Mechatronics 100 (2024) 103190

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Model predictive path-following control for truck–trailer systems with specific guidance points — design and experimental validation ☆

Markus Lukassek [a,b,*], Julian Dahlmann [a], Andreas Völz [a], Knut Graichen [a]

[a] Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany
[b] ZF Friedrichshafen AG, 88039 Friedrichshafen, Germany

ARTICLE INFO

Keywords:
Model predictive control
Path-following control
Vehicle state estimation
Truck–trailer systems
Automotive
Microcontroller

ABSTRACT

This article presents an experimental validation of a model predictive path-following control algorithm (PF-MPC) applied to a truck–trailer system, encompassing both forward and backward motions. The proposed controller is designed to precisely follow a predefined path generated by a path planner, with a designated guidance point positioned on either the truck or the trailer. The algorithm's performance is assessed through implementation and validation on a model-scaled truck–trailer system, where MPC, state estimation, and low-level control are executed on a microcontroller (MCU). The experimental results demonstrate the effectiveness of the proposed control approach in achieving highly accurate path-following performance, even when operating in the challenging context of unstable backward motion, and with the involvement of up to two trailers. Moreover, the successful implementation of the algorithm on a microcontroller underscores its suitability for real-time control applications. The results of this study collectively highlight the promising potential of the proposed control algorithm for practical utilization in autonomous driving systems.

1. Introduction

The transportation of cargo by truck–trailer combinations represents a cornerstone of contemporary logistics, providing cost-effective and dependable delivery over long distances [1]. Maneuvering of such articulated vehicles poses a challenging task for both human drivers and automated control systems, particularly in tight spaces or with multiple trailers. This difficulty is evident when backing up trailers towards a loading dock in a busy and narrow yard. Exceptional skill, precision, and spatial awareness are needed to coordinate the movement of the tractor unit and trailers. This becomes even more apparent in reverse motion due to the instability of the system which could cause the trailers to collapse, known as *jackknife effect* [2].

In the agricultural sector, it is often not the vehicle itself that needs to be positioned, but rather a potentially very wide implement that is mounted on the vehicle or trailer [3]. Moreover, cooperative work is commonly required, such as when a tractor and trailer must follow the combine harvester's discharge to collect the crops. Such tasks require highly skilled and attentive driving, which can be extremely tiring, particularly over long distances [4].

From a systems theory perspective, the vehicle–trailer system poses an interesting problem due to its nonlinear nature and the presence of nonholonomic constraints. In addition, the system is unstable and non-minimum-phased when reversing with trailers [2,5]. This implies that a countermovement must first be performed to position the last trailer in the combination. These characteristics make it difficult to transition the vehicle from one point to another, often requiring changes in direction.

In this work, a car-like vehicle with a steerable, kinematic front axle and a kinematic rear axle is used as the towing vehicle. The actual number of axles is not relevant as long as there are no articulated joints within the vehicle itself and it satisfies the Ackermann condition, that all wheels rotate around one instant center of rotation during cornering [6]. If trailers are added to the combination, each joint-axle pair is considered as trailer, which kinematically means that there may be more trailers in the system than is obvious at first glance (see Fig. 1). Each kinematic trailer introduces another nonholonomic constraint to the system [7], which makes maneuvering increasingly challenging with multiple trailers.

The system category depends on where the coupling points of each trailer are located in relation to the preceding axle [8]. If all of the trailers are coupled in the middle of the axle (*on-axle hitch*), it is classified as a Standard n-Trailer system (SnT), where n indicates the number of kinematic trailers. On the other hand, if all of the couplings

☆ This paper was recommended for publication by Associate Editor Joachim Rudolph.

* Corresponding author at: Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany.
E-mail addresses: markus.lukassek@fau.de, markus.lukassek@zf.com (M. Lukassek), julian.dahlmann@fau.de (J. Dahlmann), andreas.voelz@fau.de (A. Völz), knut.graichen@fau.de (K. Graichen).

https://doi.org/10.1016/j.mechatronics.2024.103190
Received 22 December 2023; Received in revised form 26 March 2024; Accepted 1 April 2024
Available online 17 April 2024

are longitudinally offset from the preceding axle (*off-axle hitch*), it is classified as a non-Standard n-Trailer system (nSnT). When both types of couplings are present, it is referred to as a General n-Trailer system (GnT). This work focuses on the GnT system category, as *EuroCombi* variants of truck–trailer combinations used in the European Union fall into this category [9].

The SnT system is differentially flat with the midpoint of the rearmost axle of the combination as linearizing output, making it suitable for control methods using exact feedback linearization and inversion-based planning. Several trajectory tracking and path-following control methods have been proposed using the flatness property [10–12]. However, these concepts are prone to modeling errors and disturbances, and it is difficult to integrate state constraints to guarantee safety as well as transformations to a particular guidance point.

In contrast, applying appropriate algorithms to the GnT and nSnT systems is even more challenging as they are not differentially flat. Several nonlinear methods, including linearization [13], Lyapunov-based methods [14], and cascade-like structures [8,15], have been proposed to address this problem. As system properties are subject to change based on coupling parameters and direction of movement, model predictive control (MPC) presents a viable solution for a range of vehicle–trailer system applications. This has already been investigated in numerous studies, most of them in simulation only [16–18]. The integration of MPC on real vehicles has been successfully shown in e. g., [19–23] with one trailer and in [24,25] with two trailers. In all of these, computations are performed on high-performance hardware. The main advantage of the optimization along a moving horizon is the prediction into the future. This can be used for collision checking and to some extent as a local trajectory planner.

Besides stabilizing control algorithms, motion planning is essential for vehicle automation. The *global planning* problem of finding a path from a starting point to an end point requires information on the behavior of the system. This implies that control signals for steering angle, speed, or driving direction are necessary, regardless of whether the algorithm is sampling-based [26,27] or lattice-based [28,29].

The task of finding these control inputs is known as *local planning*. This involves finding a control trajectory for the towing vehicle that will transition the system from the initial state to the desired target state. Approaches for local planning of SnT systems that exploit the flatness of the system are usually based on generating polynomial trajectories imposing certain requirements on the curve. By inversion of the curve for the flat output, trajectories for the towing vehicle can be calculated [30].

For GnT systems, inversion-based techniques together with a numerical approach for the non-flat system parts are demonstrated in [31]. Solving the non-flat planning problem of the nSnT system is addressed in [32]. Optimal control-based approaches are often chosen because they have the advantage of imposing additional constraints on the design, allowing, for example, safe corridors [33–35]. In addition, suboptimal paths of global planning can be improved, for example, by eliminating unnecessary direction changes or movements [36]. For this work, the path planner from [31] was adapted to generate the scenarios that do not consist of simple geometric shapes such as circles or the lying eight figure.

The main contribution of this work in contrast to the simulation studies [18,37] is the integration of the model predictive path-following control for an arbitrary guidance point on a real model-scaled truck–trailer combination with a semi-trailer (G1T) or a full trailer (G2T) and performing a feasibility study. This includes computation of the MPC algorithm on a state-of-the-art microcontroller in real-time alongside with low-level control of the actuation of steering servo and traction motor. Besides that, pose and state estimation play a crucial role to successfully operate a truck–trailer combination in reverse motion. It is important to note that solving this particular MPC problem implies solving a nonlinear, non-convex and constrained optimal control problem in deterministic time on very limited hardware with only single floating-point precision.

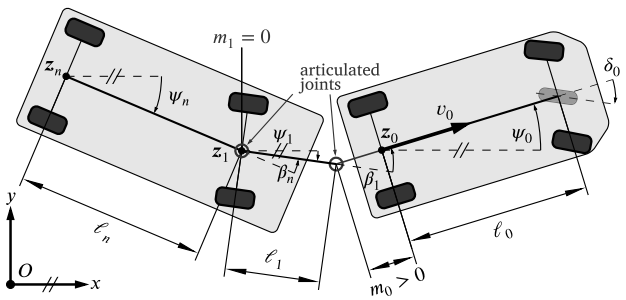


Fig. 1. Exemplary configuration of the truck with an attached full trailer, corresponding to the EuroCombi (G2T). It is kinematically modeled with two single-axle trailers, resulting in a two-joint articulated vehicle.

For performance evaluation, typical real-world scenarios are considered, such as backing up to a loading ramp with the trailer. For agricultural applications, a mower on the vehicle front and a field sprayer on the trailer are mounted where a path is followed with the edge of the respective implement. These experiments demonstrate that a model predictive path-following controller can be implemented in a real-time feasible manner on a microcontroller and that it can successfully stabilize a real truck even with two trailers during complex maneuvers involving both forward and difficult backward motions.

The paper is organized in the following way. Section 2 introduces the truck–trailer model used for the MPC. Section 3 outlines the nonlinear model predictive path-following control. Section 4 explains details regarding the implementation of the MPC algorithm on embedded hardware. Section 5 provides a detailed description of the experimental setup, encompassing mechanical, electrical, and software design, as well as state estimation and low-level control. Section 6 presents experiments with the model-scaled truck–trailer-combination and discusses the results. Finally, Section 7 provides a concise summary of the results and an outlook for possible further research.

2. System model of the truck–trailer combination

The derivation of the system model is separated into two parts: First, the vehicle–trailer model is presented for an exemplary two trailer configuration, but may be extended to n trailers. Then the transformation to the guidance point is performed.

2.1. Kinematic vehicle–trailer system model

The system considered is a car-like truck with passively coupled single-axle trailers. An exemplary vehicle configuration is shown in Fig. 1, but note that the proposed method can be applied to arbitrary combinations of a truck and one or multiple trailers. The towing truck consists of a wheelbase ℓ_0 as well as a hitching offset m_0. The actuation is performed via

$$\boldsymbol{u} = \begin{bmatrix} v_0 \\ \delta_0 \end{bmatrix} \in \mathbb{U}, \tag{1}$$

with the longitudinal rear axle velocity v_0 and the steering angle δ_0 of a virtual wheel in the middle of the front axle. During cornering, the truck follows a circular path with a radius of ρ_0. This radius is the inverse of the motion curvature κ_0 and is expressed by the relationship between the wheelbase and the steering angle as

$$\kappa_0 = \frac{1}{\rho_0} = \frac{\tan \delta_0}{\ell_0}. \tag{2}$$

Each trailer $i = 1, 2$ consists of a drawbar ℓ_i and also of a hitching offset m_i. The hitching angle is denoted as β_i and specifies the difference in yaw angles ψ_i to the preceding segment of the vehicle–trailer combination, i. e.,

$$\beta_i = \psi_{i-1} - \psi_i, \quad i = 1, 2. \tag{3}$$

To describe the motion of the truck's Cartesian rear axle pose $\boldsymbol{z}_0 = [x_0, y_0, \psi_0]^\top$, the well-known kinematic single-track model [38]

$$\dot{\boldsymbol{z}}_0 = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{\psi}_0 \end{bmatrix} = v_0 \begin{bmatrix} \cos\psi_0 \\ \sin\psi_0 \\ \kappa_0 \end{bmatrix} \tag{4}$$

is used. The dynamics of the trailers are not described in the same way using the trailer's pose, but solely using the dynamics of the hitching angles β_i. For that, consider the vector

$$\boldsymbol{w}_0 = \begin{bmatrix} \omega_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} \dot{\psi}_0 \\ v_0 \end{bmatrix} = v_0 \begin{bmatrix} \kappa_0 \\ 1 \end{bmatrix}, \tag{5}$$

describing the truck's angular and longitudinal velocity respectively. The angular and longitudinal velocities $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ of each trailer are calculated using a nonlinear transformation matrix $\boldsymbol{J}_i(\beta_i)$ multiplied by the velocity vector $\boldsymbol{w}_{i-1}$ in a recursive manner as [2]

$$\boldsymbol{w}_i = \boldsymbol{J}_i(\beta_i)\,\boldsymbol{w}_{i-1} = \begin{bmatrix} -\frac{m_{i-1}}{\ell_i}\cos\beta_i & \frac{1}{\ell_i}\sin\beta_i \\ m_{i-1}\sin\beta_i & \cos\beta_i \end{bmatrix} \boldsymbol{w}_{i-1}, \quad i = 1, 2\,. \tag{6}$$

By introducing an auxiliary matrix $\boldsymbol{\Gamma}_i(\beta_i) = \left(\boldsymbol{I}_{2\times 2} - \boldsymbol{J}_i(\beta_i)\right)$ and vector $\boldsymbol{c}^\top = [1 \;\; 0]$, the hitching angle dynamics is recursively given by

$$\begin{aligned} \dot{\beta}_i &= \omega_{i-1} - \omega_i = \boldsymbol{c}^\top(\boldsymbol{w}_{i-1} - \boldsymbol{w}_i) = \boldsymbol{c}^\top \boldsymbol{\Gamma}_i(\beta_i)\,\boldsymbol{w}_{i-1} \\ &= \boldsymbol{c}^\top \boldsymbol{\Gamma}_i(\beta_i) \left(\prod_{j=1}^{i-1} \boldsymbol{J}_j(\beta_j) \right) \boldsymbol{w}_0\,. \end{aligned} \tag{7}$$

The separate terms of the truck pose dynamics (4) and the hitching angle dynamics (7) can be summarized to the whole system model. For example, the model of the two-trailer combination in Fig. 1 is

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \dot{\boldsymbol{z}}_0 \\ \dot{\boldsymbol{\beta}} \end{bmatrix} = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{\psi}_0 \\ \dot{\beta}_1 \\ \dot{\beta}_2 \end{bmatrix} = \begin{bmatrix} v_0\cos\psi_0 \\ v_0\sin\psi_0 \\ v_0\kappa_0 \\ \boldsymbol{c}^\top \boldsymbol{\Gamma}_1(\beta_1)\,\boldsymbol{w}_0 \\ \boldsymbol{c}^\top \boldsymbol{\Gamma}_2(\beta_2)\,\boldsymbol{J}_1(\beta_1)\boldsymbol{w}_0 \end{bmatrix}. \tag{8}$$

In general, the state vector of the GnT system model is given by

$$\boldsymbol{x} = [\underbrace{x_0,\; y_0,\; \psi_0}_{\text{truck pose}},\; \underbrace{\beta_1,\; \ldots,\; \beta_n}_{\text{trailer}}]^\top \in \underbrace{\mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^n}_{\mathbb{X}}, \tag{9}$$

with the circular manifold set $\mathbb{S}^1 : \mathbb{R} \to (-\pi, \pi]$ and n as the number of attached trailers, i. e., in this work either $n = 0, 1, 2$. In case of no attached trailers, the model only consists of the truck pose. In order to obtain the pose of a specific trailer $\boldsymbol{z}_i$, the transformation

$$\boldsymbol{z}_i = \begin{bmatrix} x_i \\ y_i \\ \psi_i \end{bmatrix} = \boldsymbol{z}_0 - \sum_{j=1}^{i} \ell_j \begin{bmatrix} \cos\psi_j \\ \sin\psi_j \\ 0 \end{bmatrix} - \sum_{j=0}^{i-1} m_j \begin{bmatrix} \cos\psi_j \\ \sin\psi_j \\ 0 \end{bmatrix} - \sum_{j=1}^{i} \begin{bmatrix} 0 \\ 0 \\ \beta_j \end{bmatrix} \tag{10}$$

is used.

The resulting system model is nonholonomic, which means that taking into account rolling without slipping, movement is possible only in the direction of tires. This makes lateral repositioning particularly challenging, especially with trailers. Furthermore, the system dynamics is stable in the forward moving direction ($v_0 > 0$) for all state combinations, but for the equilibria $\boldsymbol{\beta} = \boldsymbol{0}$ the system dynamics is unstable in the backward moving direction ($v_0 < 0$). With incorrect actuation, this will unavoidably lead to the so-called *jackknife effect*, in which the trailers collapse. More in-depth details about the system properties, especially effects of different hitching offsets, are discussed in [2,7].

The vehicle model is also subject to physical limitations. The admissible set of the state space $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{X}$ is restricted by the maximum trailer angles $|\beta_i| \le \beta_{i,\max}$. Moreover, the control variables $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^2$ are also limited by maximum velocity $v_{0,\min} \le v_0 \le v_{0,\max}$ and maximum steering angle $|\delta_0| \le \delta_{0,\max}$.

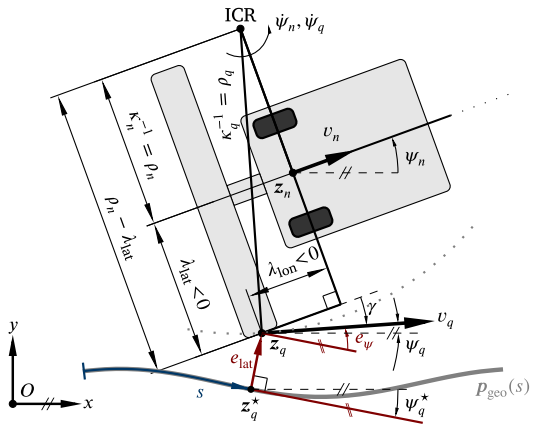


Fig. 2. Transformation from the center of the rearmost axle $\boldsymbol{z}_n$ to the guidance point $\boldsymbol{z}_q$. Velocity v_q is the tangent of the resulting dotted curve around the instant center of rotation (ICR). Illustration of the Cartesian reference curve $\boldsymbol{p}_{\text{geo}}(s)$ as well as the lateral and angular error, e_{lat} and e_ψ respectively, from the path to the guidance point.

2.2. Output transformation to the guidance point (GDP)

In numerous applications, particularly in the agricultural sector, the desired motion is not specified for the midpoint of a certain axle. Instead it is targeted towards a geometric point, such as the right edge of a mower or boom sprayer, referred to as the *guidance point* (GDP) in this paper. In general, the velocity v_q of such a guidance point differs in direction and magnitude to that of the axle midpoint v_n [37]. It is therefore not possible to move the guidance point tangentially to the axle on the path, but with a drift angle γ between axle movement v_n and the GDP movement v_q, see Fig. 2.

Hence, the guidance point $\boldsymbol{z}_q = [x_q, y_q, \psi_q]^\top$ is expressed as a pose, with the angle ψ_q indicating the direction of the point's velocity v_q and is calculated by

$$\boldsymbol{z}_q = \begin{bmatrix} x_q \\ y_q \\ \psi_q \end{bmatrix} = \boldsymbol{z}_n + \begin{bmatrix} \cos\psi_n & -\sin\psi_n & 0 \\ \sin\psi_n & \cos\psi_n & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_{\text{lon}} \\ \lambda_{\text{lat}} \\ \gamma \end{bmatrix}, \tag{11}$$

with the longitudinal and lateral displacements $(\lambda_{\text{lon}}, \lambda_{\text{lat}}) \in \mathbb{R}$. Note that the guidance point always refers to the final axis $\boldsymbol{z}_n$ of the combination, enabling precise positioning of trailers with booms, for example. The velocity drift angle γ is a result of the spatial displacement and may be derived by geometrical consideration as [37]

$$\gamma = \arctan \frac{\lambda_{\text{lon}}\kappa_n}{1 - \lambda_{\text{lat}}\kappa_n}, \tag{12}$$

with the motion curvature κ_n of the rearmost axle, which is calculated recursively [18]. The transformation between κ_n and the guidance point's motion curvature κ_q is given by [18]

$$\kappa_n = \frac{\kappa_q}{\sqrt{1 - (\lambda_{\text{lon}}\kappa_q)^2} + \lambda_{\text{lat}}\kappa_q}, \tag{13}$$

where κ_q is considered as the main objective and thus can be seen as the desired curvature given by the path to be followed.

The objective variables of the guidance point and the vehicle–trailer combination are mapped in the *system output vector*

$$\boldsymbol{y} = \underbrace{[x_q,\; y_q,\; \psi_q}_{\text{GDP output}},\; \underbrace{v_0,\; \delta_0,\; \beta_1, \ldots, \beta_n]}_{\text{vehicle-trailer output}} \in \underbrace{\mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^{2+n}}_{\mathbb{Y}} \tag{14}$$

whereby all transformations are performed using the transformation map $\boldsymbol{h} : \mathbb{X} \times \mathbb{U} \to \mathbb{Y}$ as $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$.

3. Nonlinear model predictive path-following control (PF-MPC)

In this section, the nonlinear model predictive path-following control formulation is presented and a brief summary about the path planning algorithm is given.

3.1. Model predictive path-following formulation

The main objective of the path-following controller is to guide the vehicle–trailer combination with the guidance point in a predefined direction along a given Cartesian reference curve $\boldsymbol{p}_{\text{geo}}$ in the (x, y)-plane, see Fig. 2. In contrast to a trajectory tracking formulation, path-following offers a decoupling of timing demands. This simply means that the controller itself may decide where to be at which time. This property is especially important when reversing with trailers. A strict time coupling would not grant the controller the flexibility to compensate for tracking deviations which could lead to a collapse of the vehicle–trailer combination.

The reference curve $\boldsymbol{p}_{\text{geo}}$ can be augmented with additional information to form the reference path for the output (14) as

$$\boldsymbol{p}(s) = \Big[\underbrace{x_q^\star,\ y_q^\star,\ \psi_q^\star}_{\text{GDP reference}},\ \underbrace{v_0^\star,\ \delta_0^\star,\ \beta_1^\star,\ \dots,\ \beta_n^\star}_{\text{vehicle-trailer reference}}\Big]^\top = \boldsymbol{y}^\star(s)\,, \tag{15}$$

where all desired outputs are denoted as $(\cdot)^\star$ and n is the number of attached trailers. The path $\boldsymbol{p} : [0, s_{\text{G}}] \to \mathbb{Y}$ is parameterized by the path parameter s, with the goal position $s_{\text{G}} \in \mathbb{R}^+$ at the end of the path and s representing the arc-length of the Cartesian reference curve. The course angle $\psi_q^\star = \operatorname{atan2}\big(\frac{\mathrm{d}}{\mathrm{d}s} y_q^\star, \frac{\mathrm{d}}{\mathrm{d}s} x_q^\star\big)$ defines the desired orientation of the vehicle–trailer combination and indicates together with the velocity assignment $v_0^\star$ the desired driving direction. The velocity assignment may also be used to demand velocity profiles. To ensure that the front wheels are parallel when stationary, steering assignment $\delta_0^\star$ is required during halt process as boundary condition. Furthermore, the trailer angles $\beta_i^\star$ may be specified as a reference and to achieve a precise trailer alignment when stopping.

The path may originate from a global path planning such as [39] and may be discretized and therefore consist of point-wise samples. A major advantage of the chosen concept is that the Cartesian reference curve does not necessarily have to be fully kinematically feasible for the system (8). Moreover, it may also be sufficient to specify only the Cartesian reference coordinates $[x_q^\star, y_q^\star]^\top$ of the desired path and omit all other reference outputs, unless proper initialization such that the driving direction is not ambiguous. Since the chosen model predictive approach operates locally and suboptimally (cf. Section 4.1), it is advantageous to specify as many reference outputs as possible in order to avoid the risk of running into local minima, especially if $n > 0$ and $[\lambda_{\text{lon}}, \lambda_{\text{lat}}]^\top \neq \boldsymbol{0}$.

It is essential to know the current position relative to the reference, i. e., the orthogonal projection of the guidance point to the Cartesian reference curve for calculating the tracking error at any time. However, solving the projection problem

$$s_\perp = \arg\min_s \left\| \begin{bmatrix} x_q(t) \\ y_q(t) \end{bmatrix} - \begin{bmatrix} x_q^\star(s(t)) \\ y_q^\star(s(t)) \end{bmatrix} \right\|_2 \tag{16}$$

is generally time consuming. Therefore, the path parameter s is incorporated as an auxiliary part of the optimization problem to obtain a timing law $t \mapsto s(t)$ and consider the path position as an additional degree of freedom. To this end, the dynamics of the path variable is modeled as an integrator whose input, i. e., the path velocity, is used as a virtual control input v as [40]

$$\dot{s} = v\,. \tag{17}$$

In order to incorporate rate limits for the inputs (1) in the continuous-time MPC formulation, the system is extended by integrators for the inputs, which is common practice in the MPC literature [41]. Then, the virtual inputs are defined as

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \dot{v}_0 \\ \dot{\delta}_0 \\ \dot{s} \end{bmatrix} = \begin{bmatrix} a \\ \varphi \\ v \end{bmatrix}, \tag{18}$$

with the truck's longitudinal acceleration a, the steering rate φ and the path velocity v. Consequently, the internal state vector of the MPC is composed of

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \\ s \end{bmatrix}. \tag{19}$$

Taking (18) and (19), introducing $\tilde{\boldsymbol{f}} : \mathbb{X} \times \mathbb{U} \times \mathbb{R} \to \mathbb{X} \times \mathbb{U} \times \mathbb{R}$ by using (8) and following [18], the optimal control problem (OCP) is formulated as

$$\min_{\tilde{\boldsymbol{u}}} \quad J(\tilde{\boldsymbol{u}}; \tilde{\boldsymbol{x}}_k) = V\big(\tilde{\boldsymbol{x}}(T_{\text{hor}})\big) + \int_0^{T_{\text{hor}}} l\big(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)\big)\,\mathrm{d}\tau \tag{20a}$$

$$\text{s. t.} \quad \dot{\tilde{\boldsymbol{x}}}(\tau) = \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau))\,, \quad \tilde{\boldsymbol{x}}(0) = \tilde{\boldsymbol{x}}_k \tag{20b}$$

$$\tilde{\boldsymbol{x}}(T_{\text{hor}}) \in \tilde{\mathcal{X}}_T \tag{20c}$$

$$\tilde{\boldsymbol{x}}(\tau) \in \tilde{\mathcal{X}} \tag{20d}$$

$$\tilde{\boldsymbol{u}}(\tau) \in \tilde{\mathcal{U}} \tag{20e}$$

and the control trajectory $\tilde{\boldsymbol{u}}(\tau)$ is optimized by solving the OCP over the horizon $\tau \in [0, T_{\text{hor}}]$ in each sampling instance $t_k = k\,\Delta t_{\text{MPC}}$, $k \in \mathbb{N}$, with the initial state $\tilde{\boldsymbol{x}}_k$, the terminal costs $V\big(\tilde{\boldsymbol{x}}(T_{\text{hor}})\big)$ and integral costs $l\big(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)\big)$, as well as the state and control constraints (20d) respectively (20e). For stopping maneuvers, terminal regions (20c) in combination with shrinking horizon are activated, which will be discussed in more detail in Section 3.3.

3.2. Control objective and constraint formulation

All control objectives of PF-MPC are incorporated into the terminal and integral costs, which are to be minimized. The integral costs

$$l(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}) = \|\boldsymbol{e}(\boldsymbol{y}, s)\|_{\boldsymbol{Q}}^2 + \left\| \begin{bmatrix} a \\ \varphi \end{bmatrix} \right\|_{\boldsymbol{R}}^2 + r_v (v - v_0^\star)^2 + q_s\, \phi(s) \tag{21}$$

comprise the penalization of tracking error metric $\boldsymbol{e}(\boldsymbol{y}, s)$ and virtual control inputs $\tilde{\boldsymbol{u}}$, along with a function $\phi(s)$ that penalizes the distance of the path parameter to the goal. For concise representation, the weighted squared quadratic norm $\|\boldsymbol{x}\|_{\boldsymbol{Q}}^2 = \boldsymbol{x}^\top \boldsymbol{Q} \boldsymbol{x}$ is used. The weighting matrices $\boldsymbol{Q} \in \mathbb{R}^{(5+n)\times(5+n)}$ and $\boldsymbol{R} \in \mathbb{R}^{2\times 2}$ are both diagonal and positive semi-definite, whereas q_s and r_v are scalar weighting factors, as these belong to the auxiliary quantities.

The error metric function $\boldsymbol{e} : \mathbb{X} \times \mathbb{U} \to \mathbb{Y}$ consists of the tracking error between outputs and path, i. e., $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) - \boldsymbol{p}(s)$. However, for the spatial error, it is more convenient to use a relative error, i. e., longitudinal and lateral instead of the Cartesian. This is done by rotating the Cartesian error around the path orientation yielding the so-called Frenet frame

$$\begin{bmatrix} e_{\text{lon}} \\ e_{\text{lat}} \end{bmatrix} = \begin{bmatrix} \cos\psi_q^\star & \sin\psi_q^\star \\ -\sin\psi_q^\star & \cos\psi_q^\star \end{bmatrix} \begin{bmatrix} x_q - x_q^\star \\ y_q - y_q^\star \end{bmatrix}. \tag{22}$$

Note that the longitudinal error can be reduced by adjusting the path parameter s. Based on the above, the tracking error metric is given by

$$\boldsymbol{e}(\boldsymbol{y}, s) := \begin{bmatrix} e_{\text{lon}} \\ e_{\text{lat}} \\ e_\psi \\ e_v \\ e_\delta \\ e_{\beta,1} \\ \vdots \\ e_{\beta,n} \end{bmatrix} = \begin{bmatrix} (22) \\ (22) \\ \measuredangle(\psi_n + \gamma^\star - \psi_q^\star) \\ v_0 - v_0^\star \\ \delta_0 - \delta_0^\star \\ \beta_1 - \beta_1^\star \\ \vdots \\ \beta_n - \beta_n^\star \end{bmatrix} \in \mathbb{Y}\,, \tag{23}$$

where $\angle(\alpha) := \{\alpha \in \mathbb{R} \mid \angle(\alpha) \mapsto (-\pi, \pi]\}$ denotes the mapping of periodic angles to circular manifolds $\mathbb{S}^1$ using the modulo operation. For the orientation angle error the following heuristic is used. Instead of calculating the velocity drift angle γ throughout the vehicle-trailer combination, the angle $\gamma^\star$ is calculated in reverse by using the reference curve's curvature $\kappa_q^\star = \frac{\mathrm{d}}{\mathrm{d}s}\psi_q^\star$. This is done for two reasons: firstly, computation time is saved as recursive calculation is avoided (cf. the concept of virtual steering wheels in [5]) and, more importantly, a recursive calculation may be ambiguous due to several degrees of freedom when having trailers attached. It is important to note that the tracking of specific outputs may be deactivated by setting the corresponding weights to zero.

The penalization of the goal distance, i. e., the progress error, requires a different approach, as the relative weighting between tracking and progress error is relevant. With quadratic weighting, high tracking errors might be accepted at the beginning of a long path to quickly reduce the progress error. Therefore, it is preferable to use linear weighting such that the gradient of the cost does not depend on the remaining path length. However, near the goal position, it is in fact the other way round: without quadratic penalization, it might come to oscillations. For this reason the function similar to [42]

$$\phi(s) = \begin{cases} 2\left|s - s_\mathrm{G}\right| - S_\mathrm{G} & \text{if } \left|s - s_\mathrm{G}\right| > S_\mathrm{G} \\ (s - s_\mathrm{G})^2 / S_\mathrm{G} & \text{else} \end{cases} \tag{24}$$

is applied, which switches to the quadratic penalization in a region S_G near the goal s_G. The function is chosen in such a way that the gradient is continuous on the switching point.

The braking distance resulting from the target velocity $v_0^\star$ and the maximum deceleration a_min is used to determine the target region as

$$S_\mathrm{G} = \frac{1}{2}\frac{\left(v_0^\star\right)^2}{\left|a_\mathrm{min}\right|}. \tag{25}$$

The terminal costs are formulated similar to (21), but without the virtual control input terms, i. e.,

$$V\big(\tilde{\boldsymbol{x}}(T_\mathrm{hor})\big) = \|\boldsymbol{e}(\boldsymbol{y}, s)\|_{\boldsymbol{P}}^2 + p_s\,\phi(s) \tag{26}$$

and are useful especially if there are large deviations. The weight matrix $\boldsymbol{P} \in \mathbb{R}^{(5+n)\times(5+n)}$ and weight factor p_s are also defined the same way as for the integral costs.

Finally, it should be noted that both control and state constraints are formulated as box constraints, with

$$\tilde{\mathcal{X}} = \left\{ \begin{array}{l|rl} \boldsymbol{x} \in \mathbb{X} & v_0 & \in [\ v_{0,\mathrm{min}}, v_{0,\mathrm{max}}] \\ \boldsymbol{u} \in \mathbb{U} & \delta_0 & \in [-\delta_{0,\mathrm{max}}, \delta_{0,\mathrm{max}}] \\ s \in \mathbb{R} & \beta_i & \in [-\beta_{i,\mathrm{max}}, \beta_{i,\mathrm{max}}] \\ & s & \in [0, s_\mathrm{G}] \end{array} \right\} \tag{27}$$

representing state constraints (20d) and

$$\tilde{\mathcal{U}} = \left\{ \tilde{\boldsymbol{u}} \in \mathbb{R}^3 \left| \begin{array}{rl} a & \in [\ a_\mathrm{min}, a_\mathrm{max}] \\ \varphi & \in [-\varphi_\mathrm{max}, \varphi_\mathrm{max}] \\ v & \in [\ v_\mathrm{min}, v_\mathrm{max}] \end{array} \right. \right\} \tag{28}$$

representing control constraints (20e).

3.3. Performing stopping and turnaround maneuvers

Using the formulation for the PF-MPC introduced in (20), the controller operates asymptotically. While this formulation is suitable for the driving task, it is not appropriate for stopping maneuvers. To stop accurately and within a specified time, it is preferable to shorten the horizon time linearly in each PF-MPC sampling step k as

$$T_{\mathrm{hor},\,k+1} = \max\left\{T_{\mathrm{hor},\,k} - \alpha\,\Delta t_\mathrm{MPC}\ ,\ T_{\mathrm{hor,\,min}}\right\}, \tag{29}$$

with the time decrease factor $\alpha \in \mathbb{R}_0^+$ and the parameter $T_{\mathrm{hor,\,min}} > 0$ during the stopping process. This method, known as *shrinking horizon* [43], allows to reach the goal pose in finite time. Simulation-based analysis has shown that when the horizon time is optimized, it decreases by a roughly constant factor $\alpha = 0.75$, which was also chosen for the experiments.

To enhance the stopping accuracy, specific outputs are subjected to terminal regions (20c) formulated as

$$\tilde{\mathcal{X}}_T = \left\{ \tilde{\boldsymbol{x}} \in \mathbb{X} \times \mathbb{U} \times \mathbb{R} \;\middle|\; V\big(\tilde{\boldsymbol{x}}(T_\mathrm{hor})\big) \le \Omega \right\}, \quad \Omega > 0, \tag{30}$$

with a terminal cost threshold Ω, which are activated during the halting process. The result is a highly precise stop.

Aside from stopping, turning maneuvers pose the greatest challenge. This process requires precise stopping in order to correctly pre-position the vehicle and trailer before attempting the turn. Therefore, it is reasonable to divide a path based on the driving directions and allow PF-MPC to handle each segment with a single driving direction at a time. This allows to use the aforementioned techniques when executing a turn and achieving a precise halt before altering direction.

3.4. Global path planning

While the results in [18] show that the reference path does not necessarily have to be kinematically feasible, a valid initial solution for the maneuvering problem is beneficial in terms of tracking performance and helps to prevent obstacle collisions.

Therefore, the hierarchical planner presented in [44] is applied. It first uses a lattice planner based on a discretized state space representation to simplify the motion planning problem to a graph search problem. In it, several trajectory segments are combined, which are calculated in advance using an inversion-based planning algorithm [31].

The resulting global solution is complemented by a time-optimal velocity profile and finally optimized using a model predictive optimization scheme, which can be executed either in advance or optionally during the execution phase [36,39].

4. Implementation of the PF-MPC algorithm on an MCU

This section will address some practical considerations for successfully implementing the MPC algorithm on an embedded microcontroller (MCU).

4.1. Numerical solution and implementation aspects

Solving nonlinear OCPs is computationally demanding compared to classical control techniques. This is especially challenging for microcontrollers such as those used on state-of-the-art electronic control units (ECU) in automotive applications, since their central processing units are magnitudes slower and have less memory available compared to personal computers.

For this reason, GRAMPC [43] is chosen as solver. It is a highly efficient software framework that is particularly suitable for execution on embedded hardware, such as MCUs. It is implemented entirely in C and provides an interface for MATLAB® and C++. The main algorithm is based on the augmented Lagrangian [45] approach which reformulates the OCP as a max–min-problem. The inner minimization problem is tackled by a tailored projected gradient method whereas the outer maximization loop is solved via steepest ascent. Using a fixed number of iterations, a suboptimal solution is rapidly provided at low computational costs and in deterministic time. Nevertheless, under certain conditions it can be shown that the solution improves over time [46].

To enhance computational efficiency, pre-calculation of path gradients is performed. In addition, equidistant path sampling is used to calculate the path index by dividing the actual arc-length by the path sampling step, avoiding the need for path array index search, e. g., using a binary search.

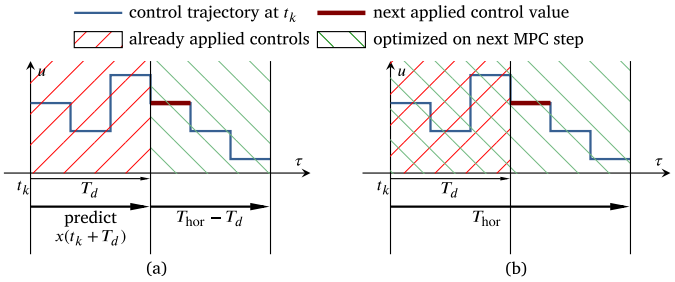


Fig. 3. Comparison of two delay time compensation methods at current time instant t_k. (a) Prediction of states after delay time and optimizing with shorter horizon time $T_{hor} - T_d$. (b) The used approach: zeroing the control input gradients for $\tau \in [0, T_{d,j}]$ and optimize whole T_{hor}.

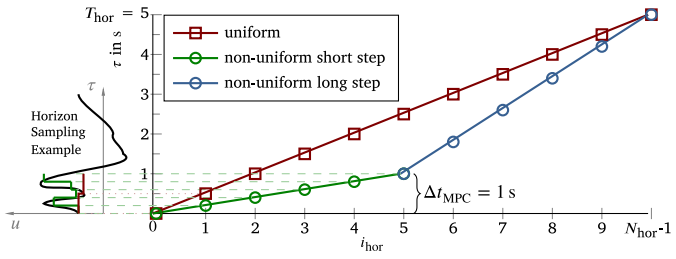


Fig. 4. Comparison of the MPC horizon sampling: In red, uniform discretization with equal step size. In green and blue, non-uniform discretization with two step sizes. Description in the text. All values are exemplary.

4.2. Delay time compensation

In real systems, dealing with delay times plays a quite important role. In order to compensate for these, they have to be taken into account during optimization. There are several methods to achieve a compensation, e. g., by predicting the state after the delay time $T_{d,j}, j = 1, \dots, \dim(\tilde{\boldsymbol{u}})$ and performing the optimization in the new state with a shorter horizon time, see Fig. 3a. However, the drawback is that this method is particularly suitable when all delay times are of the same length. Otherwise, a combination with other methods becomes necessary. Therefore, a method is used that is simple to implement using the gradient method, see Fig. 3b. The basic concept is that those control inputs that have already been applied to the system, but not actuated due to the delay time, will not be optimized in the next cycle [18]. This is performed by zeroing the gradients $\partial \tilde{\boldsymbol{f}} / \partial \tilde{\boldsymbol{u}}$ and $\partial l / \partial \tilde{\boldsymbol{u}}$ belonging to $\tilde{u}_j$ for $\tau < T_{d,j}$.

4.3. Sampling of the horizon time

When implementing computationally-intensive algorithms like MPC on limited hardware, it is crucial to consider timing carefully. If the algorithm's cycle time is not feasible for the real-time part, it is recommended to make sure that the MPC cycle time is an integer multiple of the real-time part's cycle time. This prevents the unnecessary post-processing of the calculated results. Due to the MPC's longer cycle time compared with the main firmware, it is apparent that the low-level control will follow the solution trajectory of the new MPC calculation until the next recalculation.

The crucial point here is the sampling of the prediction horizon. This is essential because the control variables have to be interpolated from the most recent solution to match the ongoing time progression as the vehicle is driving. To attain the maximum accuracy in the relevant regions, a sampling in a non-equally distributed manner has been applied, as can be seen in Fig. 4. Half of the samples are taken with short steps and discretize the horizon interval $[0, \Delta t_{MPC}]$. The remaining samples discretize the interval $(\Delta t_{MPC}, T_{hor}]$ with long steps.

4.4. Feedback strategy

When considering a feedback strategy, the overall control process illustrated in Fig. 5 is examined first. Before the actual control loop, a global path planning is performed. Based on the current pose of the vehicle–trailer combination and the desired pose, including necessary parameters such as desired velocity, vehicle dimensions, and environment, a kinematically valid path $\boldsymbol{p}(s)$ is calculated. Once the path has been calculated, the actual control loop begins. It consists of the PF-MPC, which can be referred to as high-level control, the trajectory tracking control for steering and velocity, which is referred to as low-level control, as well as the measurement and estimation unit and the vehicle. The PF-MPC is used for both trajectory planning and position control. The calculated steering and velocity trajectory is fed to the low-level controller. No subordinate low-level control is provided for position control as the vehicle moves at such a low velocity that it is unnecessary.

The question arises how the feedback for high- and low-level control should look like. Position measurement is trivial: since PF-MPC performs the position control, the measured position can be fed back directly. However, it is not as straightforward when it comes to velocity and steering angle. This is because both PF-MPC and low-level control need the signals to be fed back. Nevertheless, closing the feedback loop twice should be avoided [47]. Consequently, the PF-MPC uses the predicted velocity and steering angle values, while the low-level control obtains the actual estimated values. The PF-MPC relies on the low-level control to achieve accurate tracking behavior. Significant deviations from the path are the only reason for the PF-MPC to be initialized with current values of velocity and steering angle.

4.5. Reference generation for low-level control

A further concern pertains to the reference variables of the low-level control, see *control strategy* in Fig. 5. The PF-MPC leverages derivatives to ensure the continuity of optimized velocity and steering angle trajectories, cf. (18). Nonetheless, they cannot serve directly as a reference variable for low-level control with this setup. A naive approach is to integrate the PF-MPC control trajectories over time. Since the PF-MPC has a notably slower cycle time and due to discretization errors, this approach may lead to instabilities, especially in the case of backward motion with trailers. Moreover, additional feedback and logic are necessary to ensure that the integrated reference variable adheres to the intended target variable and does not deviate. The fact that PF-MPC contains the required reference variables as states provides a simpler and more robust solution. These variables can be extracted from the predicted velocity and steering angle trajectories, with evaluation time given by

$$t_{eval} = t_{traj,act} + T_{d,j} + t_{ahead}\,, \quad j = \delta_0, v_0\,, \tag{31}$$

with the actual trajectory time $t_{traj,act}$ since last PF-MPC call, the delay time of the specific actuator $T_{d,j}$ and a constant look-ahead time t_{ahead} which is usually chosen as the cycle time of the low-level controller.

4.6. Computation times and memory resources

In the following, an overview of the required resources on the microcontroller (see Section 5.1.2) is given. Table 1 summarizes the computation times and the required RAM for each scenario from Section 6. The memory is divided into the needs of the solver (GRAMPC) and into the problem specific memory such as the parameters (OCP). It can be seen that the RAM mainly depends on the number of attached trailers. The computation times depend as well on the number of trailers, but also on whether the guidance point is outside the center of the rear axle, because then additional trigonometric calculations are necessary. All computation times are far below the MPC cycle time of 250 ms, so that real-time capability is guaranteed.

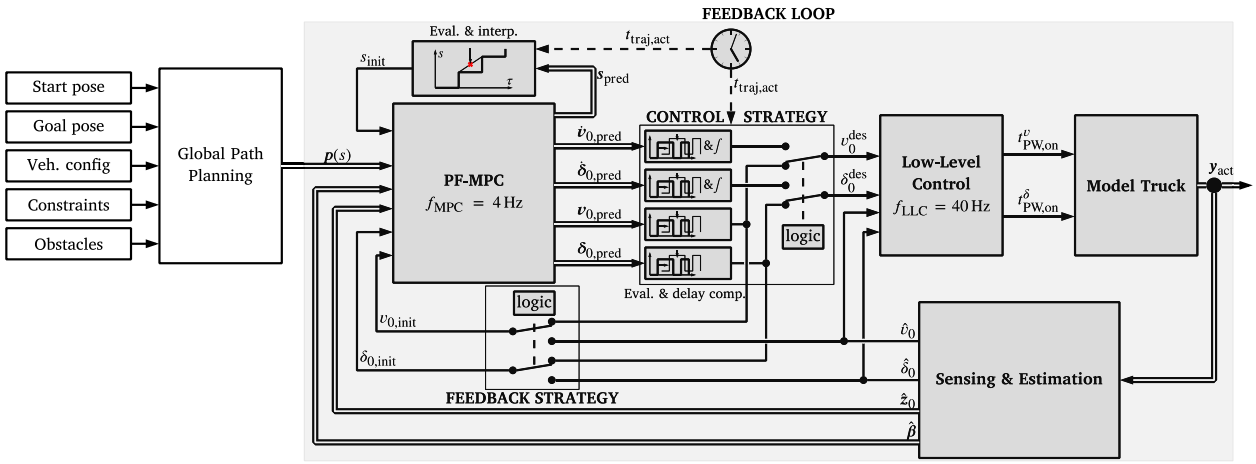


Fig. 5. MPC feedback and control variables strategy. All parts of the gray feedback loop are developed in this work.

Table 1
MPC computation times and RAM usage of each scenario in Section 6.

Scenario	I	II	III	IV	V	VI
Computation time in ms	43 ± 4	48 ± 7	138 ± 13	77 ± 1	106 ± 2	112 ± 5
RAM GRAMPC in Byte	2988	2992	3308	3324	3636	3636
RAM OCP in Byte	208	208	240	240	272	272

5. Experimental setup

This section describes the whole experimental setup, i. e., the dimensions of the model truck, the hard- and software structure as well as state estimation and low-level control.

5.1. Model-scaled truck–trailer combination

Here, the details of the model truck are described, beginning with the mechanical and electrical specifications and concluding with an outline of the software structure.

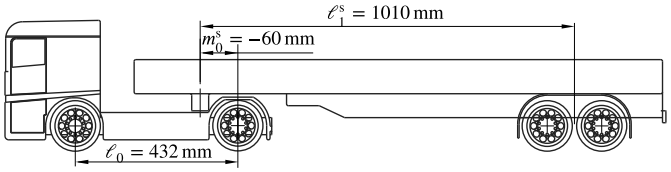
5.1.1. Mechanical setup

The experimental setup mainly consists of three parts: a tractor unit, a semi-trailer and a two-axle turntable trailer. Care was taken to ensure that the proportions of all parts correspond to those in full-scale so that the results can be transferred to real vehicles as far as possible. The tractor unit is a 1:8 model-scaled cab-over body truck with a steerable front axis and a rigid axle at the rear, see Fig. 6(a). An electric traction motor is mounted onto the rear axle with a reduction gearing which enables very low operating velocities between $5\,\mathrm{cm\,s^{-1}}$ and $60\,\mathrm{cm\,s^{-1}}$ for both driving directions. A digital servo motor on the front axis allows precise steering angle settings of maximum 33 deg which enables a wall-to-wall turning circle radius of 1.03 m.

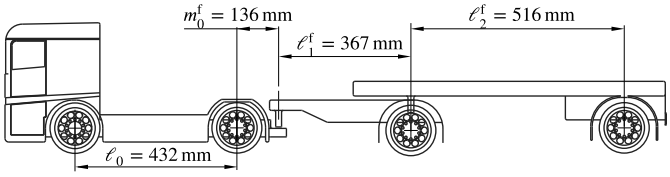
In the semi-trailer configuration, shown in Fig. 6(b), the hitching m_0^{s} is realized as a kingpin, which is positioned in front of the rear axle. This off-axle hitching gives the truck–trailer combination the specific characteristic to be less sensitive to the hitching angle β_1^{s} in reversing curves. Semi-trailers are commonly equipped with non-articulated multiple axles. This is represented by the chosen tandem axle. As the semi-trailer is mounted above the tractor unit's rear wheels, the hitching angle is therefore only limited by the cabin. However, due to cable routing, the maximum hitching angle $|\beta_{1,\max}^{\mathrm{s}}|$ is limited to about 80 deg.

The two-axle trailer combination in Fig. 6(c) consists of two one-axle trailers: the first trailer is a so-called *dolly trailer*. It consists of a drawbar ℓ_1^{f} and one axle with a coupling directly above it, i. e., $m_1^{\mathrm{f}} = 0$. The second trailer is similar to a semi-trailer which is coupled by a turntable bearing on the dolly's axle. Such two trailer combinations are also called *full trailer*. The coupling of the dolly to the tractor unit is different in contrast to the semi-trailer. Here, the drawbar of the dolly is mounted behind the rear axle of the tractor unit, which results in a distinct positive off-axle coupling m_0^{f}. The full trailer configuration is significantly more challenging to stabilize in reverse drive not only because of the second trailer, but especially due to the shorter drawbar lengths. For the sake of robustness, the maximum dolly trailer angle $|\beta_{1,\max}^{\mathrm{f}}|$ is set to 42 deg and the second trailer angle $|\beta_{2,\max}^{\mathrm{f}}|$ to 35 deg. More detailed considerations on the stability limits of two trailer systems can be found in [48,49]. All dimensions and parameters of both configurations described are summarized in Table 2.

(a) The model truck with the full trailer configuration.



(b) Dimensions of truck with semi-trailer configuration.



(c) Dimensions of truck with full trailer configuration.

Fig. 6. The experimental model-scaled truck and trailer combination.

Table 2
Dimensions and parameters of the model truck–trailer combination.

Tractor unit		Semi-trailer		Full trailer	
ℓ_0	0.432 m	m_0^s	−0.06 m	m_0^f	0.136 m
$\ell_{tw,0}$	0.320 m	ℓ_1^s	1.010 m	ℓ_1^f	0.367 m
r_{wheel}	0.052 m	$\beta_{1,max}^s$	±80 deg	$\beta_{1,max}^f$	±42 deg
$\delta_{0,max}$	±33 deg			m_1^f	0.000 m
$\dot{\delta}_{max}$	±15 deg/s			ℓ_2^f	0.516 m
$v_{0,max}$	±0.6 m s^{-1}			$\beta_{2,max}^f$	±35 deg
a_{max}	±1.00 m s^{-2}				

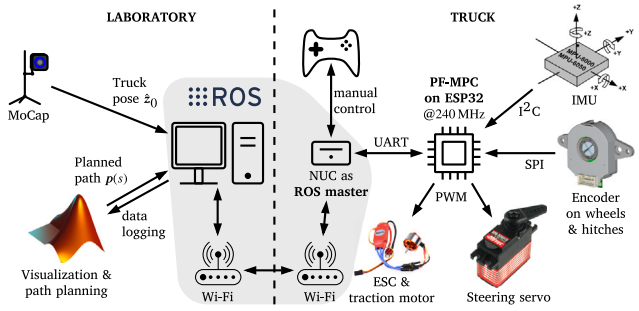


Fig. 7. Electrical setup of the truck and in the laboratory.

5.1.2. Electrical setup

In the following, the electrical setup of the truck and the laboratory will be described, which is shown in Fig. 7. One of the most important parts of autonomous driving is the exact tracking of the vehicle. In our setup, this is done using the camera-based motion capture (MoCap) system *OptiTrack Prime^x 13*, that provides sub-millimeter accuracies. To transfer the measured vehicle pose, the MoCap host PC is connected via Wi-Fi with the truck. For communication, the open-source robotics middleware *Robot Operating System* (ROS) is used. This allows a robust and flexible data exchange as well as the possibility to make measurements, to set parameters at runtime, to display the data in a variety of graphical views, and to connect to MATLAB®.

On the truck, there is also a Wi-Fi router installed and connected via LAN to an Intel® NUC mini PC. This serves as ROS master and coordinates all communication and also performs measurements.

The core and main controlling part of the truck is a self-made mainboard equipped with the 32-bit MCU *Espressif Systems ESP32 Transcilla Xtensa LX6* running on two cores @240 MHz. This MCU is chosen because of the straight forward usage and a performance comparable to state-of-the-art automotive MCU's like Infinion's AURIX™ 32-bit TC3xx MCU based on TriCore architecture.

The electric traction motor is a brushless DC motor (BLDC) connected to a closed-loop electronic speed control (ESC) unit. Therefore, only motor speed demands are possible and there is no influence on control time and quality. For adjusting steering angles, likewise a brushless servomotor in closed-loop is used, which only accepts direct angle inputs.

In addition to global positioning based on motion capture, the following sensors are installed on the truck. For measuring the truck's yaw rate, the six-axis inertial measurement unit (IMU) *TDK MPU-6050* is installed. Using yaw rate and actual truck velocity, the steering angle is determinable. Absolute encoders are used to precisely measure wheel spin and trailer angles. This is done with *CUI Devices AMT22*, which provide 4096 distinct positions per revolution.

In order to have an option for user-friendly maneuver control and visualization, MATLAB® is used on a separate PC in connection with the ROS Toolbox. Here, on the one hand, predefined scenarios and saved or generated maneuvers by means of path planning (for details about the planning algorithm see Section 3.4) may be transmitted to the truck's NUC via ROS. On the other hand, the positions measured via MoCap, as well as the states of the truck–trailer combination are visualized over time. To control the truck manually, a gamepad is wirelessly connected to the NUC, which sends control signals to the MCU via ROS.

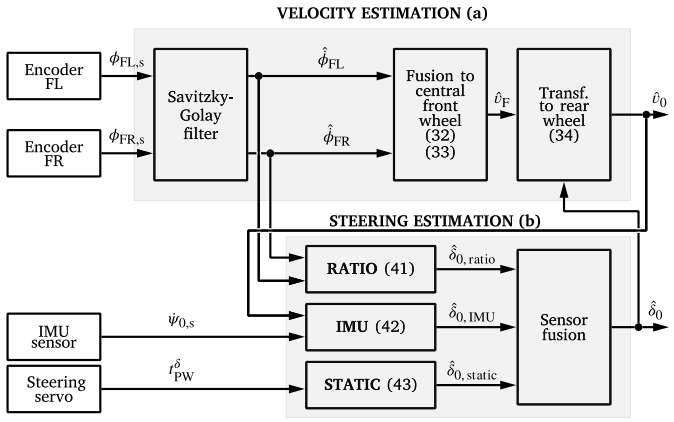


Fig. 8. Basic structure of velocity (a) and steering angle (b) state estimation.

5.1.3. Software setup

The software on the MCU for actuating the truck in real-time is written in C++. There are two tasks, each of which runs on one of the two cores: A real-time main task with 40 Hz on core 1 which handles input/output, state estimation, low-level control, and management of the solver algorithm GRAMPC. On core 2, the MPC task runs with 4 Hz completely independent from other parts of the MCU software and without interruption, yielding deterministic computation times well below the MPC cycle time.

5.2. State estimation and low-level control

In combination, state estimation and low-level control form the basis for reliable guidance of the model truck and truck–trailer combination. In theory, it is possible to obtain the required states such as position, velocity, steering angle and hitching angles via the highly accurate, global measurement of the MoCap. However, the following two reasons argue against the exclusive use of the MoCap. On the one hand, there is an unnecessarily long time delay due to filtering and data transmission from the MoCap to the truck via Wi-Fi. On the other hand, such a precise global measurement is usually not available in reality. Therefore, as commonly practiced, all quantities measurable by on-board sensors should be determined in real-time on the truck itself. This includes trailer angles β_i, steering angle δ_0 and truck velocity v_0, measured by encoders. MoCap is used to determine the position (x_0, y_0) and orientation ψ_0 of the truck only.

The fundamental structure for determining velocity (a) and steering angle (b) is shown in Fig. 8. The encoders on the two front wheels of the truck form the basis for both values. By applying derivative filters [50], the rotational speeds of the wheels are determined and from this, both the vehicle velocity and the steering angle are calculated. To further improve the steering angle information, the yaw rate of the IMU gyro and a static mapping between servo actuation and steering angle is used.

Although these components are not part of the main contribution, their importance and the positive results obtained warrant a thorough examination. In the following, a detailed explanation of the state estimation process for both velocity and steering angle, followed by a discussion of the low-level control strategies for these values is given. All quantities which are measured or estimated are denoted by $\hat{(\cdot)}$.

5.2.1. Velocity estimation using derivative estimators

To estimate the velocity $\hat{v}_0$ of the truck as accurately as possible, the rotational speeds of the two front wheels are determined although it is more elaborate than using the rear wheels. The reasons for this are the following: the rear axle is rigid, which means that the rear wheels are forced to slip in a curve. This distorts the velocity gained over those considerably. In addition, the rear axle is driven by the traction motor, which further increases the measurement inaccuracy while accelerating or decelerating.

Once the rotational speeds $\hat{\dot{\phi}}_{\text{wheel},i}$ of the two front wheels have been determined, they are translated into velocities using

$$\hat{v}_i = \hat{\dot{\phi}}_{\text{wheel},i}\, r_{\text{wheel}}\,, \quad i = \text{L}, \text{R}\,, \tag{32}$$

where r_{wheel} is the wheel radius. Both velocities are merged with

$$\hat{v}_\text{F} = \frac{\hat{v}_\text{L} + \hat{v}_\text{R}}{2} \tag{33}$$

to the front axle midpoint velocity $\hat{v}_\text{F}$. Thereafter, $\hat{v}_\text{F}$ is transformed to the rear axle velocity $\hat{v}_0$, as shown in Fig. 8, using

$$\hat{v}_0 = \hat{v}_\text{F} \cos\hat{\delta}_0\,. \tag{34}$$

Here, an estimate of the steering angle $\hat{\delta}_0$ is necessary, which will be described in Section 5.2.2. The following is a brief explanation of how to determine the front wheel rotational speeds from their measured absolute angles $\phi \in [0, 2\pi)$. The absolute encoders provide an integer value ϕ_s corresponding to an angle, i. e.,

$$\phi = \angle(\phi_\text{s}) := \{\phi_\text{s} \in [0, 4095] \mid \angle(\phi_\text{s}) \mapsto [0, 2\pi)\}\,. \tag{35}$$

Continuous rotation of a wheel therefore results in a sawtooth signal form, which has to be unwrapped, i. e., transformed to a continuous signal. In order to determine the angular velocity based on this signal, it is reasonable to use derivative estimators. Although the signal is noiseless, the angle signal has a slight nonlinearity, due to manufacturing tolerances of the wheels, which are not perfectly round. The resulting ripple in the derivative can be sufficiently suppressed using the filters. To this end, the Savitzky–Golay (SG) filter [51] is used, which smooths data based on local least-squares polynomial approximation.

On the model truck, the sampling of the encoders is not exactly equidistant, which has a negative effect on the calculation of derivatives if the samples are assumed to be equidistant. Therefore, a FIR formulation of the SG filter is implemented based on [50] that recalculates the coefficients of the filter in each cycle according to the sampling times stored in a ring buffer. In addition, the coefficients are chosen in such a way that the approximated polynomial is evaluated at the last sampling point, resulting in a derivative signal without time delay.

5.2.2. Steering angle estimation

Estimating the steering angle is far more difficult than estimating the velocity. This is due to the strong nonlinearities, wheel slippage, elasticity, backlash and hysteresis of the steering kinematics, as well as manufacturing and assembly tolerances. For this reason, it is beneficial to combine several methods of estimation in order to compensate for the disadvantages and improve the estimated value. With the sensors on board, the following approaches are possible, as shown in Fig. 8(b):

1. Wheel's angular speed ratio (dynamic)
2. Learning of steering characteristics (static and dynamic)
3. Yaw rate and velocity (dynamic)

Approach 1 utilizes the kinematic relationship between the front wheels and Ackermann steering kinematics to estimate the steering angle, as depicted in Fig. 9. When the vehicle is turning, the two front wheels follow distinct tracks, resulting in different angular velocities. By determining the speed ratio, a characteristic curve is derived, as illustrated in Fig. 10, covering the entire steering angle range. Remarkably, even with a rigid rear axle, the steering behavior of the model truck adheres closely to the Ackermann model.

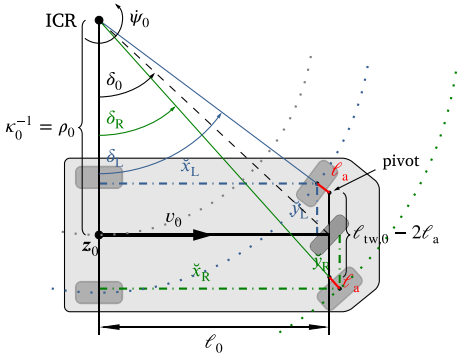


Fig. 9. Derivation of the Ackermann steering with A-arms ℓ_a and the front wheels rotating around the pivots.

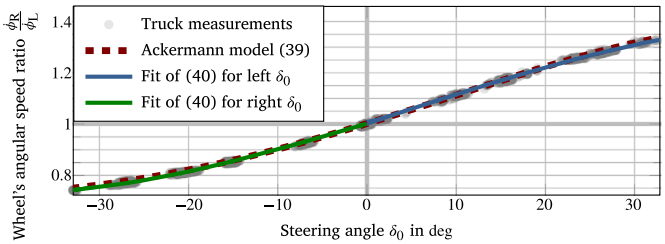


Fig. 10. Wheel's angular speed ratio vs. steering angle — Parameter fitting for separate negative and positive steering angles using the function (40) and showing the ideal Ackermann model (39).

To derive the rotational speeds of both wheels, it is assumed that the steering movement occurs at the pivot point, cf. Fig. 9. The wheels are attached to A-arms with a length of ℓ_a. To calculate the turning radii ρ_L and ρ_R, the steering angles of each front wheel must be calculated beforehand using

$$\delta_\text{L,R} = \arctan \frac{\ell_0\, \kappa_0}{1 \mp \left(\frac{1}{2}\ell_\text{tw,0} - \ell_\text{a}\right)\kappa_0}\,. \tag{36}$$

The turning radii can then be calculated using the resulting triangles formed by the A-arms by

$$\rho_\text{L,R} = \sqrt{\kappa_0^2\, \check{x}_\text{L,R}^2 + (1 \mp \kappa_0\, \check{y}_\text{L,R})^2} \tag{37a}$$

where $\check{x}_\text{L,R}$ and $\check{y}_\text{L,R}$ are defined as

$$\check{x}_\text{L,R} = \ell_0 \mp \ell_\text{a} \sin\delta_\text{L,R} \tag{37b}$$

$$\check{y}_\text{L,R} = \frac{1}{2}\ell_\text{tw,0} + \ell_\text{a}\left(\cos\delta_\text{L,R} - 1\right)\,. \tag{37c}$$

The wheel's rotational speeds can now be calculated using the relationship

$$\dot{\phi}_\text{L,R} = \rho_\text{L,R}\, \kappa_0\, \frac{v_0}{r_\text{wheel}}\,, \quad (\rho_\text{L,R}\, \kappa_0 \to 1)|_{\kappa_0 \to 0}\,. \tag{38}$$

Finally, the rotational speed ratio is computed as

$$\Upsilon = \frac{\dot{\phi}_\text{R}}{\dot{\phi}_\text{L}} = \frac{\sqrt{\kappa_0^2\, \check{x}_\text{R}^2 + (1 + \kappa_0\, \check{y}_\text{R})^2}}{\sqrt{\kappa_0^2\, \check{x}_\text{L}^2 + (1 - \kappa_0\, \check{y}_\text{L})^2}}\,. \tag{39}$$

In order to calculate the steering angle from (39), it must be solved with respect to κ_0, which unfortunately cannot be done analytically. Instead, the characteristic curve from Fig. 10, obtained through MoCap measurements with the model truck, is fitted in least-squares sense for both left and right steering angles using the function

$$\Upsilon = c_4^\delta \sin(c_5^\delta\, \hat{\delta}_{0,\text{ratio}}) + 1\,. \tag{40}$$

The inverse is then given by

$$\hat{\delta}_{0,\text{ratio}} = \frac{1}{c_5^\delta} \arcsin \frac{Y-1}{c_4^\delta} . \tag{41}$$

Thus, the steering angle $\hat{\delta}_{0,\text{ratio}}$ can be estimated if the wheel speeds are measured and the ratio $\hat{Y}$ is calculated. With this approach, the steering angle can be determined precisely, particularly for very low velocities and small steering angles. However, this method is valid only for low velocities, as long as the front wheels do not slip during turning and as long the steering angle is constant, i. e., $\varphi \approx 0$.

Approach 2 uses the measured yaw rate $\dot{\psi}_{0,\text{s}}$ of the IMU's angular rate sensor and the current vehicle velocity $\hat{v}_0$ to estimate the current steering angle $\hat{\delta}_{0,\text{IMU}}$. The third line of Eq. (4) yields

$$\hat{\delta}_{0,\text{IMU}} = \arctan \ell_0 \frac{\dot{\psi}_{0,\text{s}}}{\hat{v}_0} . \tag{42}$$

There are two drawbacks to this method: On the one hand, a minimum velocity of the vehicle is required to avoid a singularity in Eq. (42), and indeed the truck operates typically at very low velocities. Secondly, the yaw rate sensor may drift over time, which requires calibration. In addition, the sensor signal is typically noisy and affected by external influences such as vibrations and shocks. However, with appropriate filtering, it is possible to estimate a reliable steering angle.

Approach 3 involves learning the relationship between the commanded servo angle and the resulting steering angle. This requires the exact tracking of the truck via MoCap in order to determine the resulting curvature of the truck. As the steering is strongly dependent on vehicle speed and direction due to elasticity, backlash and hysteresis, learning the entire characteristic is very effortful. The obtained characteristic is solely employed for standstill since it is the only means of acquiring steering angle information.

This is done by measuring the truck's position via MoCap at different but constant steering angles and velocities. The resulting curvatures and thus steering angles are calculated from these measurements. Finally, the coefficients c_1^δ, c_2^δ and c_3^δ of the function

$$\hat{\delta}_{0,\text{static}} = c_1^\delta \arctan c_2^\delta (t_{\text{PW}}^\delta - c_3^\delta) \tag{43}$$

are gained by solving the nonlinear curve fitting problem in a least-squares sense, separated for forward and backward motion. Using the function (43), the actual standstill steering angle is calculated passing the current commanded steering servo PWM on time $t_{\text{PW,act}}^\delta$.

In the final step, the three estimates of the steering angle values need to be merged. Mean averaging the values is the simplest approach, but it does not consider the variance of each estimate. To address this, it is recommended to use a Kalman filter [52], accounting for state-dependent variances. As process model, a first order delay is used alongside with the static characteristic curve $\hat{\delta}_{0,\text{static}}$ as input. The measurements are $\hat{\delta}_{0,\text{ratio}}$ and $\hat{\delta}_{0,\text{IMU}}$. Fig. 11 illustrates the variances on the top and the root-mean-square error (RMSE) on the bottom with respect to the whole steering angle domain for each measurement and both merging methods. It can be seen that using a Kalman filter for sensor fusion is superior to simple mean averaging in almost all cases.

5.2.3. Low-level control

To precisely track commands from high-level control (PF-MPC) on the actuators at the lowest level (steering servo and traction motor), a low-level control for each actuator is necessary. Moreover, the demands have to be converted into a compatible control signal for the respective actuators. In our case, the actuators are a digital steering servo and an electronic speed controller for the brushless DC motor. In both cases, the control value is a PWM signal with a fixed frequency, whereby the desired value is coded by means of the pulse width time of $t_{\text{PW,min}} = 1\,\text{ms}$ and $t_{\text{PW,max}} = 2\,\text{ms}$ on a linear scale between 0 % and 100 % respectively. In terms of the steering servo, the center position is 50 %, while full left is 0 % and full right is 100 %. The motor controller operates with 50 % standstill, 0 % full reverse and 100 % full forward.

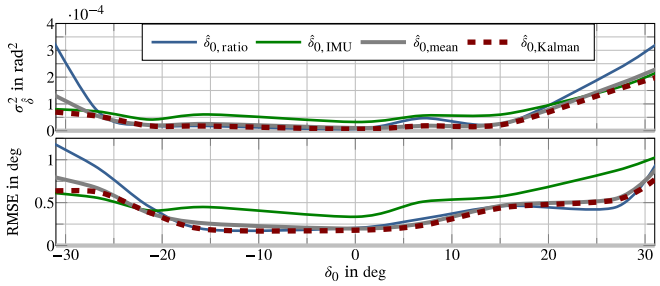


Fig. 11. Evaluation of the steering angle estimation. The upper plot shows the variance of each estimation and fusion method over the entire steering angle range. The lower plot shows the root mean square error, i. e., $\text{RMSE}(\hat{\theta}) = \text{E}((\hat{\theta}-\theta)^2)^{1/2}$. The velocity dimension is not shown for the sake of simplicity.

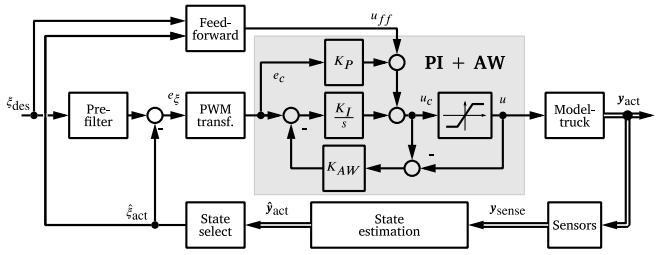


Fig. 12. Basic structure of the low-level controller for velocity and steering angle.

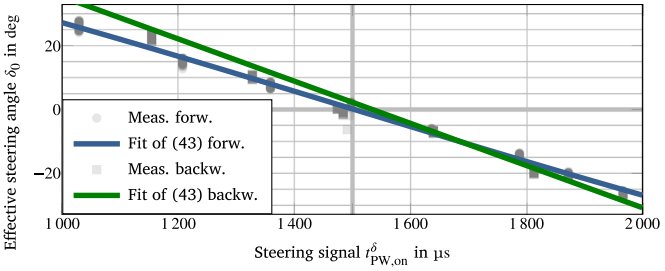


Fig. 13. Effective steering angle vs. steering servo command — Parameter fitting using function (43).

Each of the two actuators is controlled by a PI controller with anti-windup using back-calculation technique [53] and a feedforward component, illustrated in Fig. 12. The feedforward control is essentially the most important part, because it contains considerable system knowledge so that the controller only has to compensate for deviations.

Implementing the feedforward control for steering is relatively straightforward in this context. Given the model truck's exceptionally low velocities, the feedforward design assumes minimal impact of driving speed on steering behavior. The primary task involves establishing the relationship between the instructed servo angle and the resulting effective steering angle, derived from MoCap's position measurements and the calculation of the driven curvature, which is illustrated in Fig. 13. By employing these datasets, the coefficients $c_1^\delta, c_2^\delta, c_3^\delta$ in the function (43) are determined using least-squares optimization. The inverse of (43), i. e.,

$$t_{\text{PW,ff}}^\delta = c_3^\delta + \frac{1}{c_2^\delta} \tan \frac{\delta_0^{\text{des}}}{c_1^\delta} \tag{44}$$

is then employed as static feedforward control input.

In case of velocity control, it is more challenging to design reasonable feedforward control. The truck's velocity is significantly influenced by the current steering angle position, induced by increased load of the turned wheels. To prevent the PI controller from compensating for

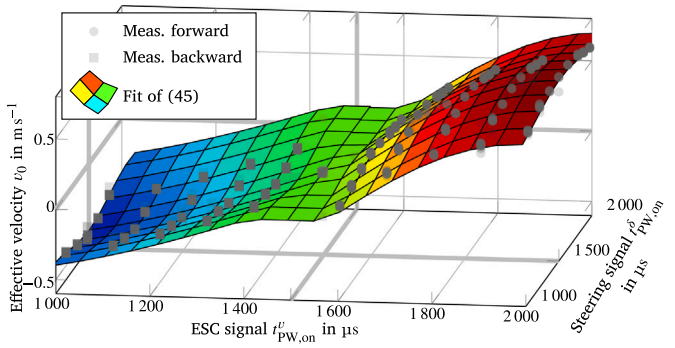


Fig. 14. Effective velocity vs. motor controller command — Parameter fitting using (45) separately for forward and backward drive. The influence of the steering angle on the velocity is clearly visible in the form of a convexity.

this load, it is advisable to model this behavior in the open-loop path. Therefore, the effect of the steering on the speed is determined again by precise position measurements via MoCap. This assessment is made for several setpoints of the ESC and steering angles. The result is a two-dimensional relationship, shown in Fig. 14, and may be modeled by

$$\begin{aligned} v_0^{\mathrm{eff}} = c_1^v + c_2^v \cos\left(c_3^v \left(t^\delta_{\mathrm{PW}} - t^\delta_{\mathrm{PW,zero}}\right)\right) \\ + c_4^v \sin\left(c_5^v \left(t^v_{\mathrm{PW}} - t^v_{\mathrm{PW,zero}}\right)\right) \end{aligned} \tag{45}$$

with coefficients c_1^v to c_5^v which are also determined using least-squares optimization separated in forward and backward drive. The feedforward control value for the ESC is obtained using the inverse of (45) as

$$t^v_{\mathrm{PW,ff}} = t^v_{\mathrm{PW,zero}} + \frac{1}{c_4^v} \arcsin \frac{v_0^{\mathrm{des}} - c_1^v \cos\left(c_2^v \left(t^\delta_{\mathrm{PW,act}} - t^\delta_{\mathrm{PW,zero}}\right)\right)}{c_3^v}. \tag{46}$$

6. Experimental validation

To test the described algorithms, including PF-MPC, low-level control, and state estimation, and to showcase their performance, this section features driving scenarios using the model truck and either a semitrailer or turntable trailer introduced in Section 5.1.1. The experiments are performed in a 6.8 m by 7.5 m laboratory as can be seen in Fig. 15.

Each experiment is presented visually through a four segment figure as shown in Figs. 16 to 21. The left segment displays a top view of the laboratory, which includes the divided path for forward $\boldsymbol{p}_{\mathrm{fw}}(s)$ and backward $\boldsymbol{p}_{\mathrm{bw}}(s)$ directions. Furthermore, the top view contains multiple snapshots of the truck–trailer combination, which includes the MPC predictions. Key points are marked with ⓒ.

Three time graphs on the right display the following information: The top graph shows the MPC commanded velocity v_0^{mpc}, the reference of the low-level velocity control v_0^{des}, and the current velocity v_0^{act}. Likewise, the middle graph exhibits the MPC commanded steering angle δ_0^{mpc}, the reference of the low-level steering control δ_0^{des}, and the current steering angle δ_0^{act}. The hitch angles β_i of trailers are also included. The bottom graph shows the lateral e_{lat} and angular e_ψ errors of the guidance point.

All parameters used during the experiments are summarized in Table 3, whereby the number of n (kinematically) attached trailers are stated as GnT. The MPC horizon discretization is equal for all scenarios with $N_{\mathrm{hor}} = 11$ as well as the control cost weights with $r_a = r_\varphi = r_v = 0.1$. The horizon length T_{hor} is mainly dependent on the number

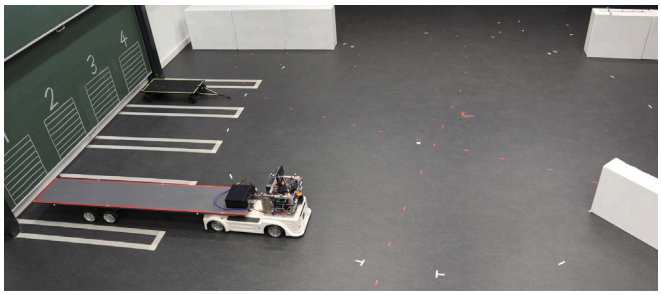


Fig. 15. The laboratory setup: Model truck with the semi-trailer attached and the full trailer on docking slot 4.

of trailers and the direction of travel, so that driving backwards with trailers has a longer horizon time than driving forwards. In addition, the admissible velocity for reversing is reduced compared to driving forwards. The weighting of the lateral distance to the path can be given a high value in forward movement, since the main goal is to stay close to the track. In reverse, however, a balanced weighting in addition to the angle error and the trailer angles is necessary to keep the truck–trailer combination stable. The path progress weighting should always be kept small, as it is only needed as artificial progress benefit, otherwise curves will be cut excessively. The terminal cost weights are needed almost exclusively for reversing, on the one hand for stabilization and on the other hand for precise stopping.

6.1. Experiments without trailer (G0T)

In the first scenario, the truck performs a turning maneuver and approaches a trailer with the intention of attaching. In the second scenario, the vehicle is equipped with an agricultural implement, such as a mower.

6.1.1. Scenario I (G0T): "coupling of a trailer"

With this scenario shown in Fig. 16, the aim is to illustrate a common use-case: a truck needs to attach a trailer at a depot to haul it later. Since the objective is to reach the trailer drawbar with the truck's hitch, it is selected as the guidance point.

The scenario starts with a slight deviation from the reference path at marker ⓐ in the forward direction towards marker ⓑ. Then, the turnaround is performed, from which the trailer is approached in reverse at marker ⓒ.

The effect of pre-filtering can be seen in the velocity's time graph. Initially, MPC sets the target velocity to $v_0^{\mathrm{mpc}} = 0.12\,\mathrm{m\,s^{-1}}$. When the drive is unlocked, the desired velocity smoothly increases to the commanded value due to the filtering. This behavior is consistent across all scenarios and also applies to the steering. Pre-filtering is utilized to prevent potential high-frequency demands from harming the actuators and destabilizing control.

Furthermore, the time graph of the steering shows that it is at maximum while turning to the right, which is why the initial deviation cannot be compensated. After the turnaround, smaller curvatures are driven and the truck finally comes to a standstill with a lateral deviation of 1.5 cm. In relation to the truck width, this is an error of 4.78 %. Related to a real truck width of about 2.55 m,[1] this would result in an error of about 12 cm. The frequently used towing hitches with funnels[2] have an opening of about 36 cm. This would allow the vehicle to dock with the trailer without a corrective move and without human assistance. In the case of fifth-wheel couplings, this is in fact even better because the horseshoe-shaped openings are significantly larger.[3]

[1] See "Council Directive 96/53/EC" of 25 July 1996, ANNEX I for maximum vehicle dimensions in the European Union.

[2] See e. g., "ROCKINGER RO*500".

[3] See e. g., "Jost 37 CZW" with about 50 cm opening.

Table 3
Parameters used in the experimental scenarios.

#	Scenario	T_{hor}	$\lvert v_{0,\max}\rvert$	Integral cost weights $(q_{\text{lon}}, q_{\text{lat}}, q_\psi, q_v, q_{\beta,1}, q_{\beta,2}, q_s)$	Terminal cost weights $(p_{\text{lon}}, p_{\text{lat}}, p_\psi, p_v, p_{\beta,1}, p_{\beta,2}, p_s)$	GDP offsets $\lambda = [\lambda_{\text{lon}}, \lambda_{\text{lat}}]^\top$
I	G0T coupling fw	4 s	0.12 m s^{-1}	(10, 30, 1, 0.1, 0.9)	(0, 10, 20, 0, 1)	$[-0.136\,\text{m}, 0\,\text{m}]^\top$
	G0T coupling bw	4 s	0.12 m s^{-1}	(10, 20, 20, 0.1, 1.5)	(0, 50, 10, 0, 0)	$[-0.136\,\text{m}, 0\,\text{m}]^\top$
II	G0T implement	5 s	0.30 m s^{-1}	(10, 30, 1, 0.1, 0.9)	(0, 10, 0, 0, 1)	$[0.61\,\text{m}, -0.44\,\text{m}]^\top$
III	G1T implement	6 s	0.30 m s^{-1}	(10, 50, 10, 0.1, 0, 2.0)	(0, 20, 0, 0, 0, 0)	$[-0.54\,\text{m}, -0.38\,\text{m}]^\top$
IV	G1T yard fw	6 s	0.20 m s^{-1}	(10, 50, 2, 0.1, 1, 0.8)	(0, 0, 0, 0, 50, 0)	$[0\,\text{m}, 0\,\text{m}]^\top$
	G1T yard bw	8 s	0.15 m s^{-1}	(10, 20, 10, 0.5, 3, 0.9)	(10, 50, 50, 1, 50, 0)	$[0\,\text{m}, 0\,\text{m}]^\top$
V	G2T yard fw	6 s	0.25 m s^{-1}	(10, 50, 2, 0.1, 1, 1, 0.8)	(0, 0, 0, 0, 0, 0, 0)	$[0\,\text{m}, 0\,\text{m}]^\top$
	G2T yard bw	8 s	0.15 m s^{-1}	(10, 20, 10, 0.5, 3, 3, 0.9)	(10, 50, 50, 1, 50, 50, 0)	$[0\,\text{m}, 0\,\text{m}]^\top$
VI	G2T circle	8 s	0.15 m s^{-1}	(10, 20, 10, 0.5, 3, 3, 0.9)	(10, 50, 50, 1, 50, 50, 0)	$[0\,\text{m}, 0\,\text{m}]^\top$

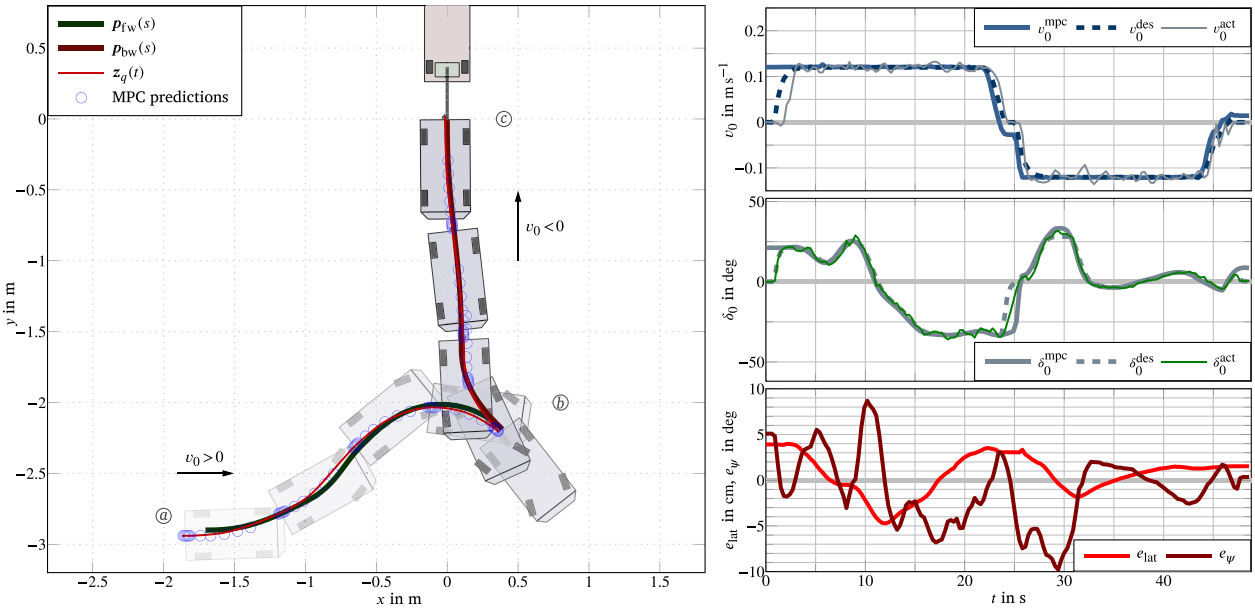


Fig. 16. Scenario I (G0T) "Coupling of a trailer": Turnaround scenario with truck only.

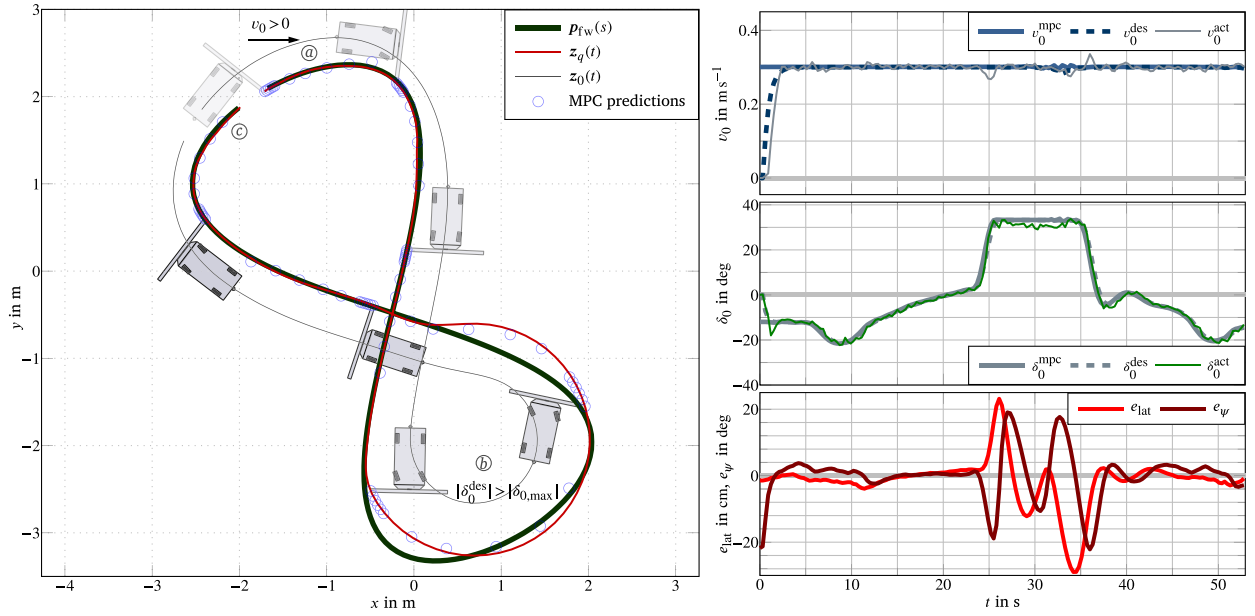


Fig. 17. Scenario II (G0T) "Truck pretending to be a tractor with mower implement": Driving along infeasible path with the right mower edge.

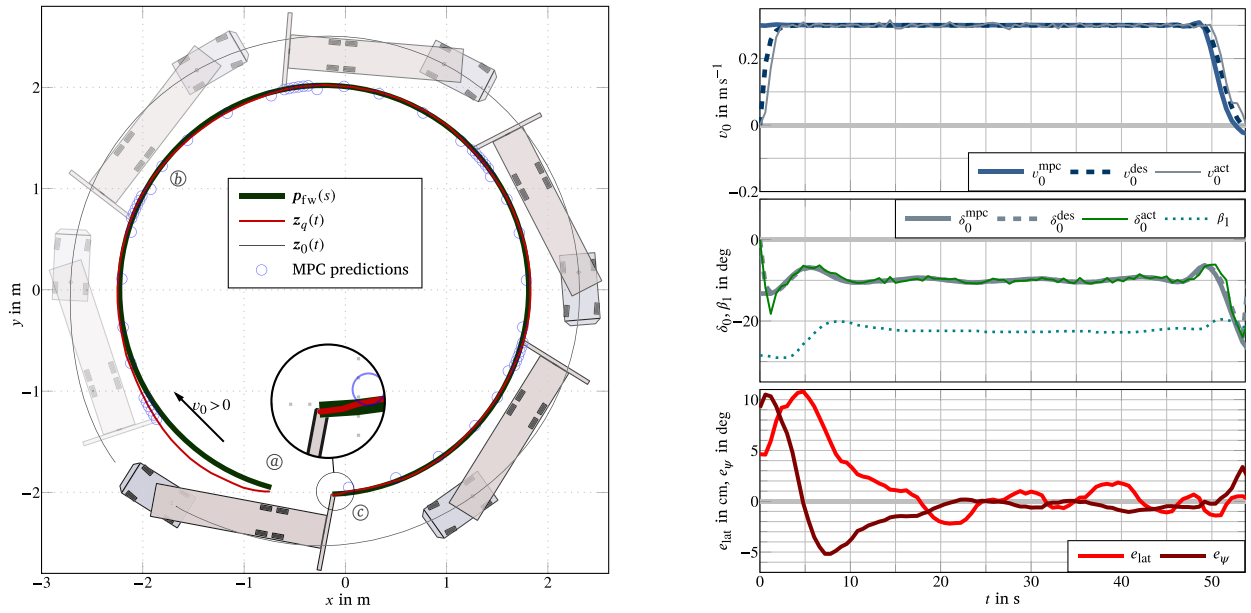


Fig. 18. Scenario III (G1T) "Truck pretending to be a tractor with a field sprayer trailer": Driving along a circle with the right edge of the sprayer.

6.1.2. Scenario II (G0T): "Vehicle only scenario with an agricultural implement on the front"

The second scenario shown in Fig. 17 is intended to demonstrate the potential of the variable guidance point. The model truck is assumed to be a tractor with a mower on the front end. Starting at marker ⓐ, the right side of the mower is moved forward along a path in the shape of a lying eight. Here the curve at marker ⓑ is so narrow that it cannot be followed with the wide sweeping mower. The finish at marker ⓒ does not end with a stop, as the scenario consists of several laps, which are not presented here for the sake of clarity.

If the section before and after the curve that is too narrow is taken into account, the mean lateral error is 1.2 cm. Related to the width of the model mower of 88 cm, this results in an error of 1.42 %. Extrapolating this to a real tractor with a 3 m wide mower unit,[4] the error would be in the range of 4 cm.

6.2. Experiments with semi-trailer (G1T)

6.2.1. Scenario III (G1T): "Field sprayer"

Like the previous experiment, this is a demonstration of the variable guidance point. Here, the vehicle consists of a tractor and trailer, with a mounted boom representing a field sprayer being towed, see Fig. 18. Due to the large dimensions, only a simple circle scenario can be executed in the laboratory. Starting at marker ⓐ with an initial deviation of 5 cm, the circle is directed towards marker ⓒ with the right edge of the boom while moving forward. After reaching marker ⓑ at about 18 s, the initial error is settled and the lateral error thereafter remains at an average of 9 mm. This corresponds to 1.16 % of the boom's width which measures 76 cm, and approximately 17 cm in reality considering a boom of 15 m width.[5] The simulation study [18] demonstrates that even intricate scenarios can be executed with high precision.

6.2.2. Scenario IV (G1T): "Parking scenario on yard"

This experiment illustrates a common situation in a depot, as can be seen in Fig. 19. A truck with a semi-trailer enters a narrow yard and must approach dock 1 with the trailer. The objective for successful loading/unloading is to come to a stop in the center of the bay and at a right angle to the loading dock.

The experiment is started at marker ⓐ with an initial lateral deviation of 7 cm. Next, the truck moves forward with a velocity of 0.20 m s^{-1} to marker ⓑ where the turn is performed. It is crucial to approach the turning point accurately and to stretch the vehicle so that no space is lost for maneuvering backwards to marker ⓒ. At the turning point, the trailer is positioned laterally deviated by 0.9 cm from the path and already slightly bent ($\beta_1 < 0$) towards the subsequent reverse curve.

After completing the turn, the combination performs a reverse motion at a velocity of −0.12 m s^{-1} around the corner and heads up to the loading ramp. The final deviation measures at 0.8 cm sideways and 0.6 cm lengthwise, with only a 0.5 deg angular error. The relative lateral error in comparison to the track width is 2.6 %. With a full-scale trailer having a track width of 2.55 m, this would translate to an error of about 6.6 cm.

6.3. Experiments with full trailer (G2T)

6.3.1. Scenario V (G2T): "Parking scenario on yard"

The experiment in this section is akin to that in Section 6.2.2, but here with the full trailer, which is equivalent to two rigid drawbar trailers. Additionally, dock 2 is the point of approach in this scenario, as can be seen in Fig. 20.

Starting at marker ⓐ, the model truck moves forward towards marker ⓑ at a velocity of 0.25 m s^{-1} with an initial lateral deviation of 1 cm. At marker ⓑ, the trailer stops closely to the turnaround point, and the lateral deviation remains at 2 cm. The vehicle is fully stretched, providing the best conditions for reversing around the curve towards the loading ramp at marker ⓒ, with a velocity of −0.15 m s^{-1}. At the dock, the trailer finally stops with a lateral deviation of 1.2 cm at a 0.04 deg angle error. That results in a relative lateral error of 3.6 %, which would be about 9 cm for a full-scale trailer with 2.55 m width.

6.3.2. Scenario VI (G2T): "Reversing on a circle"

To showcase the robustness of PF-MPC, the full trailer is driven a long distance in reverse, see Fig. 21, starting from marker ⓐ on a 4 m diameter circle to marker ⓒ with an initial deviation of 4 cm to the inside. The total distance traveled is 12 m and the target velocity is −0.15 m s^{-1}.

[4] See e. g., "Kuhn GMD 3121 F-FF".
[5] See e. g., "Amazone UG 3000 Super".

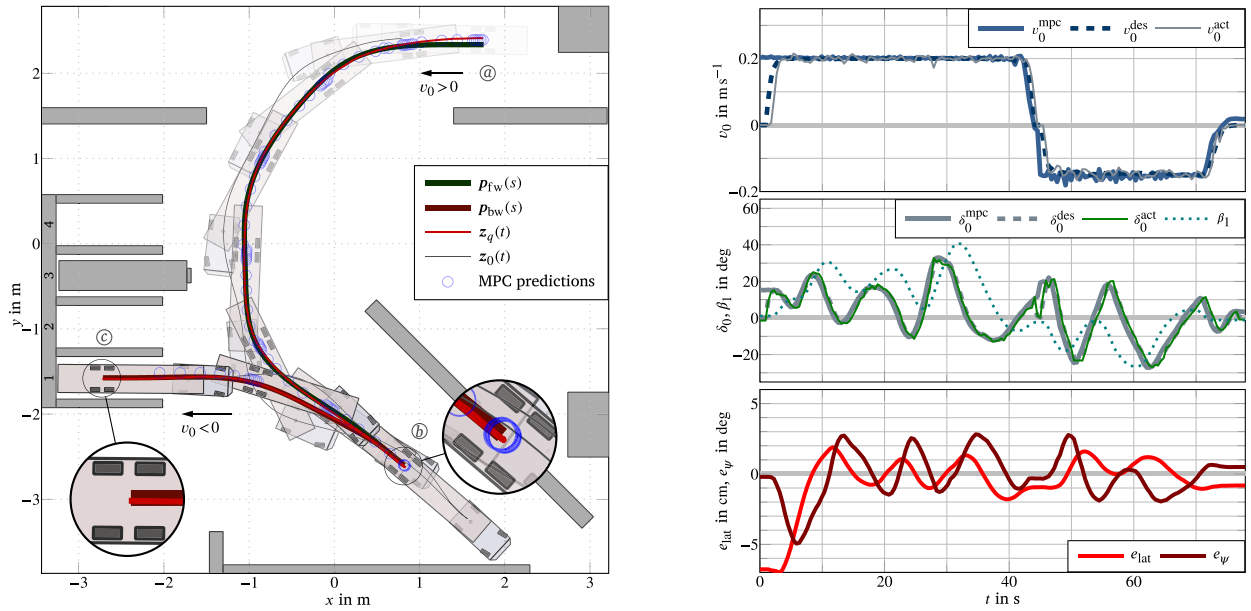


Fig. 19. Scenario IV (G1T) "Parking in the yard": Turnaround and precise parking with truck and semi-trailer. A video clip of this scenario is provided at https://www.fau.tv/clip/id/51212.

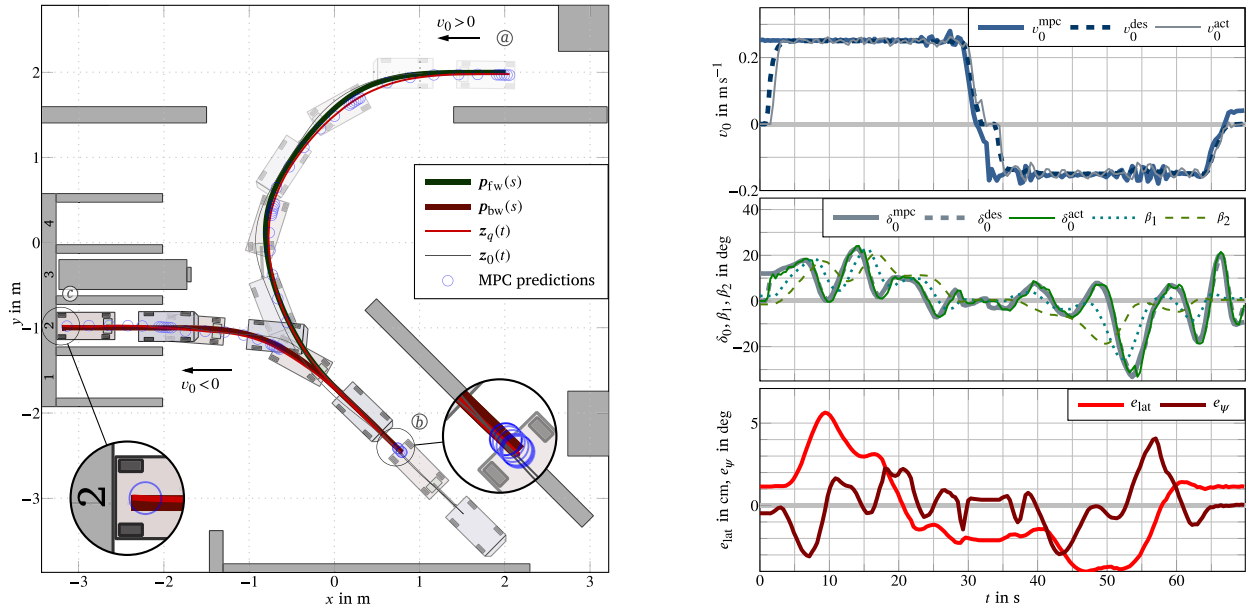


Fig. 20. Scenario V (G2T) "Parking in the yard": Turnaround and precise parking with truck and full trailer. A video clip of this scenario is provided at https://www.fau.tv/clip/id/51213.

It is observed that the deviation decreases linearly with time respectively with distance and is considered to be eliminated after about 40 s at marker ⓑ. Subsequently, the average deviation is only 0.6 cm, while the deviation at the end of the path is reduced to 0.03 cm.

This experiment demonstrates that it is technically possible to precisely reverse a two-axle trailer over a long distance, including curves, without running the risk of jackknifing.

7. Discussion and conclusion

This paper addresses challenging driving maneuvers such as parking truck–trailer combinations and precisely guiding tractors with agricultural implements. The research focuses on the development and implementation of a model predictive path-following control formulation (referred to as PF-MPC) using the kinematic single-track model. The concept offers the option to choose an arbitrary reference point, known as the guidance point, making it well-suited for agricultural machinery with implements and adaptable for various applications.

The main contribution of this publication is the successful adaptation and implementation of the PF-MPC on an embedded microcontroller in a model-scaled experimental vehicle. To ensure precise and prompt stops, terminal regions and a shrinking horizon are introduced, as well as a path parameter penalization that is switched close to the target position. Furthermore, insight into the essential requirements for the operation of experimental vehicles, including filtering and state estimation and low-level control, is provided.

The potential of model predictive control is emphasized by the experimental results. Even when handling simulated agricultural equipment, remarkable precision is achieved during forward driving, with

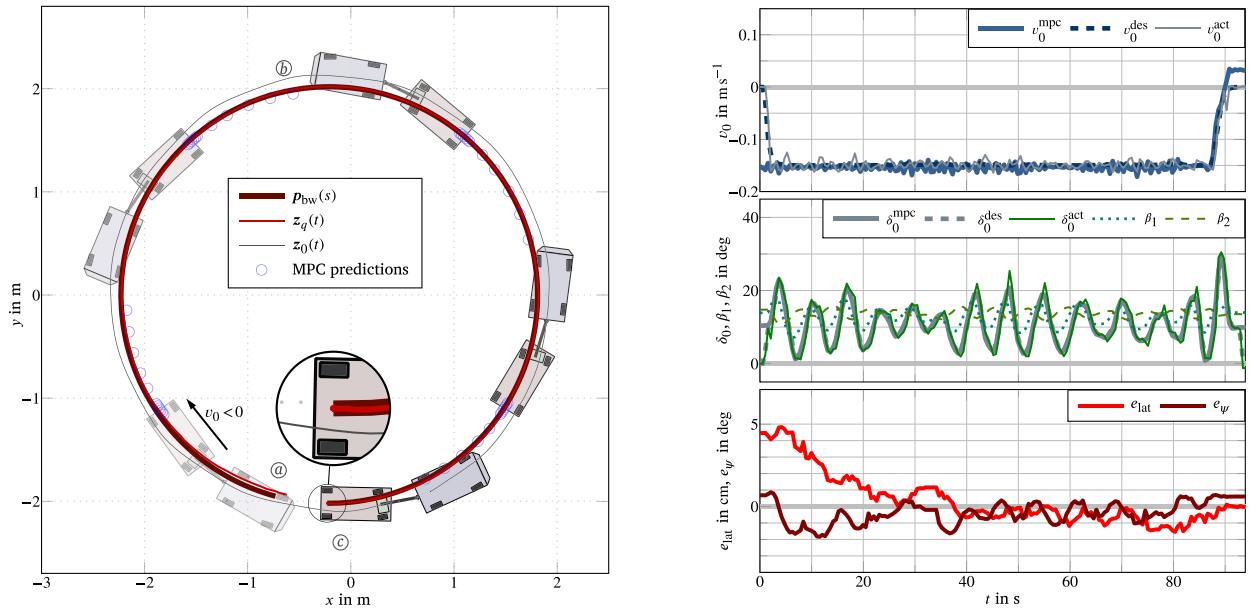


Fig. 21. Scenario VI (G2T) "Reversing on a circle": Traveling a long distance in reverse with truck and full trailer. A video clip of this scenario is provided at https://www.fau.tv/clip/id/51214.

lateral deviations of about 1.5 % of the equipment's width. When reversing and parking with trailers, the measured accuracy is 2.6 % of the trailer track width for semi-trailers and 3.6 % for full trailers.

The results of this research are very promising. They can streamline extended, precision-demanding drives in agricultural and construction machinery, enhancing efficiency and reducing operator fatigue. Furthermore, they allow for the automation of complex maneuvers, such as reversing to a loading dock or swap body, and precise parking in tight spaces with up to two kinematic trailers.

The method's robustness is demonstrated by utilizing a simplified prediction model for the truck–trailer combination. This model does not include properties like the truck's rigid axle, the semi-trailer's tandem axle, or phenomena such as tire slipping or backlash and hysteresis in steering and hitching mechanics.

However, it is essential to note that the control system relies on high-precision absolute tracking using motion capture technology. Such high-precision sensors are typically not installed in real-world applications. This presents opportunities for further research, particularly concerning the feasibility of reversing with multiple trailers using only on-board sensors such as LiDAR, cameras, IMUs, and encoders. Scaling up to full-size vehicle–trailer combinations is a crucial step in verifying the controller's applicability for real-world use.

CRediT authorship contribution statement

Markus Lukassek: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Julian Dahlmann:** Writing – review & editing, Validation, Software, Methodology. **Andreas Völz:** Writing – review & editing, Supervision. **Knut Graichen:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

The team of authors is grateful to ZF Friedrichshafen AG for generously donating the truck chassis and drive train that served as the platform of our experimental vehicle.

A special gratitude is extended to Prof. Dr.-Ing. Wilhelm Ruckdeschel of Ravensburg Cooperative State University in Germany for his committed participation in the experimental aspect of this project. Professor Ruckdeschel played a key role in configuring the truck and manufacturing the semi-trailer.

References

[1] Belman D, White C. Trucking in the age of information. Routledge; 2018.
[2] Michałek MM. Agile maneuvering with intelligent articulated vehicles: a control perspective. In: Proc. of IFAC symposium on intelligent autonomous vehicles, vol. 52, no. 8. 2019, p. 458–73.
[3] Renius KT. Fundamentals of tractor design. Springer International Publishing; 2020, p. 217–60.
[4] Bashiri B, Mann DD. Drivers' mental workload in agricultural semi-autonomous vehicles. In: Proceedings of the human factors and ergonomics society annual meeting, vol. 57, no. 1. Los Angeles, CA: SAGE Publications Sage CA; 2013, p. 1795–9.
[5] Michałek MM, Pazderski D. Forward tracking of complex trajectories with non-Standard N-Trailers of non-minimum-phase kinematics avoiding a jackknife effect. Internat J Control 2019;92(11):2547–60.
[6] Szakács T, et al. Modelling and simulation of tow angle between agricultural tractors and trailers.. Landtechnik 2010;65(3):178–81.
[7] Altafini C. Some properties of the general n-trailer. Internat J Control 2001;74(4):409–24.
[8] Michałek MM. A highly scalable path-following controller for N-trailers with off-axle hitching. Control Eng Pract 2014;29:61–73.
[9] Sklyarenko Y, Schreiber F, Schumacher W. Maneuvering assistant for truck and trailer combinations with arbitrary trailer hitching. In: Proc. of IEEE international conference on mechatronics. 2013, p. 774–9.
[10] Rouchon P, Fliess M, Lévine J, Martin P. Flatness and motion planning: the car with n trailers. In: Proc. of European control conference. 1993, p. 1518–22.
[11] Sampei M, Tamura T, Kobayashi T, Shibui N. Arbitrary path tracking control of articulated vehicles using nonlinear control theory. IEEE Trans Control Syst Technol 1995;3(1):125–31.
[12] Tilbury D, Laumond J-P, Murray R, Sastry SS, Walsh G. Steering car-like systems with trailers using sinusoids. In: Proc. of IEEE international conference on robotics and automation. IEEE; 1992, p. 1993–8.
[13] Bolzern P, DeSantis RM, Locatelli A, Masciocchi D. Path-tracking for articulated vehicles with off-axle hitching. IEEE Trans Control Syst Technol 1998;6(4):515–23.

[14] Astolfi A, Bolzern P, Locatelli A. Path-tracking of a tractor-trailer vehicle along rectilinear and circular paths: a Lyapunov-based approach. IEEE Trans Robot Autom 2004;20(1):154–60.
[15] Evestedt N, Ljungqvist O, Axehill D. Path tracking and stabilization for a reversing general 2-trailer configuration using a cascaded control approach. In: Proc. of IEEE intelligent vehicles symposium. 2016, p. 1156–61.
[16] Wu T, Hung JY. Path following for a tractor-trailer system using model predictive control. In: Proc. of IEEE SoutheastCon. 2017, p. 1–5.
[17] Feng J, Sun Z. Path-tracking control and following control of tractor-semitrailer combination based on improved MPC. SAE Tech Pap Ser 2023.
[18] Lukassek M, Völz A, Szabo T, Graichen K. Model predictive path-following control for general n-trailer systems with an arbitrary guidance point. In: Proc. of European control conference. 2021, p. 1335–40.
[19] Kayacan E, Peschel JM, Kayacan E. Centralized, decentralized and distributed nonlinear model predictive control of a tractor-trailer system: A comparative study. In: Proc. of American control conference. 2016, p. 4403–8.
[20] Backman J, Oksanen T, Visala A. Nonlinear model predictive trajectory control in tractor-trailer system for parallel guidance in agricultural field operations. In: Proc. of IFAC conference AGRICONTROL. 2010, p. 133–8.
[21] Bos M, Vandewal B, Decré W, Swevers J. MPC-based motion planning for autonomous truck-trailer maneuvering. IFAC-PapersOnLine 2023;56(2):4877–82, 22nd IFAC World Congress.
[22] Kumar M, Haas A, Strauss P, Kraus S, Taş ÖŞ, Stiller C. Conception and experimental validation of a model predictive control (MPC) for lateral control of a truck-trailer. In: 2022 IEEE intelligent vehicles symposium. 2022, p. 1550–7.
[23] Chen J, Jiang S, Zhou Z, Zhang M, Ming X, Guo N. Lateral semi-trailer truck control using a parameter self-learning MPC method in urban environment. Proc Inst Mech Eng D 2023.
[24] Beglini M, Belvedere T, Lanari L, Oriolo G. An intrinsically stable MPC approach for anti-jackknifing control of tractor-trailer vehicles. IEEE/ASME Trans Mechatronics 2022;27(6):4417–28.
[25] Ljungqvist O, Axehill D, Pettersson H. On sensing-aware model predictive path-following control for a reversing general 2-trailer with a car-like tractor. In: Proc. of IEEE international conference on robotics and automation. 2020, p. 8813–9.
[26] Evestedt N, Ljungqvist O, Axehill D. Motion planning for a reversing general 2-trailer configuration using closed-loop RRT. In: Proc. of IEEE/RSJ international conference on intelligent robots and systems. 2016, p. 3690–7.
[27] Palmieri L, Koenig S, Arras KO. RRT-based nonholonomic motion planning using any-angle path biasing. In: Proc. of IEEE international conference on robotics and automation. 2016, p. 2775–81.
[28] Rimmer AJ, Cebon D. Planning collision-free trajectories for reversing multiply-articulated vehicles. IEEE Trans Intell Transp Syst 2016;17(7):1998–2007.
[29] Ljungqvist O, Evestedt N, Cirillo M, Axehill D, Holmer O. Lattice-based motion planning for a general 2-trailer system. In: Proc. of IEEE intelligent vehicles symposium. 2017, p. 819–24.
[30] Ghilardelli F, Lini G, Piazzi A. Path generation using η^4-splines for a truck and trailer vehicle. IEEE Trans Autom Sci Eng 2014;11(1):187–203.
[31] Dahlmann J, Völz A, Szabo T, Graichen K. A numerical approach for solving the inversion problem for n-trailer systems. In: Proc. of American control conference. 2022, p. 2018–24.
[32] Michałek MM, Pazderski D. Computing the admissible reference state-trajectories for differentially non-flat kinematics of non-standard N-trailers. In: Proc. of European control conference. 2018, p. 551–6.
[33] Zips P, Böck M, Kugi A. An optimisation-based path planner for truck-trailer systems with driving direction changes. In: Proc. of IEEE international conference on robotics and automation. 2015, p. 630–6.
[34] Cen H, Li B, Acarman T, Zhang Y, Ouyang Y, Dong Y. Optimization-based maneuver planning for a tractor-trailer vehicle in complex environments using safe travel corridors. In: Proc. of IEEE intelligent vehicles symposium. 2021, p. 974–9.
[35] Li B, Zhang Y, Acarman T, Kong Q, Zhang Y. Trajectory planning for a tractor with multiple trailers in extremely narrow environments: A unified approach. In: Proc. of international conference on robotics and automation. IEEE; 2019, p. 8557–62.
[36] Dahlmann J, Völz A, Lukassek M, Graichen K. Local predictive optimization of globally planned motions for truck-trailer systems. IEEE Transactions on Control Systems Technology (TCST) [in press], doi: 10.1109/TCST.2023.3345169 (Early access).
[37] Lukassek M, Völz A, Szabo T, Graichen K. Model predictive control for agricultural machines with implements. In: Proc. of mediterranean conference on control and automation. 2020, p. 387–92.
[38] Laumond J-P, Sekhavat S, Lamiraux F. Robot motion planning and control. Springer-Verlag Berlin; 1998, p. 1–44 (Chapter Guidelines in Nonholonomic Motion Planning for Mobile Robots).
[39] Dahlmann J, Völz A, Szabo T, Graichen K. Trajectory optimization for truck-trailer systems based on predictive path-following control. In: Proc. of conference on control technology and applications. 2022.
[40] Faulwasser T, Matschek J, Zometa P, Findeisen R. Predictive path-following control: Concept and implementation for an industrial robot. In: Proc. of the IEEE int. conf. on control applications. 2013.
[41] Rawlings JB, Mayne DQ, Diehl M, et al. Model predictive control: theory, computation, and design, vol. 2, WI: Nob Hill Publishing Madison; 2017.
[42] Völz A. Collision-free motion planning for dual-arm robots in changing environments (Ph.D. thesis), Shaker Verlag, Ulm University; 2019, p. 109–13.
[43] Englert T, Völz A, Mesmer F, Rhein S, Graichen K. A software framework for embedded nonlinear model predictive control using a gradient-based augmented Lagrangian approach (GRAMPC). Opt Eng 2019;20(3):769–809.
[44] Dahlmann J, Völz A, Graichen K. A concept for automated maneuvering of vehicles with trailers. at - Automatisierungstechnik (AT) 2024;72(4):354–67.
[45] Fortin M, Glowinski R. Augmented lagrangian methods: Application to the solution of boundary-value problems. Elsevier Science; 1983.
[46] Graichen K, Kugi A. Stability and incremental improvement of suboptimal MPC without terminal constraints. IEEE Trans Automat Control 2010;55(11):2576–80.
[47] Werling M. Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien (Ph.D. thesis), Karlsruher Institut für Technologie (KIT); 2010.
[48] Ljungqvist O, Evestedt N, Axehill D, Cirillo M, Pettersson H. A path planning and path-following control framework for a general 2-trailer with a car-like tractor. J Field Robotics 2019;36(8):1345–77.
[49] Altafini C, Speranzon A, Wahlberg B. A feedback control scheme for reversing a truck and trailer vehicle. IEEE Trans Robot Autom 2001;17(6):915–22.
[50] Mai P, Hillermeier C. Least squares based derivative estimation: Theory and tuning rules for the practical application. at - Automatisierungstechnik 2008;56(10):530–8.
[51] Savitzky A, Golay MJ. Smoothing and differentiation of data by simplified least squares procedures. Anal Chem 1964;36:1627–39.
[52] Bar-Shalom Y. Estimation with applications to tracking and navigation: Theory algorithms and software. Hoboken: Wiley; 2001.
[53] Adamy J. Nichtlineare Systeme und Regelungen. Springer; 2018.

Markus Lukassek received the M.Sc. degree in electrical engineering with the focus on automation technology from Ulm University, Ulm, Germany, in 2018. Until the end of 2021, he has been a research assistant with the Chair of Automatic Control, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany. His research interests included model predictive control with a focus on maneuvering truck-trailer systems. He is currently with ZF Friedrichshafen AG, Friedrichshafen, Germany, where he develops automated operations for industrial vehicles as a software engineer.

Julian Dahlmann received the M.Sc. degree in electrical engineering from Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany, in 2019. He is currently working towards the Ph.D. (Dr.-Ing.) degree with the Chair of Automatic Control, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany. His research interests include autonomous maneuvering of mobile robots with a focus on motion planning and control for general n-trailer systems.

Andreas Völz received the M.Sc. degree in communications and computer engineering and the Ph.D. (Dr.-Ing.) degree from Ulm University, Ulm, Germany, in 2014 and 2019, respectively. Since 2019, he has been with the Chair of Automatic Control, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany. His current research interests include optimization-based motion planning and model predictive control with a focus on robotics applications.

Knut Graichen (Senior Member, IEEE) received the Diploma-Ing. degree in Engineering Cybernetics and the Ph.D. (Dr.-Ing.) degree from the University of Stuttgart, Stuttgart, Germany, in 2002 and 2006. In 2007, he was a Post-Doctoral Researcher with the Centre Automatique et Systémes, MINES ParisTech, France. In 2008, he joined the Automation and Control Institute, Vienna University of Technology, Vienna, Austria, as a Senior Researcher. In 2010, he became a Professor with the Institute of Measurement, Control and Microtechnology, Ulm University, Ulm, Germany. Since 2019, he has been the Head of the Chair of Automatic Control, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany. His current research interests include distributed and learning control as well as model predictive control of dynamical systems with applications in mechatronic, robotic, and networked systems. He is the Deputy Editor-in-Chief of Control Engineering Practice.