An Optimisation-Based Path Planner for Truck-Trailer Systems with Driving Direction Changes

authored by **P. Zips, M. Böck, and A. Kugi**

and published in *Proc. IEEE Int. Conf. on Robotics and Automation*.

Cite this article as:

P. Zips, M. Böck, and A. Kugi, "An optimisation-based path planner for truck-trailer systems with driving direction changes", in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 630–636. DOI: 10.1109/ICRA.2015.7139245

BibTex entry:

```
@InProceedings{acinpaper,
  Title = {An Optimisation-Based Path Planner for Truck-Trailer Systems with Driving Direction Changes},
  Author = {Zips, P. and B{\"o}ck, M. and Kugi, A.},
  Booktitle = {Proc. IEEE Int. Conf. on Robotics and Automation},
  Year = {2015},

  Address = {Seattle, WA, USA},
  Month = may,
  Pages = {630-636},

  Doi = {10.1109/ICRA.2015.7139245}
}
```

Link to original paper:

http://dx.doi.org/10.1109/ICRA.2015.7139245

Contact:

Automation and Control Institute (ACIN)
Vienna University of Technology
Gusshausstrasse 27-29/E376
1040 Vienna, Austria

Internet: www.acin.tuwien.ac.at
E-mail: office@acin.tuwien.ac.at
Phone: +43 1 58801 37601
Fax: +43 1 58801 37699

An Optimisation-Based Path Planner for Truck-Trailer Systems with Driving Direction Changes

Patrik Zips, Martin Böck, Andreas Kugi

***Abstract*— This paper presents a path planning concept for trucks with trailers with kingpin hitching. This system is non-holonomic, has no flat output and is not stable in backwards driving direction. These properties are major challenges for path planning. The presented approach concentrates on the loading bay scenario. The considered task is to plan a path for the truck-trailer system from a start to a specified target configuration corresponding to the loading bay. Thereby, close distances to obstacles and multiple driving direction changes have to be handled. Furthermore, a so-called jackknife position has to be avoided. In a first step, an initial path is planned from the target to the start configuration using a tree-based path planner. Afterwards this path is refined locally by solving an optimal control problem. Due to the local nature of the planner, heuristic rules for direction changes are formulated. The performance of the proposed path planner is evaluated in simulation studies.**

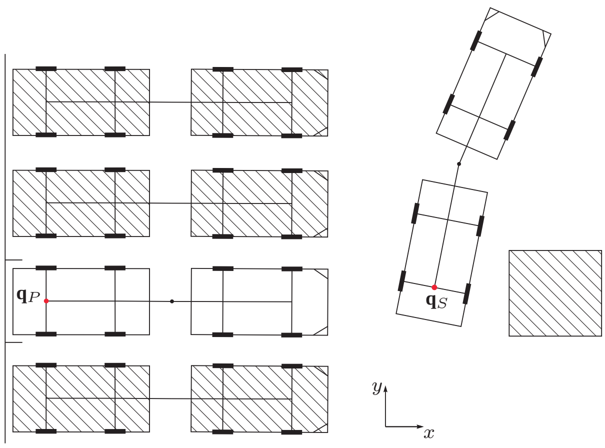


Fig. 1: Loading bay with obstacles.

I. Introduction

Large trucks constitute an essential part in the local and global transportation of goods. As the traffic is increasing, a decrease of the overall number of vehicles is pursued to disburden the roads and to reduce exhaust gas emissions. In order to decrease the number of vehicles, trailers are usually attached to the trucks. In many countries, road trains are already a regular component of the traffic. Such long trucks with trailers can be easily manoeuvred on motorways but getting to a loading bay or parking spot is challenging even for experienced drivers.

In the last decades, several approaches were developed to plan a path for a system with (multiple) trailers which are attached to the axle of the towing vehicle. For simplicity these trailers are typically supposed to have only one axle. A trailer with more than one axles can be modelled by combining several trailers with one axle [1]. The flatness property was proven in [2].

If the first trailer is not attached directly to the rear axle but to a kingpin instead the system is referred to as tractor-trailer system. This off-axle hitching introduces several properties, e.g., loss of flatness and instability in backward driving direction [3]. Therefore, planning a path for this type of systems turns out to be a more complex task.

In this paper we propose a path planning concept for a truck-trailer system which is a special subclass of the tractor-trailer system with one trailer and a towing vehicle each with two axles. The planner calculates a path from a starting to a parking configuration for a loading bay scenario by combining a global with a local approach. In Section II, we give a description of the problem under consideration as well as an overview of the state of the art. Section III explains the proposed path planner and in Section IV the feasibility of the planner is shown by means of several simulation studies. Section V contains some conclusions and an outlook on further works.

The authors are with the Automation and Control Institute, Vienna University of Technology, 1040 Vienna, Austria (e-mail: {zips,boeck,kugi}@acin.tuwien.ac.at).

II. Problem Statement

This paper is concerned with the development of a method to manoeuvre a truck-trailer system with kingpin hitching from a starting configuration $\mathbf{q}_S$ to a certain parking configuration $\mathbf{q}_P$, see Fig. 1. The shaded polygons depict obstacles like other trucks with trailers or a placed container. We only consider polygonal obstacles as every other shape can be included into a slightly larger polygon. Collision checks are carried out using the Minkowski-sum [4] where the whole configuration of the truck-trailer system including its dimensions is checked for collisions.

In the following, we concentrate on the loading bay scenario and aim for a path planner which is fast enough to be used in real-time applications. The planned path shall connect the starting and the parking configuration respecting the system dynamics of the truck-trailer system as well as avoid obstacles and contain direction switching points if necessary. Additionally a so-called jackknife position, where the truck and the trailer are nearly anti-parallel, shall be avoided.

Post-print version of the article: P. Zips, M. Böck, and A. Kugi, "An optimisation-based path planner for truck-trailer systems with driving direction changes", in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 630–636. doi: 10.1109/ICRA.2015.7139245
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

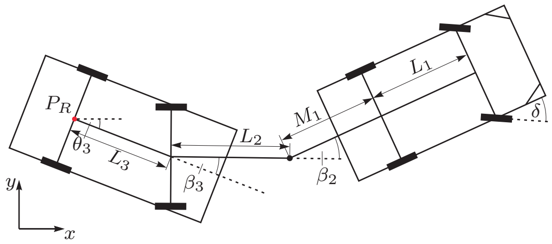


Fig. 2: Model of the truck-trailer system.

A. System Dynamics

The truck-trailer system with kingpin hitching is depicted in Fig. 2. Using the abbreviations $s_\sigma = \sin(\sigma)$, $c_\sigma = \cos(\sigma)$, and $t_\sigma = \tan(\sigma)$, the system can be described by the differential equations [3]

$$x_3' = Dc_{\beta_3}c_{\beta_2}\left(1 + \frac{M_1}{L_1}t_{\beta_2}t_\delta\right)c_{\theta_3} \tag{1a}$$

$$y_3' = Dc_{\beta_3}c_{\beta_2}\left(1 + \frac{M_1}{L_1}t_{\beta_2}t_\delta\right)s_{\theta_3} \tag{1b}$$

$$\theta_3' = D\frac{s_{\beta_3}c_{\beta_2}}{L_3}\left(1 + \frac{M_1}{L_1}t_{\beta_2}t_\delta\right) \tag{1c}$$

$$\beta_3' = Dc_{\beta_2}\left(\frac{1}{L_2}\left(t_{\beta_2} - \frac{M_1}{L_1}t_\delta\right) - \frac{s_{\beta_3}}{L_3}\left(1 + \frac{M_1}{L_1}t_{\beta_2}t_\delta\right)\right) \tag{1d}$$

$$\beta_2' = D\left(\frac{t_\delta}{L_1} - \frac{s_{\beta_2}}{L_2} + \frac{M_1}{L_1L_2}c_{\beta_2}t_\delta\right), \tag{1e}$$

where $(\cdot)'$ denotes the derivative with respect to the path parameter s. The variables x_3 and y_3 describe the position of the reference point P_R on the rear axle of the trailer and θ_3 the orientation of the trailer with respect to the x-axis. The angle β_3 denotes the difference between the trailer orientation and the tow-bar and β_2 the angle between the tow-bar and the truck. The absolute values of both angles are constrained by a maximum value $\beta_{3_{max}}$ and $\beta_{2_{max}}$, respectively. The steering angle δ of the truck serves as input u for the system. Its absolute value is constrained by a maximum steering angle δ_{max}. The driving direction is determined by $D \in \{-1, 1\}$ and is $D = -1$ for backward and $D = 1$ for forward driving. The parameters L_1, L_2, L_3, and M_1 denote the physical dimensions of the truck-trailer as depicted in Fig. 2. In the following, (1) is written in vector notation

$$\mathbf{q}' = \mathbf{f}(\mathbf{q}, u) \tag{2}$$

with the state vector $\mathbf{q} = [x_3, y_3, \theta_3, \beta_3, \beta_2]^{\mathrm{T}}$.

B. State of the Art

Motion planning for vehicles with trailers received a lot of attention in the last decades. The structure of the differential equations of such systems does not allow to find an explicit solution in a straightforward manner or to calculate an approximate numeric solution with acceptable computational costs. Therefore, an overview of some solutions for n-trailer systems, tractor-trailer systems as well as truck-trailer systems will be given in the following.

A feasible path for all three classes of systems can be found by connecting the starting with the parking position with lines and clothoids [5], [6]. This method uses specific combinations of lines and clothoids which are known from experience. Therefore, only certain scenarios can be handled which have to be learned in advance. Also any further obstacles cannot be considered in an easy way.

In order to account for obstacles, an initial path obtained, e.g., by the method mentioned above, can be deformed until no more collision occurs. Lamiraux proposed a deformation method for non-holonomic systems which is based on minimising a potential field [7]. In [8] the path deformation is used to correct localisation errors at the end-position of the path at a loading bay for a truck-trailer system. The drawbacks of this method are that no additional direction switching points can be inserted and that the numerical effort is quite high.

Another possibility to utilise potential fields is to follow the gradient of such a field with its minimum at the parking position and high values at obstacles. For this, some modifications respecting the non-holonomic nature of the system are necessary [9]. This method is not well suited for systems with trailers as the potential field usually does not consider the orientation of the vehicle. This means that the planner moves the truck and trailer always away from the obstacles even if a rotation would be more efficient. If little space is available like in a loading bay scenario the method is likely to fail. Moreover, the issue of introducing appropriate direction switching points if necessary remains unsolved.

Another method of classical robot motion planning applied to these systems is to use graph based planners like the A* or D* path planner [10], [11]. Thereby, the five dimensional configuration space has to be sampled which leads to a very large amount of data for realistic scenarios. By allocating the cells dynamically, the memory usage can be reduced but still these algorithms are not feasible for real-time applications.

A widely spread method to find a path for systems with trailers is to steer the system using a controller stabilising the parking position [12], [13]. By using a controller for the velocity of the truck, the driving direction will be switched in order to reach the desired target position but obstacles cannot be considered at all.

In [14] the path planning problem is formulated as a root finding problem and solved using an ODE solver. State constraints as well as simple obstacles (e.g., a line) can be considered by adding a potential field to the root finding problem.

III. Motion Planning

For the truck-trailer system with kingpin hitching, finding a path with driving direction switching points is not possible in a global fashion within reasonable computing time. Therefore, a different approach is proposed in this paper which combines a global and a local planning scheme.

Post-print version of the article: P. Zips, M. Böck, and A. Kugi, "An optimisation-based path planner for truck-trailer systems with driving direction changes", in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 630–636. DOI: 10.1109/ICRA.2015.7139245

Compared to the methods presented in the previous section, this path planner is able to handle obstacles as well as driving direction switching points with low computational costs by focusing on the loading bay scenario. This scenario is characterised by rather short driving paths in a narrow environment where often multiple driving direction switching points are necessary.

The proposed path planner combines a global with a local approach. Due to the local nature driving direction switching points can be calculated in an efficient way. The path planning does not depend on a certain starting or parking configuration and exhibits low computational costs and memory requirements. The following subsection will present the general idea of the path planner. Thereby, a path is regarded as a sequence of states $\mathbf{q}$ parametrised in the path parameter s.

A. Principle of the Path Planner

The overall path planning algorithm is divided into two steps. Firstly, an initial path is calculated beginning from the parking configuration $\mathbf{q}_P$. This initial path respects the differential equations (2) and the limit of the steering angle δ_{max}. Moreover, it is planned in such a way that it is free of collisions with obstacles. However, the initial path does not have to lead exactly to the starting configuration $\mathbf{q}_S$ but to a neighbourhood of the starting position. In addition, the state constraints on the angles β_3 and β_2 do not have to be fulfilled. Under these assumptions, a variety of motion planners are available for generating the initial path in a time efficient way. Henceforth, a tree-based planner, which is fast and has good convergence properties, will be employed, see Section III-B.

In the second step, the initial path serves as a reference. Beginning from the exact starting configuration $\mathbf{q}_S$, the aim is to find a path which, in the sense of a prescribed objective functional, is locally as close as possible to the initial path under consideration of the system equations (2), obstacles, as well as the state and input constraints. Finally, the path shall reach the parking configuration $\mathbf{q}_P$ up to a certain numerical accuracy. In general, close to the parking configuration $\mathbf{q}_P$ the initial path is feasible with respect to the state constraints. The further away from the parking configuration the more variation of the initial path is needed to account for the state constraints and to correct the deviation between the initial path and the actual starting configuration $\mathbf{q}_S$. This may entail the need for several driving direction changes in this region. Therefore, the overall path from $\mathbf{q}_S$ to $\mathbf{q}_P$ is composed of phases of driving in backward and forward direction. These sub-paths are determined by repeatedly solving optimal control problems which, amongst others, aim at minimising the deviation to the state trajectories of the initial path, cf. Sections III-C and III-E. The driving direction changes are determined based on heuristic rules, see Section III-D.

B. Initial Path

As already mentioned before, the first task is to find a collision free path connecting the parking configuration $\mathbf{q}_P$ with a neighbourhood of the starting position. For this, the fast motion planner from [15], which respects the system dynamics (2) and the maximum steering angle δ_{max} is used. It is worth noting that a different choice of the initial path planner can yield better convergence results. However, in the considered scenario the used planner has good convergence properties at very low computing costs.

The tree-based motion planner applies a randomised input $-\delta_{max} \leq u \leq \delta_{max}$ over a certain distance and integrates the system equations (2) to obtain a new state called node. Starting from this node, a new randomised input is chosen and the system is integrated again. If a node is in collision it will be discarded and an already existing node is randomly chosen as a new starting point for integration. Within this approach it is not possible to account for driving direction changes in a straightforward way.

To obtain a feasible path, provided that one exists without changing the driving direction, it is required that the last node is close to the starting configuration $\mathbf{q}_S$ in all five states $\mathbf{q}$ of the truck-trailer system. The likelihood to obtain such a node with a randomised planner is rather small. As the effort is exponentially increasing with the number of states the computing costs are in general quite high. Therefore, the planning of the initial path is considered successful if a node is in a rather large neighbourhood of the starting position.

The output of the tree-based path planner is an initial path $\mathbf{q}_I(s)$ with the path parameter $s_{0,I} \leq s \leq s_{E,I}$, the initial path value $s_{0,I}$, and the end value $s_{E,I}$. The initial path connects the parking configuration $\mathbf{q}_I(s_{E,I})$ with a neighbourhood of the starting position. It is collision free and respects the system dynamics (2) as well as the maximum steering angle δ_{max} but does not account for the constraints on the angles β_3 and β_2.

C. Path Following

In order to obtain a feasible path connecting the starting and the parking configuration a local path following approach is proposed. For this the following optimal control problem (OCP)

$$\min_{u(\cdot)} \; J(u(\cdot)) = \int_{s_0}^{s_1} l_F(\mathbf{q}, u)\mathrm{d}s \tag{3a}$$

$$\text{s.t. } \mathbf{q}' = \mathbf{f}(\mathbf{q}, u), \quad \mathbf{q}(s_0) = \mathbf{q}_0 \tag{3b}$$

$$|u| \leq \delta_{max}, \tag{3c}$$

$$|\beta_3| \leq \beta_{3_{max}}, \quad |\beta_2| \leq \beta_{2_{max}}, \tag{3d}$$

with the integral costs

$$l_F = \mathbf{e}_q^{\mathrm{T}} \mathbf{Q} \mathbf{e}_q + r_F c_F(\mathbf{q}) \tag{4}$$

is formulated.

Thereby, the positive definite matrix $\mathbf{Q}$ describes a weighting matrix for the deviation $\mathbf{e}_q = \mathbf{q} - \mathbf{q}_I$ between the truck-trailer state $\mathbf{q}$ and the state of the initial path $\mathbf{q}_I$. The term $c_F(\mathbf{q})$ together with its weighting term $r_F > 0$ is used for

Post-print version of the article: P. Zips, M. Böck, and A. Kugi, "An optimisation-based path planner for truck-trailer systems with driving direction changes", in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 630–636. DOI: 10.1109/ICRA.2015.7139245
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

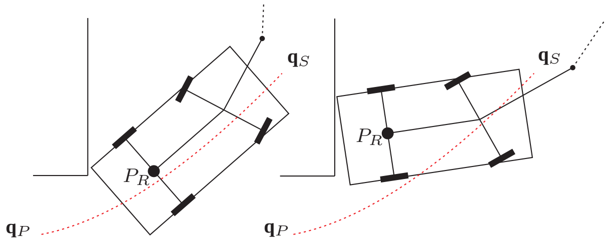


Fig. 3: Avoidable (left) and not avoidable (right) collision.

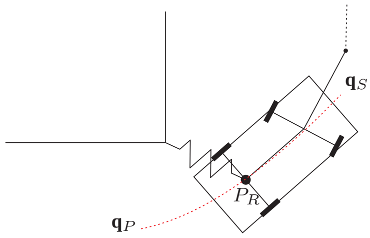


Fig. 4: Repelling spring.

obstacle avoidance as will be explained later in Section III-D. Constraint (3b) accounts for the system dynamics (2) and (3c) for the input constraint. In (3d) the angles β_3 and β_2 are constrained by a maximum value to avoid self collision in a jackknife position.

Initially, the OCP (3) is being solved beginning from the starting configuration in backward driving direction by setting $\mathbf{q}_0 \leftarrow \mathbf{q}_S$ and $s_0 \leftarrow s_{0,I}$. It is not solved globally for the whole path but only locally for a certain distance $S = s_1 - s_0$. From a computational point of view, this is more efficient than solving the OCP for the whole path at once. Additionally, this allows to introduce heuristic rules for driving direction changes as will be explained in the following subsection. After solving the OCP, the truck with the trailer is moved forward for a distance $s_S < S$ and the OCP is solved again from the new configuration $\mathbf{q}_0 \leftarrow \mathbf{q}(s_0 + s_S), s_0 \leftarrow s_0 + s_S, s_1 \leftarrow s_1 + s_S$ for the distance S. This process is iteratively repeated until the parking configuration is reached. Thus, the path planning task is performed in a receding horizon fashion.

D. Collision Avoidance and Driving Direction Changes

As long as the end configuration of the initial path is in the vicinity of the starting configuration $\mathbf{q}_S$ and no obstacles are close to the initial path, the recursive solution of the OCP (3) yields a feasible path. If there are obstacles close to the initial path, small deviations from the initial path may already lead to collisions. These collisions can in turn be avoided by deforming the path away from the obstacles. If the deviation between the starting configuration and the initial path is too large, a feasible path that avoids the collision with obstacles may not be found without changing the driving direction. Fig. 3 shows two examples, on the left hand side a scenario with an avoidable collision and on the right hand side a situation where a driving direction change is necessary to be able to pass the obstacle. The initial path of the trailer is depicted as the dashed line.

At first, the avoidable collision will be treated. In the OCP, we refrain from doing an exact collision check as this would entail too high computing costs. Moreover, including an exact collision test as additional constraint in the OCP (3) may lead to numerical difficulties for the solver. Instead, a path deformation strategy is pursued by suitably designing the cost function (4). To this end, a repelling spring is introduced at each corner of the obstacles, see Fig. 4. These springs deform the path away from the obstacles. The spring force of corner i is defined as

$$F_{F_i} = -\frac{d_i}{d_i + d_0}, \tag{5}$$

where d_i denotes the Euclidean distance between the reference point P_R on the rear axle of the trailer and the corresponding corner i of the obstacle and $d_0 > 0$ is a small distance to avoid singularities at $d_i = 0$. The function (5) has a large slope close to the obstacle corner while decreasing quickly to -1 further away from the obstacle. The spring force of all corners of all obstacles are summed up in the term

$$c_F(\mathbf{q}) = \sum_i F_{F_i} \tag{6}$$

and added to the integral costs (4) weighted by $r_F > 0$.

If the deviation from the initial path is too large the repelling spring may not push the path away from the obstacle. Looking at the example on the right hand side of Fig. 3, a repelling spring at the corner of the obstacle would rather exhibit a totally wrong behaviour by pushing the trailer towards the obstacle.

To account for such collisions, a collision check is executed for the next configuration after each solution of the OCP. If this point is in collision, the last solution of the OCP is discarded and a driving direction switching point is inserted instead. The subsequent forward driving is expected to improve the configuration of the truck-trailer system in order to increase the chance of passing the obstacle. The forward driving strategy will be explained in the next subsection.

Changes in the driving direction are not only necessary for collision avoidance but also to allow for convergence to the initial path in case the starting configuration is too far away from the initial path. In particular errors in the orientation of the trailer θ_3 and in the angles β_3 and β_2 may not be correctable while driving backward as this can lead to a jackknife position. Therefore, the behaviour of a human driver should be imitated similar to the presented car parking path planning approach in [16].

For this, the cost function

$$V = \bar{\mathbf{e}}_q^{\mathrm{T}} \mathbf{R} \bar{\mathbf{e}}_q, \quad \mathbf{R} > 0 \tag{7}$$

is introduced, containing the deviation $\bar{\mathbf{e}}_q = [x_3 - x_{3_I}, y_3 - y_{3_I}, \theta_3 - \theta_{3_I}]^{\mathrm{T}}$ between the trailer states and the trailer states of the initial path.

Post-print version of the article: P. Zips, M. Böck, and A. Kugi, "An optimisation-based path planner for truck-trailer systems with driving direction changes", in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 630–636. DOI: 10.1109/ICRA.2015.7139245
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

As a heuristic rule, the driving direction is changed if

$$V > kV_{min}, \tag{8}$$

with a suitable constant $k > 1$. $V_{min} = \min_s V(s)$ denotes the minimum cost value along the path so far. With this rule, the driving direction is changed if the truck-trailer system moves further away from the path, in the sense of (7), compared to the closest configuration reached so far. For example, this can occur if the trailer position is close to the path but the angles β_3 and β_2 do not correspond to the initial path. In the next iterations, the trailer position cannot be kept close to the initial path. Continuing driving in backward direction would further increase the angles β_3 and β_2 of the truck-trailer system most likely leading to a jackknife configuration. However, by changing the direction and driving forward the angles can be improved with respect to the initial path yielding a configuration of the truck-trailer system with higher manoeuvrability. In extensive simulation studies, jackknife configurations could be prevented using this heuristic rule.

E. Forward Driving

As mentioned before, the purpose of forward driving is to improve the position of the truck-trailer system to be able to follow the initial path more precisely and to allow for the avoidance of obstacles under consideration of the system constraints. Additionally, the forward driving has the effect of preventing a jackknife configuration. Therefore, the forward driving strategy intends to bring the truck-trailer system into a position where it can get closer to the initial path. For this purpose, again an OCP

$$\min_{u(\cdot)} \; J_F(u(\cdot)) = r_{\theta_3} e_{\theta_3}^2 + \int_{s_P}^{s_F} 1 \,\mathrm{d}s \tag{9a}$$

$$\text{s.t. } \mathbf{q}' = \mathbf{f}(\mathbf{q}, u), \quad \mathbf{q}(s_P) = \mathbf{q}(s_0) \tag{9b}$$

$$|u| \leq \delta_{max}, \tag{9c}$$

$$|\beta_3| \leq \beta_{3_{max}}, \quad |\beta_2| \leq \beta_{2_{max}}, \tag{9d}$$

is formulated, with $r_{\theta_3} > 0$ and $e_{\theta_3} = \theta_3(s_F) - \theta_{3_I}(s_P)$ denoting the difference between the trailer angle $\theta_3(s_F)$ at the final point $s = s_F$ of the optimisation horizon and the trailer angle $\theta_{3_I}(s_P)$ of the initial path $\mathbf{q}_I$ at the starting point $s = s_P$ of the optimisation horizon. The OCP is carried out starting from the current path position s_P to a free but limited path end position s_F. The starting configuration corresponds to the state $\mathbf{q}(s_0)$ at the current path position.

The aim of the OCP (9) is to align the angle of the trailer $\theta_3(s_F)$ with the angle of the initial path $\theta_{3_I}(s_P)$ by minimising the driving distance necessary for this manoeuvre at the same time. The latter is taken into account by the integral term in the cost functional (9a).

The new configuration increases the manoeuvrability of the truck-trailer system by decreasing the angles β_2 and β_3. After solving the OCP (9) a collision check is carried out. If a collision occurs at any position of the forward driving path, this position is taken as end point s_F of the forward driving distance.

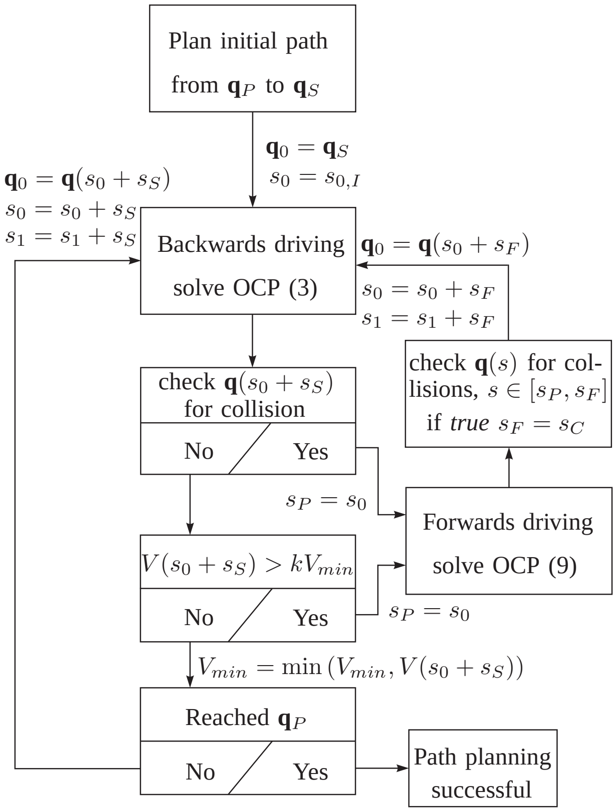


Fig. 5: Flow chart of the path planning algorithm.

TABLE I: Parameters of the truck-trailer system.

L_1	L_2	L_3	M_1	δ_{max}	$\beta_{3_{max}}$	$\beta_{2_{max}}$
2m	1m	1.8m	1.2m	30°	75°	40°

F. Path Planning Algorithm

Fig. 5 summarises the proposed path planning algorithm for the truck-trailer system in form of a flow chart. Thereby, the collision checking block after the forward driving checks all configurations $\mathbf{q}(s), s \in [s_P, s_F]$ of the forward driving path for collision. If a collision occurs the method returns *true* as well as the last path position s_C along the forward driving path which is not in collision. The termination condition is applied if the parking configuration $\mathbf{q}_P$ is reached within a small numerical tolerance.

IV. Simulation Studies

To show the feasibility of the path planner, several simulation studies were carried out using MATLAB on an Intel Core i7 3.4 GHz machine where only one core was used for the calculation. The corresponding parameters of the truck-trailer system are given in Table I.

In order to solve the OCPs (3) and (9), a full discretisation with N_D discretisation points is employed and the resulting static optimisation problems are solved by means of the SQP-solver of the numeric software package SNOPT [17]. The parameters of the OCPs are shown in Table II, where diag$(\cdot)$

Post-print version of the article: P. Zips, M. Böck, and A. Kugi, "An optimisation-based path planner for truck-trailer systems with driving direction changes", in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 630–636. DOI: 10.1109/ICRA.2015.7139245
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

TABLE II: Parameters of the OCPs.

$\mathbf{Q}$	r_F	d_0	S
$\mathrm{diag}(1, 1, 13, 0.5, 0.5)$	10	$0.1\mathrm{m}$	$[5.5, 6.5]\mathrm{m}$
k	r_{θ_3}	N_D	$\mathbf{R}$
1.5	1	20	$\mathrm{diag}(1, 1, 13)$

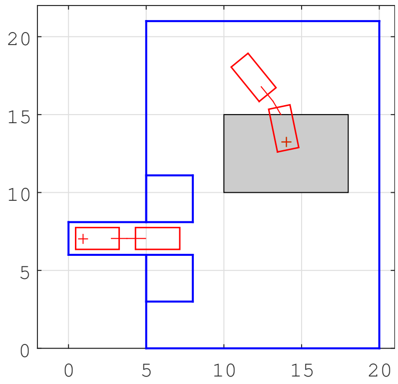


Fig. 6: Simulation environment for the loading bay.

denotes a diagonal matrix. It turns out that the convergence behaviour of the OCP (3) can be improved by considering a free path length constrained by a small set. Collision checking is done using the Minkowski-sum [4].

In the following, two different simulation studies for the loading bay scenario are presented. The first one is depicted in Fig. 6. Thereby, 10^4 runs are simulated with a random starting configuration where the reference point P_R lies inside the shaded area. It is presumed that the truck is directed upwards at the starting configuration so that the tree-based path planner is able to find an initial path without driving direction changes. The ranges of the angles for the starting configuration are $\theta_3 \in 90° + [-30°, 30°]$, $\beta_3 \in [-15°, 15°]$ and $\beta_2 \in [-15°, 15°]$. A possible starting configuration is shown in Fig. 6 where the $+$ corresponds to the reference point P_R. In this simulation scenario, the tree-based path planner finds an initial path in 99.5% of all considered cases and the path planning is successfully completed in 97.5%. The average calculation time of all 10^4 runs amounts to $t_C = 0.2s$. Due to the random character of the initial path planner a recalculation of the failed paths for a second time yields success rates of the path planning greater than 99%.

Fig. 7 presents one representative of the planned paths. The left subfigure shows the path of the trailer, where the solid line corresponds to the initial path and the dashed line to the final path. The forward driving path is highlighted by the bold line. The remaining figures depict the animated final path with a backward, forward and another backward path from the left to the right. The truck is illustrated as a dashed and the trailer as a solid rectangle. The shade of grey gets darker with increasing path parameter s.

In Fig. 8, the steering angle δ as well as the angles β_3 and β_2 are plotted over the path parameter s for the path presented in Fig. 7. Driving direction changes are marked with a vertical line. Mainly in the forward driving segment quick changes of the steering angle can be observed. Although rapid changes in the steering angle can be realised by a slower driving speed, a further path smoothing, as for instance the strategy applied in [18] to cars, can be considered to avoid fast steering. Both angles β_3 and β_2 are smooth and well below their constraints for the whole path.

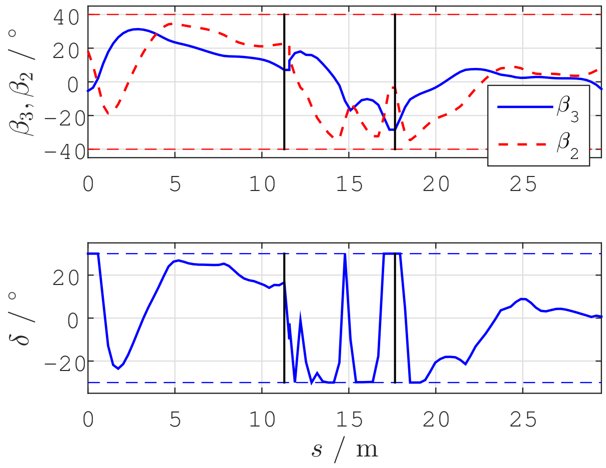


Fig. 8: Difference angles β_3 and β_2 and steering angle δ for the loading bay scenario.

In the second simulation study, a 2m long quadratic obstacle is placed inside the working space. Fig. 9 demonstrates the simulated scenario again with the initial and final path in the left subfigure and the animated path to the right. Here, the system cannot pass the obstacle in the first backward driving segment. After repositioning with a forward driving segment the path planner manages to find a feasible path into the loading bay. For different starting configurations more than 2 driving direction switching points may be necessary which is not presented in this paper due to page restrictions. Nevertheless, the path planner manages to find a path with an arbitrary number of direction changes.

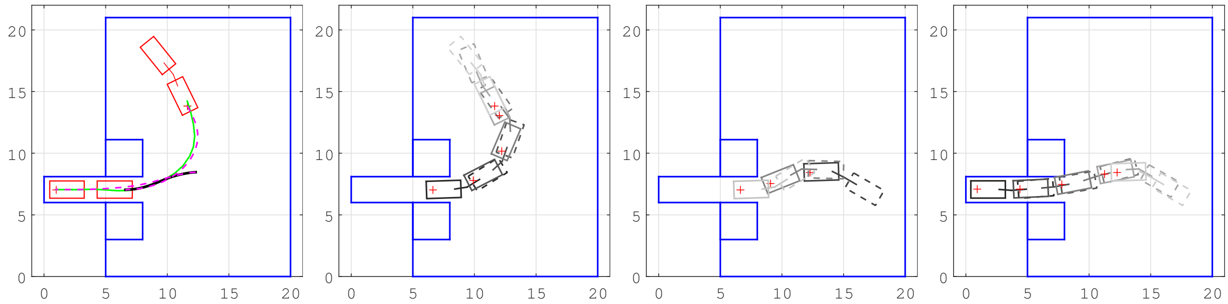
V. Conclusions

In this paper, a path planning approach for a truck-trailer system with kingpin hitching was presented. Thereby, an initial path obtained by a tree-based planner is followed locally in a receding horizon fashion using an optimal control problem. Repelling springs push the path away from obstacles. Based on the local behaviour of the planner, two heuristic rules for direction changing were introduced.

Monte Carlo simulation studies demonstrate the high convergence rate at low computational costs for the proposed planner. Even for starting positions with high discrepancies in the angles in a narrow loading bay scenario, feasible paths can be found. However, several questions still remain to be answered in future work.

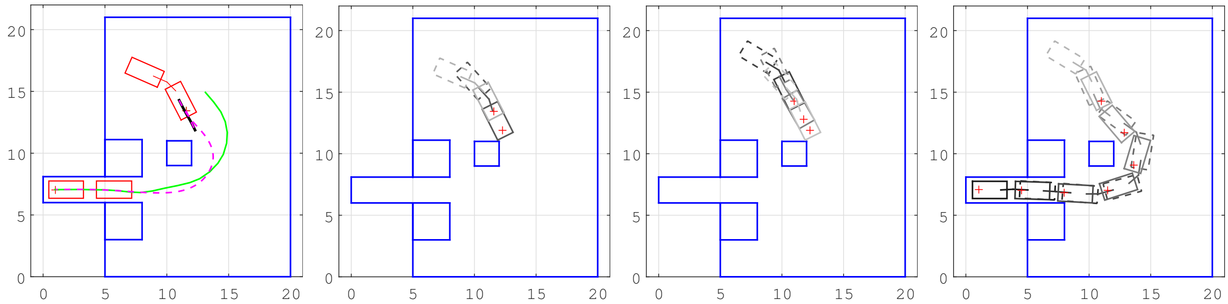
First, a different strategy for calculating the initial path will be investigated to be able to handle more complex scenarios. For example a combination of the tree-based planner with

Post-print version of the article: P. Zips, M. Böck, and A. Kugi, "An optimisation-based path planner for truck-trailer systems with driving direction changes", in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 630–636. DOI: 10.1109/ICRA.2015.7139245



(a) Initial and final path. (b) Backward driving. (c) Forward driving. (d) Backward driving.

Fig. 7: Animated path for a loading bay scenario. The shade of grey gets darker with increasing path parameter.



(a) Initial and final path. (b) Backward driving. (c) Forward driving. (d) Backward driving.

Fig. 9: Animated path for a loading bay scenario with additional obstacle. The shade of grey gets darker with increasing path parameter.

a holonomic path planner may lead to an improvement. Furthermore, scenarios with a longer driving distance as well as realistic conditions in terms of parameter and sensor uncertainties will be examined in more detail. The receding horizon character of the proposed path planner also allows a recurrent replanning which may be initiated by sensor uncertainties or dynamic obstacles.

References

[1] D. Tilbury, R. Murray, and S. Sastry, "Trajectory generation for the N-trailer problem using Goursat normal form," *IEEE Trans. Autom. Control*, vol. 40, no. 5, pp. 802–819, 1995.

[2] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *Int. Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[3] C. Altafini, "Some properties of the general n-trailer," *Int. Journal of Control*, vol. 74, no. 4, pp. 409–424, 2001.

[4] T. Lozano-Pérez, "Spatial planning: A configuration space approach," *IEEE Trans. Comput.*, vol. C-32, no. 2, pp. 108–120, 1983.

[5] P. Ŝvestka and J. Vleugels, "Exact motion planning for tractor-trailer robots," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 1995, pp. 2445–2450.

[6] C. Cariou, R. Lenain, B. Thuilot, and P. Martinet, "Autonomous maneuver of a farm vehicle with a trailed implement: motion planner and lateral-longitudinal controllers," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, USA, May 2010, pp. 3819–3824.

[7] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *IEEE J. Robot. Autom.*, vol. 20, no. 6, pp. 967–977, 2004.

[8] O. Lefebvre and F. Lamiraux, "Docking task for nonholonomic mobile robots," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Orlando, FL, USA, May 2006, pp. 3736–3741.

[9] S. Sekhavat and M. Chybo, "Nonholonomic deformation of a potential field for motion planning," in *Proc. IEEE Int. Conf. an Robotics and Automation*, Detroit, MI, USA, May 1999, pp. 817–822.

[10] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. Int. Conf. on Robotics and Automation*, San Diego, CA, USA, May 1994, pp. 3310–3317.

[11] R. Stahn, T. Stark, and A. Stopp, "Laser scanner-based navigation and motion planning for truck-trailer combinations," in *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Zurich, Switzerland, Sep. 2007, pp. 1–6.

[12] S. Sekhavat and J.-P. Laumond, "Topological property for collision-free nonholonomic motion planning: the case of sinusoidal inputs for chained form systems," *IEEE J. Robot. Autom.*, vol. 14, no. 5, pp. 671–680, 1998.

[13] R. Kong, "Accurate parking planning of tractor-trailer-trailer mobile robot," in *IEEE Proc. Int. Conf on Mechatronics and Automation*, Chengdu, China, Aug. 2012, pp. 1220–1226.

[14] A. W. Divelbiss and J. Wen, "A global approach to nonholonomic motion planning," in *Proc. Conf. on Decision and Control*, Tucson, AZ, USA, Dec. 1992, pp. 1597–1602.

[15] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," in *Proc. 4th Int. Workshop on the Algorithmic Foundations of Robotics*, Hanover, NH, USA, Mar. 2000, pp. 247–264.

[16] P. Zips, M. Böck, and A. Kugi, "A fast motion planning algorithm for car parking based on static optimization," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013, pp. 2392–2397.

[17] P. E. Gill, W. Murray, and M. A. Saunders, *Users Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, Dept. of Mathematics, University of California, San Diego, CA, USA, Feb. 2006.

[18] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

Post-print version of the article: P. Zips, M. Böck, and A. Kugi, "An optimisation-based path planner for truck-trailer systems with driving direction changes", in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 630–636. DOI: 10.1109/ICRA.2015.7139245