$ into the controller design. This is achieved through a reference model covering the states predefined by the motion demand.

The vehicle model inputs \mathbf{u}_C are virtual acceleration actuators representing the Control Allocator interface

$$\mathbf{u}_C = (a_{C,x}^C \quad a_{C,y}^C \quad \ddot{\psi}_C)^T. \quad (11)$$

The model outputs \mathbf{y}_C are the vehicle velocity along the path tangent $v_{C,x}^P$, the lateral displacement e_y^P and the orientation offset $\Delta\psi$, which gives

$$\mathbf{y}_C = (v_{C,x}^P \quad e_y^P \quad \Delta\psi)^T. \quad (12)$$

All output signals are expressed with respect to the reference system $\{\mathbf{x}^P, \mathbf{y}^P\}$ defined at the reference position given by the path parameter $s(t)$ as illustrated in Fig. 1. Based on this input and output definitions the state space model of the vehicle is

$$\mathbf{x}_C = (p_{C,x}^I \quad p_{C,y}^I \quad \psi_C \quad v_{C,x}^C \quad v_{C,y}^C \quad \dot{\psi}_C)^T, \quad (13)$$

$$\dot{\mathbf{x}}_C = \begin{pmatrix} v_{C,x}^C \cos \psi_C - v_{C,y}^C \sin \psi_C \\ v_{C,x}^C \sin \psi_C + v_{C,y}^C \cos \psi_C \\ \dot{\psi}_C \\ \psi_C \cdot v_{C,y}^C \\ \psi_C \cdot v_{C,x}^C \\ 0 \end{pmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{u}_C, \quad (14)$$

$$\mathbf{y}_C = \begin{pmatrix} v_{C,x}^C \cdot \cos(\psi_C - \psi_P) - v_{C,y}^C \cdot \sin(\psi_C - \psi_P) \\ \sin(\psi_P) \cdot (p_{C,x}^I - p_{P,x}^I) - \cos(\psi_P) \cdot (p_{C,y}^I - p_{P,y}^I) \\ \psi_P - \psi_C \end{pmatrix}. \quad (15)$$

In addition to the vehicle model a reference model, representing the dynamics of the demand values is employed to calculate the vehicle model output defined in (15). The input \mathbf{u}_P of the reference system is defined as acceleration of the reference body in the inertial frame

$$\mathbf{u}_P = (a_{P,x}^I \quad a_{P,y}^I \quad \ddot{\psi}_P)^T. \quad (16)$$

With this input definition the state space model of the reference system is given as three parallel double integrators with the system equations

$$\mathbf{x}_P = (p_{P,x}^I \quad p_{P,y}^I \quad \psi_P \quad v_{P,x}^I \quad v_{P,y}^I \quad \dot{\psi}_P)^T, \quad (17)$$

$$\dot{\mathbf{x}}_P = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \mathbf{x}_P + \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{u}_P, \quad (18)$$

$$\mathbf{y}_P = (p_{P,x}^I \quad p_{P,y}^I \quad \psi_P)^T. \quad (19)$$

These two dynamic systems are combined to an overall design model used in the tracking controller development. By applying the I/O-linearization method introduced in [9] to this design model, the derivation of an analytic feedback law is possible. The result can be represented in the form of

$$\begin{pmatrix} \mathbf{u}_C \\ \mathbf{u}_P \end{pmatrix} = \mathbf{A}(\mathbf{x}_C, \mathbf{x}_P)^{-1} \cdot \left(\begin{pmatrix} \mathbf{y}_C \\ \mathbf{y}_P \end{pmatrix} - \mathbf{d}(\mathbf{x}_C, \mathbf{x}_P) \right), \quad (20)$$

with the virtual control inputs \mathbf{v}_C , \mathbf{v}_P , decoupling matrix \mathbf{A} and decoupling vector \mathbf{d} . The virtual control variable \mathbf{v}_P is given by the second time derivative of the reference path interpolated with the TIPI algorithm introduced in sec. IV. Hence only the virtual vehicle input \mathbf{v}_C remains as the

control input. The calculation of this input is performed by linear feedback (FB) control and a feed forward (FF) part:

$$\mathbf{v}_C = \underbrace{\begin{pmatrix} \dot{v}_x^{P*} \\ \dot{e}_y^{P*} \\ \Delta \dot{\psi}^* \end{pmatrix}}_{\text{FF}} + \underbrace{\begin{pmatrix} k_v(v_x^{P*} - v_{C,x}^P) \\ k_e(\dot{e}_y^{P*} - \dot{e}_y^P) + k_e(e_y^{P*} - e_y^P) \\ k_\psi(\Delta \dot{\psi}^* - \Delta \dot{\psi}) + k_\psi(\Delta \psi^* - \Delta \psi) \end{pmatrix}}_{\text{FB}}. \quad (21)$$

Assuming perfect control allocation and I/O feedback linearization, the overall dynamics of the vehicle from \mathbf{v}_C to \mathbf{y}_C correspond to three decoupled integrator systems. As a result, the error dynamics of the closed loop longitudinal velocity control can be interpreted as a first order lag element. In the same way the error dynamics of the closed loop lateral displacement and orientation offset control correspond to the dynamics of second order lag elements. This proves the stability of the closed loop system and furthermore it allows a simple and interpretable parameterization of the tracking controller by selecting the time- and damping-constants of these standard control elements.

C. Control Allocation

The control method described so far rests upon the assumption of a generalized actuator, which allows the direct control of vehicle acceleration (11) corresponding to the planar motion degrees of freedom. Since such an actuator does not exist, it is necessary to provide an interface between the tracking controller and the actual vehicle actuators. This interface is the Control Allocator (CA) depicted in Fig. 5.

The task of the CA is to map the virtual demand \mathbf{u}_C to the control input vector $\tilde{\mathbf{u}}$ of the vehicle. For the considered overactuated vehicle architecture, the ROMO, the control input $\tilde{\mathbf{u}}$ has eight dimensions. These inputs are the tire rotational speeds $\boldsymbol{\omega}$ and the steering angles $\boldsymbol{\delta}$ with

$$\tilde{\mathbf{u}} = (\omega_1 \quad \omega_2 \quad \omega_3 \quad \omega_4 \quad \delta_1 \quad \delta_2 \quad \delta_3 \quad \delta_4)^T. \quad (22)$$

The virtual demand \mathbf{u}_C computed by the tracking controller is three-dimensional and represents the generalized acceleration of the vehicle's center of gravity according to (11). Using a double track model [10] of the ROMO with a static tire model, e.g. Burckhardt tire [11], the relation between generalized acceleration and vehicle control input $\tilde{\mathbf{u}}$, wheel load \mathbf{F}_z and the generalized vehicle velocity $\mathbf{v}_{\text{gen}}^C = (v_{C,x}^C \quad v_{C,y}^C \quad \dot{\psi}_C)^T$ can be described as

$$\mathbf{u}_C = \mathbf{b} \left(\underbrace{\boldsymbol{\omega}, \boldsymbol{\delta}}_{\tilde{\mathbf{u}}}, \mathbf{F}_z, \mathbf{v}_{\text{gen}}^C \right)^T. \quad (23)$$

The control efficiency function $\mathbf{b}(\cdot)$ is a surjective function $\mathbb{R}^8 \rightarrow \mathbb{R}^3$ which maps control input configurations to generalized accelerations. The CA task is to calculate a solution $\tilde{\mathbf{u}}$ for a virtual input \mathbf{u}_C which satisfies (23) for given \mathbf{F}_z and $\mathbf{v}_{\text{gen}}^C$. This solution has to be physically feasible, i.e.

$$\underline{\tilde{\mathbf{u}}} \leq \tilde{\mathbf{u}} \leq \bar{\tilde{\mathbf{u}}} \quad (24)$$

with lower and upper bounds $\underline{\tilde{\mathbf{u}}}$ and $\bar{\tilde{\mathbf{u}}}$, unique and must always exist. Two cases have to be distinguished to achieve the solution of this problem. In the case of one or more solutions which satisfy (23) and (24) the CA has to select one element out of the solution set. In case no input

satisfies (23) and (24) an approximate solution satisfying (24) and minimizing some criteria $\text{dist}(\mathbf{u}_C, \mathbf{b}(\tilde{\mathbf{u}}))$ has to be selected. An optimization based control allocation method solving this problem has been applied to the energy optimal control of the ROMO in [13]. For this work the approach is further developed with the aim of wheel slip optimal demand distribution.

In order to calculate the optimization based CA problem in real time, a quadratic programming method is employed. For that purpose the nonlinear CA problem is transformed into a linear CA problem as shown in the following. The control efficiency function \mathbf{b} in (23) is considered as nonlinear output function of the vehicle. By applying the I/O-linearization method on this output the following linear relationship can be found between the first time derivatives of the output $\dot{\mathbf{u}}_C$, the input $\tilde{\mathbf{u}}$, wheel load $\dot{\mathbf{F}}_z$ and $\dot{\mathbf{v}}_{\text{gen}}^C$

$$\dot{\mathbf{u}}_C = \frac{\partial \mathbf{b}}{\partial \boldsymbol{\omega}} \dot{\boldsymbol{\omega}} + \frac{\partial \mathbf{b}}{\partial \boldsymbol{\delta}} \dot{\boldsymbol{\delta}} + \frac{\partial \mathbf{b}}{\partial \mathbf{F}_z} \dot{\mathbf{F}}_z + \frac{\partial \mathbf{b}}{\partial \mathbf{v}_{\text{gen}}^C} \dot{\mathbf{v}}_{\text{gen}}^C. \quad (25)$$

The controlled vehicle input is $\mathbf{u} = (\boldsymbol{\tau}_{\text{act}} \quad \boldsymbol{\delta})^T$ with $\boldsymbol{\delta}$ the steering angle rate and $\boldsymbol{\tau}_{\text{act}}$ the actuator torque input, which combines the drive train and brake torques $\boldsymbol{\tau}_{\text{drive}}$, $\boldsymbol{\tau}_{\text{brake}}$. To connect (25) with the vehicle input \mathbf{u} it is necessary to insert the wheel momentum balance equation

$$\dot{\boldsymbol{\omega}} = \frac{1}{\Theta} \boldsymbol{\tau}_{\text{road}} + \frac{1}{\Theta} \underbrace{(\boldsymbol{\tau}_{\text{drive}} + \boldsymbol{\tau}_{\text{brake}})}_{\boldsymbol{\tau}_{\text{act}}} \quad (26)$$

between the tire-road contact torque $\boldsymbol{\tau}_{\text{road}}$ and the torque input $\boldsymbol{\tau}_{\text{act}}$, with Θ the wheel inertia. This yield the linear CA problem

$$\dot{\mathbf{u}}_C = \mathbf{c} + \mathbf{B}\mathbf{u} \quad (27)$$

$$\underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}}, \quad (28)$$

with

$$\mathbf{c} = \frac{\partial \mathbf{b}}{\partial \boldsymbol{\omega}} \cdot \frac{1}{\Theta} \boldsymbol{\tau}_{\text{road}} + \frac{\partial \mathbf{b}}{\partial \mathbf{F}_z} \cdot \dot{\mathbf{F}}_z + \frac{\partial \mathbf{b}}{\partial \mathbf{v}_{\text{gen}}^C} \cdot \dot{\mathbf{v}}_{\text{gen}}^C \quad (29)$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial \mathbf{b}}{\partial \boldsymbol{\omega}} \cdot \frac{1}{\Theta} & \frac{\partial \mathbf{b}}{\partial \boldsymbol{\delta}} \end{bmatrix} \in \mathbb{R}^{3 \times 8}.$$

In this formulation the wheel torque $\boldsymbol{\tau}_{\text{act}}$ is the sum of all actuated torques at the wheels. The allocation of the wheel torques is not part of this work and is treated in [14].

The computation of (29) requires the measurement of wheel speed $\boldsymbol{\omega}$, steering angle $\boldsymbol{\delta}$, vehicle speed $\mathbf{v}_{\text{gen}}^C$ and vehicle speed derivative $\dot{\mathbf{v}}_{\text{gen}}^C$. The equivalent tire-road contact torque $\boldsymbol{\tau}_{\text{road}}$, the wheel load \mathbf{F}_z and its time derivative are also needed but not accessible via measurements and therefore have to be observed. The estimation of the wheel load is based on the assumption of a static redistribution of the vehicle mass as proposed in [11] and hence the wheel load rate $\dot{\mathbf{F}}_z$ is estimated as zero. The estimate of $\boldsymbol{\tau}_{\text{road}}$ is extracted from the momentum balance of each wheel (26) using measurements of wheel speeds $\boldsymbol{\omega}$ and actuator torques $\boldsymbol{\tau}_{\text{act}}$. The overall implementation of the nonlinear CA is shown in Fig. 6. It includes a linear control loop with demand saturation computing the derivative of the control input $\dot{\mathbf{u}}_C$ in the outer loop. The inner loop QP CA solves the linearized CA problem defined by (27) and (28) with the two-step optimization approach introduced in [12]. This method is

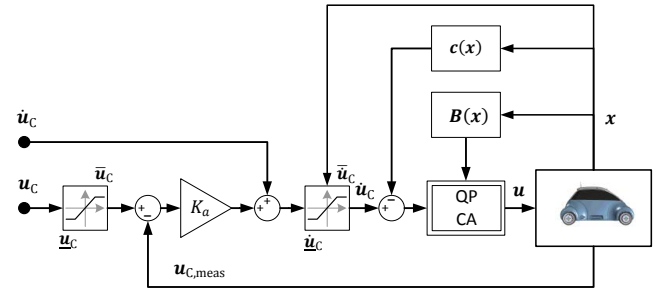


Fig. 6: Linear feedback control and QP CA employed to solve the nonlinear CA problem.

designed to preferably compute control inputs \mathbf{u} which minimize wheel slip.

VI. SIMULATION RESULTS AND EVALUATION

This section evaluates the capabilities and performance of the developed control system. The evaluation comprises simulations based on a double track model equipped with nonlinear tire models [14], steering actuator and traction motor models. The vehicle, controller and observer parameterization is identical for all test maneuvers. The evaluation is divided into two parts. The first part demonstrates the achievable control accuracy of the proposed controller in a maneuver without external disturbances. In the second part the disturbance rejection behavior of the control strategy is shown.

A. Evaluation of control accuracy

The control accuracy is tested by following the curvature optimal path of the Double Lane Change (DLC) maneuver according to ISO 3888-1 cone boundaries. This maneuver is performed at constant vehicle velocity $v_x^{P*} = 18 \frac{\text{m}}{\text{s}}$ such that its focus is mainly on the tracking abilities of the lateral position controller. The demanded path of the DLC incorporating the vehicle width is depicted in Fig. 7.

The simulation result of the DLC maneuver is depicted in Fig. 8. The plots show that the control method is suitable to negotiate the test maneuver and high control accuracy is indicated by the RMS values $0.0156 \frac{\text{m}}{\text{s}}$, 7.61 mm , 0.115° of longitudinal speed error, lateral position error and orientation error, respectively.

B. Evaluation of disturbance rejection

In addition to the path following an important capability of the proposed control algorithm is the stabilization of the vehicle in critical situations. To simulate such a situation an experiment is performed on a low friction surface with $\mu_{\text{max}} = 0.3$. During the test the demanded path is a straight line with a fixed velocity demand of $14 \frac{\text{m}}{\text{s}}$. The

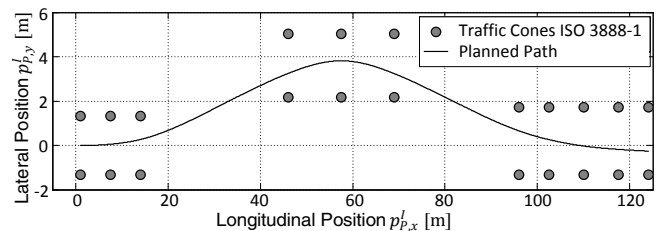


Fig. 7: Demanded path negotiating the Double Lane Change ISO 3888-1 for a vehicle width of 1.45 m.

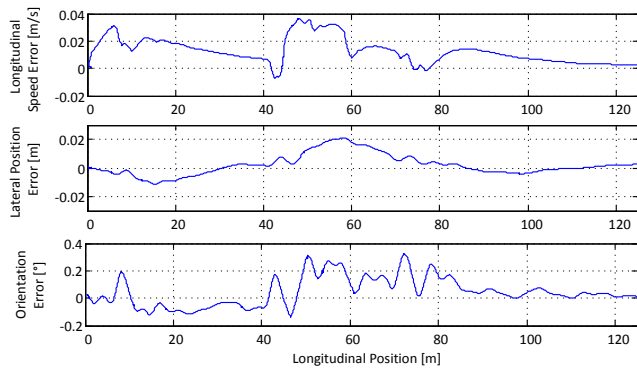


Fig. 8: Behavior of the test vehicle controlled by the path following controller negotiating the Double Lane Change maneuver.

disturbance is emulated as force pulse at the rear wheels, where each wheel is excited with a force pulse of 4000 N in lateral direction for a duration of 0.2 s. Simulation results of two experiments with and without control are compared to demonstrate the improvement of the vehicle stability due to the control. The first experiment is conducted without closed-loop control, with steering angles and traction motor torques fixed to zero. The reaction of the uncontrolled vehicle to the force impulse is depicted in Fig. 9. The plot shows that the vehicle skids out of control and performs a full spin. The lateral displacement of the vehicle is 9 m when it stops skidding. In Fig. 10 the simulation result for activated control is presented. It shows that the controller is capable to avoid spinning of the vehicle. The vehicle orientation exhibits a maximum value of 35° and the lateral displacement is limited to values smaller than 0.8 m. Most important is the fact that the controller is able to compensate the disturbance and keep the vehicle stable.

VII. CONCLUSION AND OUTLOOK

The evaluation experiments show that the control algorithm developed in this work is suitable for the automatic control of over-actuated vehicles. The proposed algorithm enables the tracking of predefined paths, like planned in [4], with high accuracy and assures the stability of the vehicle in case of severe disturbances. Furthermore, the implementation on a real-time platform shows that the calculations performed within the proposed control algorithm are feasible within less than 1 ms. All these results indicate the potential of the developed approach and motivate real world tests with the ROboMObil. In order to achieve this, an important step in future work is the investigation and alleviation of the effects introduced by noisy and delayed feedback signals.

A future application of the proposed control approach is in autonomous driving of vehicles. In this scenario the TIPI

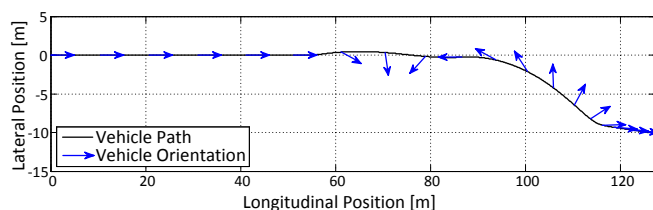


Fig. 9: Reaction of the uncontrolled vehicle to the kick plate force pulse on the rear wheels.

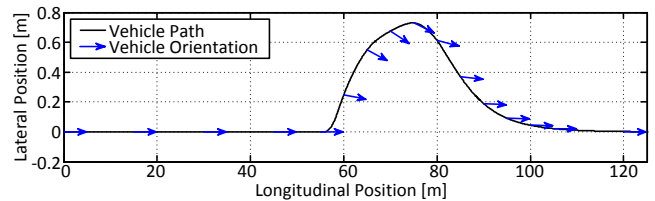


Fig. 10: Reaction of the VDC controlled vehicle to the kick plate force pulse on the rear wheels.

algorithm is extended by a system detecting the reference lane. This has the great advantage that no absolute position of the vehicle needs to be known to perform a lane following control. A second application of the path following algorithm is the follower control in a platooning scenario. In this scenario the path of the leader is recorded and transmitted to the followers via Car2Car communication.

ACKNOWLEDGMENT

The authors express their gratitude to the ROboMObil team and the head of the institute Johann Bals for their support.

REFERENCES

- [1] J. Brembeck, L. M. Ho, A. Schaub, C. Satzger and G. Hirzinger, "Romo - The robotic electric Vehicle," in *22nd International Symposium on Dynamics of Vehicle on Roads and Tracks*, 2011.
- [2] M. Pascal and S. Claude, "Motion Control of Wheeled Mobile Robots," in *Handbook of Robotics*, K. Siciliano, Ed., Springer, 2008, pp. 799-826.
- [3] R. Orend, "Modelling and control of a vehicle with single-wheel chassis actuators," in *IFAC World Congress*, Prague, 2005.
- [4] J. Brembeck and C. Winter, "Real-time capable path planning for energy management systems in future vehicle architectures," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014.
- [5] C. Woodford and C. Phillips, "Numerical Methods with Worked Examples: Matlab Edition," C. Woodford and C. Phillips, Eds., Springer, 2012, pp. 47-71.
- [6] P. Ritzer, J. Brembeck and R. Kennel, "Model Based Vehicle Dynamics Control for Modern Vehicle Architectures," Masters Thesis, TUM, Munich, 2013, <http://elib.dlr.de/87120/>.
- [7] K. Kreutz, "On manipulator control by exact linearization," in *IEEE Transactions on Automatic Control*, 1989.
- [8] T. Lombaerts and G. Looye, "Design and flight testing of nonlinear autoflight control laws," in *Guidance, Navigation, and Control and Co-located Conferences*, AIAA, 2012.
- [9] A. Isidori, *Nonlinear Control Systems*, Springer, 1995.
- [10] R. Orend, "Integrierte Fahrdynamikregelung mit Einzelradaktorik- Ein Konzept zur Darstellung des fahrdynamischen Optimums," Universitaet Erlangen, 2006.
- [11] U. Kiencke and L. Nielsen, "Vehicle Modelling," in *Automotive Control Systems*, Springer, 2005, pp. 301-349.
- [12] O. Härkegård, "Control Allocation Methods," in *Backstepping and Control Allocation with Applications to Flight Control*, Linköping, Linköping University, 2003, pp. 114-117.
- [13] J. Brembeck and P. Ritzer, "Energy Optimal Control of an Over Actuated Robotic Electric Vehicle using enhanced Control Allocation Approaches," in *IEEE Intelligent Vehicles Symposium*, Madrid, 2012.
- [14] C. Satzger, "Combined Wheel-Slip Control and Torque Blending using MPC," in *International Conference on Connected Vehicles and Expo*, Vienna, 2014.
- [15] H. B. Pacejka and I. J. M. Besselink, "Magic Formula Tyre Model with Transient Properties," *Vehicle System Dynamics*, pp. 234-249, 1997.