TU/e
EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

/ Department of
Mechanical Engineering

Dynamics and Control

Nonlinear Path Planning and Tracking Control for Unmanned Trucks at Warehouses

MSc. Thesis
DC 2021.058

Author: S.P. (Stan) van Boheemen

Supervisors: Dr.ir. A.A.J. Lefeber
Dr.ir. I.J.M. Besselink
Prof.dr. H. Nijmeijer

Eindhoven, June 24, 2021

Abstract

This thesis considers path planning and tracking control of autonomously driving tractor semi-trailer combinations around warehouses and distribution centres. These are closed-off, predictable environments with few external influences and low driving speeds. Based on a kinematic single track vehicle model in spatial domain, a kinodynamic path planning algorithm is developed. It aims to yield a feasible and smooth path, taking vehicle constraints and external obstacles into account. It takes the outputs of a pre-existing high-level planner and uses that as initialisation for a nonlinear optimisation problem formulated based on the curvature of the trailer combined axle. This curvature function is parametrised using quintic splines, fully determining the state of the truck in spatial domain. The objective function is a weighted sum of path length, required steering angle and steering angle change. Two options for taking constraints into account are presented, one based on distance functions between simple shape approximations of the truck and its environment and the other on the hyperplane separation theorem between arbitrary convex polygons. Using the same single track vehicle model, a nonlinear reference tracking controller has been designed. Error coordinates defined by shortest distance projection onto the reference path have been chosen, yielding a unique mapping between vehicle and reference state. For both forward and backward driving, a control law has been proposed based on the linearised system for which a constant curvature is assumed. For this situation, sufficient and necessary conditions for gain tuning to warrant local asymptotic stability have been derived. The conversion back to time domain is performed through an optimal velocity profile, taking powertrain, friction and comfort limitations into account. The planner, spatial controller and velocity optimisation have been applied to several challenging scenarios, showing good and reliable reference tracking.

Acknowledgements

In this master's thesis you will find the results of my graduation project which took place in 2020-2021 at the Eindhoven University of Technology. It has been a remarkable year of working from home, where I first met my supervisors and professor at the final presentation. It has not been an easy year study-wise, with me struggling to maintain my discipline and put in the required hours and effort to complete this project. Moreover, working alone made me miss the brainstorming sessions, discussions and suggestions between me and my peers that I would otherwise undoubtedly have had. Due to these circumstances, I think it is safe to say I could not have completed this project on my own, which is why I would like to thank the people that have supported me over the last year.

First of all, I would like to thank prof.dr. Henk Nijmeijer for his guidance, suggestions and interest in my project. Even though I did not always manage to clearly explain exactly what I was working on and why, I always felt that our meetings were useful and my project remained on track accordingly. Moreover, thanks for the pleasant conversations throughout my master's program, no matter the requests I came into your office with.

Secondly, thanks to dr.ir. Igo Besselink for his practical point of view, knowledge and experience with trucks and enthusiasm. A number of times when I started to drown in mathematical expressions, numerical optimisation or formal proofs, a clever comment or suggestion led me back in the right direction. Furthermore, you showed some interesting and engaging videos at the start of the project, clearly describing the problem at hand and the parties involved at distribution centres.

I would also like to express my gratitude towards dr.ir. Erjen Lefeber, who tirelessly supported me throughout the project in our weekly meetings. Despite missing one, our weekly meetings were a great way to structure and document my work, as well as a source of endless ideas and suggestions. Specifically, thanks for performing some mathematical derivations by yourself and ensuring my progression throughout the project.

Furthermore, I have greatly enjoyed the weekly 'koffieuurtjes' with Joris, Niels and Guus. Keeping up with each other's projects, helping where possible and discussing what happened around us helped me more than I would have imagined. Even though Joris was unable to join the last six months, I am glad that we continued our tradition.

Finally, I would like to thank my parents for their continuous support and assistance. During the past year, I have left my room in Eindhoven and moved back to my parents' house, which went seamlessly. I think the three of us have been a great team running our household through these weird times, which worked as the perfect foundation to allow me to complete my project.

Over the past year, I have learnt a lot about trucks, nonlinear control and optimisation, but I think that I have learnt even more about myself. Thanks to everyone who has (partially) read my thesis and provided feedback and to whoever will.

Stan van Boheemen
June 2021

Contents

Nomenclature

Accents

$'$	Spatial derivative
$''$	Second spatial derivative
$\bar{}$	Variation from equilibrium
$\dot{}$	Time-derivative
$\tilde{}$	Closed-loop system

Abbreviations

CQLF	Common Quadratic Lyapunov Function
GUAS	Global Uniform Asymptotic Stability
HL	High-Level
IMU	Inertial Measurement Unit
LiDAR	Light Detection And Ranging
LL	Low-Level
LMI	Linear Matrix Inequality
LTI	Linear Time-Invariant
NMPC	Nonlinear Model Predictive Control
ODE	Ordinary Differential Equation
RRT	Rapidly exploring Random Tree
SAT	Separating Axis Theorem
TNO	Toegepast Natuurwetenschappelijk Onderzoek
TU/e	Eindhoven University of Technology
ZOH	Zero-Order Hold

Sub- and superscripts

$+$	Forward
$-$	Backward
0	Initial
0	Of the tractor
1	Of the trailer
$\mathcal{O}$	Of obstacles
ψ	Articulation angle
ρ	Distance function
att	Attractive
backward	Backward integration
forward	Forward integration
free	Obstacle-free
max	Maximum
min	Minimum
rep	Repulsive
SAT	Of the Separating Axis Method
switching	At the switching point
trailer	Of the isolated trailer system
φ	Heading
b	Of the back
b	Of the bounding box
c	Common Quadratic Lyapunov function

d	Desired
d	Distance
f	Final
f	Of the front
i	Index 1
j	Index 2
k	Index 3
o	Overhang
r	Reference
s	Segments
x	In x-direction
y	In y-direction
y	Lateral

Matrices/Columns

η	Repulsive potential field weight
ξ	Attractive potential field weights
e	Error vector
I	Identity matrix
P	Quadratic Lyapunov function matrix
p	Curvature parameter vector
Q	Lyapunov equation matrix
q	State of the truck
T	Transformation matrix

Sets/Maps

$\emptyset$	Empty set
$\mathbb{C}$	Complex numbers
$\mathbb{N}$	Natural numbers
$\mathbb{R}$	Real numbers
$\mathbb{Z}$	Integers
$\mathcal{A}$	Truck footprint
$\mathcal{O}$	Obstacle
$\mathcal{Q}$	Configuration space

Operators

$(\cdot)'$	Spatial derivative
$\cap$	Difference of sets
$\cup$	Union of sets
Δ	Difference
δ	Increment
$\dot{(\cdot)}$	Time derivative
$\equiv$	Is defined as
$\forall$	For all
$\gtrless$	Greater for forward, smaller for backward
$\in$	Is element of
$\pm$	Positive for forward, negative for backward
$\sim$	Varies with
d	Derivative
max	Maximum
min	Minimum
sup	Supremum
$\rightarrow$	Maps to

Symbols

α	Cost function weight	
α	Spline coefficients	
β	Tractor side-slip angle at kingpin joint	[rad]
δ	Steering angle	[rad]
∞	Infinity	
κ	Curvature	[1/m]
λ	Eigenvalue	
$\mathcal{C}$	Continuity	
$\mathcal{O}$	Origin	
ψ	Articulation angle	[rad]
σ	Sigmoid function	
τ	Optimisation variable	
θ	Virtual steering angle of the trailer	[rad]
φ	Orientation	[rad]
a	Acceleration	[m/s/s]
a	Characteristic polynomial coefficients	
b	Bounding box lines	
c	Constant/gain	
d	Safety distance	[m]
e	Unit vector	
F	Collecting function	
G	Gap distance	[m]
J	Cost function	
k	Time step	
L	Length	[m]
P	Characteristic polynomial	
p	Control point position	
R	Reference points	
S	Integral of sigmoid function	
s	Distance	[m]
t	Time	[s]
U	Potential field value	
V	Lyapunov function	
v	Velocity	[m/s]
x	x-position	[m]
Y	Auxiliary Lyapunov function	
y	y-position	[m]

Chapter 1

Introduction

"Without transport all stands still" was a Dutch advertisement campaign initiated in 2009 [1]. There is no doubt that much of our modern-day society is highly dependent on fast, reliable and efficient transport and logistics. Moreover, the Netherlands has established itself as a transport hub, between the main ports of Schiphol Airport, the Port of Rotterdam and the European mainland. Even though the slogan might be a slight hyperbole, transport and logistics are essential in industry, agriculture and e-commerce. Besides, the added value of the Dutch logistics industry itself is €40 billion annually and employs more than a quarter of a million people [2]. Of the total amount of transported goods within the Dutch borders, 82% is transported over the road measured by weight, which is primarily done by trucks [3]. Even though the 2020 pandemic had a large negative effect on the sector, the outlook for the future is still growth [4],[5]. One of the looming challenges for Dutch road transport is the relatively high age of its employees, where the group of 50-60 years old is significantly larger than those aged 30-40 years [4]. Furthermore, the extremely rapid growth of e-commerce has led to a significant increase in urban cargo, which is highly reliant on fast and efficient distribution networks. Specifically, the rise of large distribution centres has seen rapid growth over the last decade [6].

Technological advances have been focused on automation, electrification and improved communication between companies [7]. Automation in warehouses is increased through robotics, autonomously picking and sorting goods, and automation of trucks, including autonomy and advanced driver-assist systems. Attention has been given to trucks on highways, with a focus on lane-keeping, safety and platooning, since highways are a relatively predictable and clear environment [8],[9]. One might argue, however, that docking stations of distribution centres are even more applicable for automation. Docking areas around warehouses are confined, clearly mapped and have barely any (unpredictable) traffic, while the driving speeds and therefore stopping distances in case of emergency are low.

In the context of logistics automation research, the Automotive Engineering Science laboratory of the Eindhoven University of Technology has started a research project with TNO to create a safe, efficient and robust distribution centre environment. In this environment, trucks are designed to navigate autonomously, dock to the distribution centre and perform parallel parking. Control and planning algorithms have been developed to fulfil these tasks, based on a scaled, kinematic model of a semi-truck trailer combination. In previous work, multiple controllers have been developed based on pure pursuit control, lookahead PID control and input-output linearisation, all combined with inverse kinematics to obtain the tractor inputs [10],[11],[12]. Additionally, the developed algorithms can be tested and validated on a scaled testing environment, shown in Figure 1.1, which is a 1:13 replica of a part of the Jumbo distribution centre in Veghel. This experimental setup consists of the warehouse docks, three semi-trailer trucks, an optical measurement system and central processing units.

The focus of this thesis is the path planning and tracking control of a single tractor semi-trailer combination around a warehouse area. Such a tractor semi-trailer combination should be able to traverse these areas safely, even in the presence of actuator limits and stationary obstacles. The considered vehicle for this project is shown in Figure 1.2 and consists of a tractor and a semi-trailer, with the kingpin joint located between the two tractor axles. The actuators of the truck are the steering actuator, capable of tracking the desired steering angle setpoint, and the motor, for which perfect velocity control is assumed. This means that the inputs of the truck are a desired steering angle and a desired velocity. Furthermore, it is assumed that the full state of the truck is known, including its dimensions, the positions of all axles as a function of time and the articulation angle between truck and trailer through time. In practice, weight, load distribution and rotation axis significantly differ between trucks and are most often unknown.

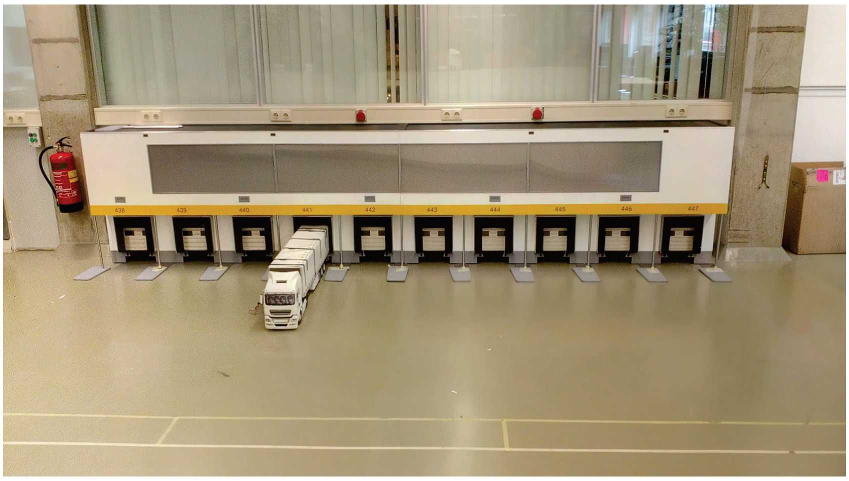


Figure 1.1: Experimental setup at TU/e [11].

Estimating these parameters online is an ongoing research area [13],[14], which is not included in the scope of this thesis.

This chapter continues with an overview of the considered system in section 1.1, establishing the control challenges and considerations. In section 1.2 the current state-of-the-art solutions are presented, where both literature and previous work done at the TU/e is introduced. The research goal and the contributions of this project are defined in section 1.3, before the contours of the thesis are outlined in section 1.4.

1.1 Control challenges

The kinematic model of the tractor semi-trailer combination is a fourth-order dynamical system, with only two inputs. It is assumed that the tractor and the trailer have a fixed articulation point. This point typically lies in between the rear axles and has the property that no sideslip can occur, meaning that only movement in the direction of the wheels is possible. This point is referred to as the simplified rear axle, or rear axle in short. In reality, the presence of multiple non-steered axles and nonlinear tyre behaviour at large side-slip angles complicates the determination of the effective rotation point of the semi-trailer. Determining this point of rotation online does not lie within the scope of this thesis. The no-side slip constraint of the rear axle restricts the movements of the tractor and trailer on a velocity level. In literature, this is referred to as a non-holonomic constraint, meaning that it can not be integrated

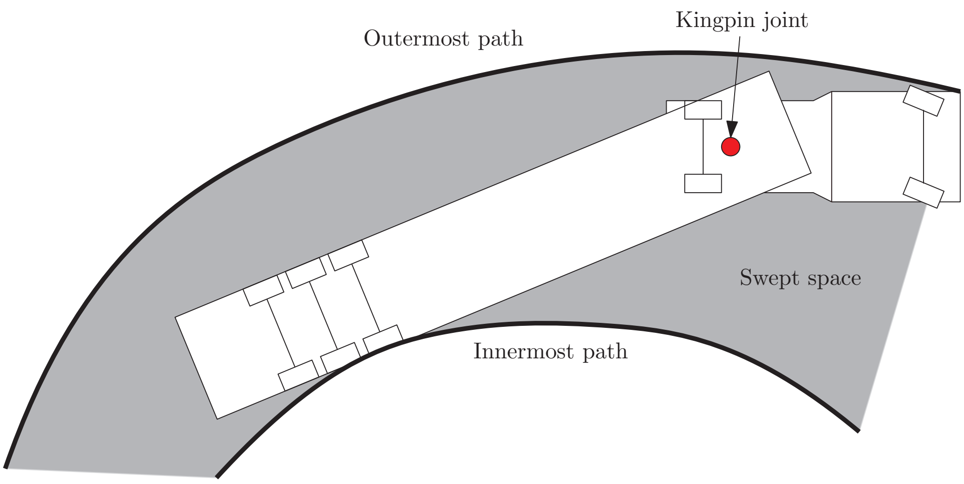


Figure 1.2: Top view tractor semi-trailer combination with swept space.

with respect to time to yield a positional constraint. As a result, this constraint does not allow for simplifications in the dynamics, as still all poses in the state space of dimension four are admissible. What it restricts is the ability to arbitrarily traverse the state space, as these rear axles can exclusively move in the one direction allowed by this constraint. This non-holonomic constraint complicates both the path planning and tracking problems, since not every coordinate can be controlled independently.

One significant difficulty in developing a safe and robust approach is the swept path by the truck as it moves around the warehouse. As indicated in Figure 1.2, the swept area of the truck may be much wider than the truck itself and is dependent on the previously travelled path. Failure to correctly identify this swept area may result in significant damage to the truck and its environment. As a result of this large swept area, it is in general not sufficient to design and stabilise a certain reference trajectory that is only focused on a single point on the truck. This can be illustrated by considering a controller switching from forward to backward motion, keeping the control point on the reference throughout. Even though the control point might be perfectly tracked and maintain a safe distance from any obstacle, this in itself provides no guarantee that the path is collision-free.

Another challenge is the required switching between forward and backward motions of the truck. At the switching point, both the reference and the actual velocity of the truck are equal to zero, which entails that the steering angle can be moved arbitrarily. In order to prevent undesired consequences from hybrid control at these switching points, such as Zeno behaviour, a smooth and uniform transition strategy is required. In addition to issues with singularities in the configuration, the differences in vehicle behaviour between both driving directions is very significant. Firstly, the position of the steered axle switches from all the way at the front of the vehicle to all the way at the back. Additionally, the rear of the tractor and the trailer are unstable when driving backwards. During the process of automatic docking, this instability needs to be completely overcome to accurately position the trailer in front of the docking station. To guarantee stability of the time-varying tracking problem, a Lyapunov-based stability proof is desired to show uniform stability of the tracking error coordinates.

1.2 Current solutions

In this section, the context of this thesis is established. Plenty of research from the automotive industry and research institutes has been performed in the field of vehicle automation. Firstly, an overview of available methods and frameworks from automotive literature is presented, applicable to autonomous trucks. Secondly, the previous work performed at the TU/e is introduced, as well as its limitations and the open challenges that remain.

Literature

In [15], use is made of a direct nonlinear model predictive control (NMPC) algorithm to encompass both path planning and tracking. Based on a discrete kinematic model of the considered autonomous vehicle, a search condition of an optimal controller is defined, which is then iteratively evaluated to seek the optimum. By using the MPC framework, it is very intuitive to take both state and input constraints into account, yielding a satisfactory and feasible solution. In most literature, however, a clear distinction is made between path planning and tracking, where first a reference signal is calculated, after which a controller seeks to track this reference. This division provides the benefit of solving the challenges separately, simplifying the resulting synthesis. In the next paragraph, path planning is explored, after which attention is given to tracking control methods.

Path planning

The path planning problem in the presence of non-holonomic and actuator constraints is often referred to as the kinodynamic path planning problem [16]. There are two main ways of tackling this problem, both with advantages and disadvantages. The first approach is a decoupled trajectory planning approach, which starts with the generation of a collision-free path [17]. Then, this path is transformed to ensure that the non-holonomic and input constraints are met. Finally, a timing function is computed that parametrises the found path with time to yield a continuous function, satisfying the kinodynamic constraints. The second approach is to directly incorporate the constraints into the optimisation. This means that only admissible inputs are considered and that the reference path has to satisfy the dynamics of the vehicle. In this section, the second approach is focused on, since a general and robust method is sought after. Properties such as time-optimality and completeness are lost when decoupling the planning steps.

One common and intuitive planning algorithm is defined by a potential field, indicating a certain penalty function on vehicle states which should be minimised [18]. It is constructed as the sum of an attractive field towards the desired position and a repulsive field away from obstacles. The attractive field may be defined as the squared distance towards the final position, to ensure smooth trajectories and continuous derivatives, for instance according to

$$U_{\text{att}} = \frac{1}{2}||\xi(q - q_d)||^2, \tag{1.1}$$

where ξ is a row vector of weights for each of the coordinates in the vehicle state q, and q_d the desired final pose. This attractive field is commonly gradually transitioned into a linear one, to prevent unbounded attractive forces [18]. These attractive forces can be determined by calculating the gradient of the potential field.

The repulsive field should be equal to zero whenever all points on the truck are far away from an obstacle and increase to infinity in the case of a collision. Consider for example the function

$$U_{\text{rep},i} = \begin{cases} \frac{1}{2}\eta_i \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2 & \text{for } \rho(q) \leq \rho_0 \\ 0 & \text{for } \rho(q) > \rho_0 \end{cases}, \tag{1.2}$$

where $\rho(q)$ describes the shortest distance between the current configuration and any workspace obstacle, ρ_0 is the area of influence of a certain obstacle and η_i is the weight of obstacle i. This continuous function for $\rho_0 \neq 0$ and $\rho(q) \neq 0$ can then be added to the attractive field to yield a map of the configuration workspace with a certain cost associated with each configuration. The strategy is then to follow the negative gradient of this cost function towards the desired final position, which should have the lowest global cost (generally zero). This optimisation can then be performed by a gradient descent algorithm, which searches for a set of optimal waypoints to traverse from the initial to the desired position. The step size should then be manually tuned to prevent the waypoints from jumping into or even across obstacles and keep the required computation time acceptable. A significant disadvantage of potential fields is the possibility of local minima in the potential field. In principle, a simple gradient-based search method does not escape these minima, and hence the path planning process may get stuck. A number of solutions exist to avoid and escape local minima, but it remains a prominent problem with this type of method.

Another method of finding an optimal path of waypoints is by the use of a probabilistic roadmap, which is a network of nodes connected by arc segments [18]. The network of nodes can be sampled by either (quasi-)random sampling or, in the case of restricted workspaces, be predefined. For our application of known and relatively simple areas around warehouses, a sophisticated network can be made and refined by hand. The next step is eliminating all nodes and arcs which cause a collision or are undesirable for other reasons. Finally, a network of possible transitions between each node, from start to final pose, is generated from which the optimal set can be retrieved by simple optimal algorithms, such as Dijkstra's or A-star algorithms [17],[19]. A disadvantage of this approach is that converging arbitrarily close to the optimal path comes at the expense of infinite computation time, as optimality is always restricted by the resolution of the gridded configuration space. A more advanced distribution of points around a found path is possible, but a completely smooth optimal trajectory can not be found within reasonable computation times, meaning that subsequent approximations have to be performed. Also in the case that nodes are defined by hand, scaling or adjusting the configuration space requires an update of the roadmap, which is not readily available.

Another work introduces a path planning method based on the expansion and connection of exploration trees in the configuration space, by integrating input functions [20]. Use is made of linearisation of the dynamics around each node to investigate the kinematic constraints and to randomly attempt multiple input signals. This way, by continuously restricting the path planning algorithm to use the kinematics of the vehicle, input exploration trees guarantee that the constraints are always met and that no collisions can occur. Therefore, this method is applicable to any dynamic system. One downside is that this approach is in general not complete, as it requires infinitely small steps to guarantee that the entire configuration space has been mapped. Moreover, the resulting solutions are in general not smooth, requiring a smoothing algorithm to yield a feasible input signal. One significant advantage of this method is that it samples over the input functions. Even a complicated non-holonomic dynamic object such as a kinematic truck with inputs on acceleration level only has two inputs. By considering longitudinal (velocity) and lateral (steering) control separately, for instance by optimising over distance instead of time, this can be brought down to a single input for both stages. This idea greatly simplifies the path optimisation problem and is used extensively in chapters 3 and 4 of this thesis.

Tracking control

Due to the highly nonlinear nature of the kinematic models for articulated vehicles, finding stabilising reference tracking control laws is no simple task. It has been proposed to consider the trailer as a virtual truck when reversing [21]. This way, control laws for the stable forward driving system can be applied, after which the inverse kinematics of the vehicle are used to translate the resulting motion commands to the tractor. This method applies to both on- and off-axle hitching, which describe the location of the kingpin joint, where it specifically focuses on on-axle hitching. Due to the lack of offset between the revolute joint and the axle of the vehicle, the kinematic restrictions of the axle impose constraints on the achievable actuation angles. The proposed method is applicable to any number and combination of dissimilar on- and off-axle hitched trailers. Due to the complexity of the inverse kinematics, this approach makes it difficult to incorporate constraints on the steering angle. Moreover, due to the reference tracking being focused on a single point, the behaviour of the remaining trailers and tractor is not considered, making collision avoidance difficult.

When considering stabilisation along a line, instead of a point, linearisation approaches remain controllable in all configurations [22]. Consequently, a hybrid controller approach is proposed, with different controllers for driving forward and backwards, based on the linearised kinematics. With this hybrid control law, good tracking results have been obtained for reversed straight-line and circle tracking [22]. Furthermore, convergence is proven for both the nominal and the perturbed system. Here, the trade-off between fast convergence of the nonlinear system towards the desired reference lines and the input saturations poses a problem. Another disadvantage is that it is assumed that the velocity v is a given constant, of which the state and input matrices are a function. Hence, this approach is restricted to steady-state situations, where a line or curve is tracked at constant velocity. For this thesis, however, the situation of changing velocities and even driving directions is essential for proper docking.

It has also been proposed to use an input-output linearisation approach to cope with the nonlinear kinematics as a function of distance [23]. As an output, the deviation from the desired path d is considered, with the control task of stabilising it towards zero. To this end, the output is differentiated three times, which is the relative degree of the considered output, until the input appears. Then, input-output linearisation is performed and a stabilising linear control law is designed. Conditions on the desired reference curvature are imposed to keep the zero dynamics within acceptable bounds. Practically, this means that the steering angle limits nor the corresponding maximum articulation angle between the tractor and the semi-trailer can be exceeded. The complexity of the resulting control law, however, requires all system parameters and states to be known or measured with high precision. In practical experiments with a car and trailer, the tracking performance was kept within 20 [cm] of the reference point, which is much better than an average human driver [23].

Another systematic nonlinear control approach is backstepping, which is for instance used in [24],[25]. The former concerns mobile robots, where backstepping and Lyapunov's direct method are applied to generate the required inputs. In addition, Lemma 2.1.3 is used to prove that the proposed tracking errors converge to zero, even for a negative semi-definite Lyapunov function derivative.

In control of platooning automobiles, a nonlinear control approach based on a Lyapunov synthesis is developed [26]. Here, the lateral and longitudinal control problems are separated through a conversion from time domain into distance domain. The lateral control problem is then solved using methods from the mobile robot domain, using a diffeomorphism as a mapping between vehicle and reference and expressing error coordinates in the vehicle frame [27]. This conversion into spatial domain and the accompanying simplifications yield good results and allow a relatively simple and intuitive controller approach. This method, however, has not yet been applied to tractor semi-trailer combinations, whose dynamic properties are principally different. The longitudinal control task is a constant time gap spacing policy, which is not relevant for this thesis.

TU/e research

The motivation to start an autonomous truck docking project at the TU/e originated with the analysis of high capacity vehicles [28]. This work identifies backward parking as one of the most problematic scenarios for professional truck drivers. Insufficient driver view, lack of awareness about controllability limits and the unstable vehicle behaviour of trucks are argued to be the main contributors to the issue with docking. This observation has led to an increase in attention towards driver support systems for docking of articulated vehicles, which takes over the path planning and tracking control responsibilities of the driver. For the path planning part, the path is defined as a set of reference points R_i, where optimality is defined as a path of minimum length between the initial pose R_0 and the terminal pose R_n. A distinction is made between the forward and backward driving paths, which are optimised over separately, based on the common switching point at which the truck is stationary. The proposed algorithm proved to be fast in calculation and applicable to arbitrary articulated vehicle combinations. However, it imposes no continuity requirements on the path and exhibits jumps in the required steering angle signal. The bi-directional tracking controller is based on vehicle kinematic behaviour and a combination of proportional integral control actions, for which the closed-loop stability is guaranteed using a linearised system. It acts on two look-ahead points to account for the lateral and orientation errors with respect to the reference path, similar to how human drivers assess the pose of their truck. The controller gains and preview distances are tuned manually based on the current truck configuration. For this tuning, a step-wise approach is developed, based on the logic of the Ziegler-Nichols method. Shortcomings of this research lie in the simplicity of the path, which does not include transient behaviour between straight lines and cornering situations. Besides, the resulting steering and yaw behaviour is oscillatory, which is undesirable as it leads to unnecessary wear on tyres and actuators and makes the truck less predictable.

Within the TU/e project, the first steps towards truck modelling and control have been made, focusing specifically on tractor semi-trailer combinations [10]. Here, a preview point located at a fixed distance behind the truck is defined, which is compared to the nearest point on the path, yielding an angular error. A simple proportional controller is applied to eliminate this error, which is then converted by inverse kinematics with the small-angle assumption to the desired steering input. By trial and error, the preview length and proportional gain are tuned to yield stable results in simulation. Even though multiple attempts have been made to obtain optimal stabilising gains, the resulting simulations did not show good tracking behaviour. Each corner was initially overshot before the controller managed to steer the errors back to zero. Besides, the small-angle assumption is in practice not valid for the steering angle, nor for the articulation angle between tractor and trailer. Since space is often limiting, large steering angles are generally required to traverse these areas.

Consequently, expansions on this work were made, with the focus on reverse docking and parallel parking [11],[12]. The main contribution is considering multiple lateral errors at the lookahead point instead of a single heading error. Here, a distinction is made between the lateral error at the rear axle, the heading error and the error due to the curvature of the path. These three errors can then be independently tuned by gains to have more control over the qualitative behaviour of the tracking. Again, the calculated virtual steering angle nor articulation angle is equal to the tractor steering angle. For this conversion, an input-output linearisation approach to cope with the nonlinear kinematics is applied in [11], whereas the same inverse kinematic relation as [10], but without the small-angle assumption has also been proposed [12]. In addition to feedback modifications, a kinematic curvature-based feedforward approach was implemented to compensate for error excitation through the reference signal. Still, the proposed control laws are incapable of perfect tracking due to the lookahead point being used as the measurement point for the error coordinates, leading to significant errors at the dock.

In addition to tracking, path planning in both forward and backward driving directions was also considered [11]. For the backward path towards the dock, an arrangement with straight lines and circular arcs was proposed to achieve a minimum distance trajectory. The forward path was generated next, using as final pose the start of the backward path. Here, use was made of a modified rapidly exploring random tree (RRT) algorithm, to cope with obstacle avoidance, input constraints and non-holonomic constraints. Good results were obtained in simulation, as well as in practice, for the proposed path planning. Open challenges remain for the continuity of the generated path, on which no requirements have been imposed. Besides, the switching point, from which the truck initiates backward parking, is simplified to be perpendicular to the parking spot, with articulation and steering angles equal to zero. In practice, space limitations of the warehouse docking area and the focus on spending as little time as possible on parking make this approach infeasible.

The latest contribution to the autonomous trucks project has been made on the topic of path planning, providing a comprehensive overview of available strategies and algorithms [29]. It proposes a graph-search A-star algorithm as a high-level planner for the trailer. Then, the waypoints generated by this path are used to streamline the closed-loop RRT algorithm, which is referred to as the low-level planner. Consequently, a study on simulation results and the computational complexity related to the discretisation steps has been performed, yielding a functional path planner in both forward and reverse directions, capable of obstacle avoidance. However, satisfying the required accuracy leads to a computationally expensive solution, preventing the approach to be implemented in online applications. Moreover, there are no continuity requirements imposed on the generated path, yielding a highly oscillatory steering reference signal.

In addition to contributions towards the control algorithms on the trucks, work has been done on developing the TruckLab experimental setup [30],[31]. Firstly, [30] determined the properties and dimensions of the two radio-controlled trucks used for scaled testing. Consequently, a MATLAB implementation containing a control architecture capable of taking over the remote control of the trucks was developed. Initially, the control was done by a pedal and wheel setup, primarily used in gaming, whereas this was later converted to hardware-in-the-loop using the previously developed control algorithms. Due to the significant amount of play and inaccurate actuation of the steering system of these trucks, [31] started on the development of an improved experimental setup. Here, the choice was made not to start with an industrial toy truck, but to develop the hardware in-house, based on an existing mobile robot platform. The major improvements are found within the steering system, the propulsion system, the software operating system and the inclusion of a LiDAR sensor and IMU.

With the previous work and state-of-practice of the control of autonomous trucks at the TruckLab, the challenges for this thesis have been introduced and outlined. In the next section, these challenges form the basis of the research goal, as well as for the requirements and preferences of the several supplementary goals of the project.

1.3 Research goal

The main goal of this research is defined as:

> ***"Develop a path planning and tracking control algorithm for docking autonomous tractor semi-trailer combinations at warehouses."***

From this goal, several subgoals can be identified, outlying the requirements and challenges towards a future practical implementation. First of all, the considered autonomous trucks are tractor semi-trailer combinations, as these are most often used near warehouses and have been used for the experimental implementation [31],[32]. Generality towards multiple articulated vehicles, for instance with dollies or more trailers, is preferred but not the main focus of this thesis. Successful docking for full-sized vehicles is defined as reverse parking the truck towards the docking station, within the tolerances of 5 cm laterally and 0.5 deg in heading compared to the perpendicular centreline [29]. Secondly, the path planning algorithm needs to contain static obstacle avoidance, while considering the entire swept path of the vehicle, as introduced in Figure 1.2. Additionally, to prevent the oscillatory and rash steering of previous implementations, it should guarantee continuity up to the steering angle velocity. Regarding the velocity signal, it should be at least once continuously differentiable, with the additional constraint of feasible longitudinal accelerations. Moreover, the kinodynamic constraints imposed by the geometry of the truck must be respected at all times, with the assumption that no lateral slip occurs at any axle. Furthermore, the path planning algorithm should be adjustable for different vehicle dimensions, and preferably also tunable for different vehicle configurations and wishes of the user. Thirdly, the tracking controller is designed to follow the reference as closely as possible and should to this end be capable of perfect reference tracking. In order to be relevant in practice, it needs to be predictable, tunable and preferably intuitive. Uniform asymptotic stability towards the reference is required for the set of feasible initial conditions expected to be encountered at a realistic warehouse. Finally, the developed algorithms are to be tested at the 1:13 scaled setup within the TruckLab. Here, the real-time applicability and the robustness to measurement and modelling errors are tested.

The research goal specifies the requirements and preferences for designing a suitable path planner and tracking controller for practical implementation. In short, this project aims to extend and improve upon existing path planning and tracking control algorithms of autonomous trucks at distribution centres. It should consider the swept path of the truck to guarantee collision-free movement, satisfy kinematics and actuator constraints, be robust to initial offsets and be proven uniformly asymptotically stable.

1.4 Thesis outline

With the background of the project and the available results from literature established, this thesis proposes a new and complete path planning and control strategy for autonomous tractor semi-trailer combinations around warehouses. Starting, the preliminaries chapter introduces the necessary mathematical foundations and theoretical results which are used throughout the thesis. Additionally, it contains the derivation of the dynamic model of the tractor semi-trailer combination based on kinematic relations. Consequently, in the path planning chapter, an algorithm is proposed that generates feasible reference paths with arbitrary initial and final conditions. Using these reference paths, the path tracking control chapter describes the analysis and controller synthesis of a robust and reliable control strategy. Finally, the main conclusions of this thesis are drawn and recommendations for future research are made.

Chapter 2

Preliminaries

In this chapter, mathematical and control concepts are introduced which are used in this thesis. First, mathematical and geometrical properties of functions and polygons are given. Next, the concept of Lyapunov stability for nonlinear systems is considered and how to prove uniformity using a Matrosov-like approach. Finally, properties of LTI systems, including controllability and stability are introduced. In addition to these mathematical concepts, the coordinate convention of the tractor semi-trailer system is proposed, on the basis of which its kinematic relations are derived. These single track vehicle dynamics serve as the starting point for the remainder of this thesis.

2.1 Theorems

Lemma 2.1.1. Continuity [33] *Consider a differentiable function $f : \mathbb{R} \to \mathbb{R}$. If a constant $M \in \mathbb{R}$ exists such that*

$$\sup_{s\in\mathbb{R}} \left| \frac{df(s)}{ds} \right| \leq M$$

then $f(s)$ is uniformly continuous on $\mathbb{R}$.

Definition 2.1.2. Continuous differentiability [33] A function $f : \mathcal{D} \to \mathbb{R}$, with $\mathcal{D} \in \mathbb{R}$, is said to be of class $\mathcal{C}^k$, if $\frac{\mathrm{d}^k f(s)}{\mathrm{d}s^k}$ exists (k times continuously differentiable) and is continuous on $\mathcal{D}$.

Lemma 2.1.3. Barbălat's Lemma [34] *Let $f : \mathbb{R}_+ \to \mathbb{R}$ be a uniformly continuous function. Suppose that $\lim_{t\to\infty} \int_0^t f(\tau)d\tau$ exists and is finite. Then*

$$\lim_{t\to\infty} f(t) = 0.$$

Definition 2.1.4. Convex polygons [35] A convex polygon is a polygon without holes and self-intersecting edges, in which no line segment between two points in the polygon crosses the edge of the polygon. Equivalently, it is a polygon whose interior is a convex set. In a convex polygon, all interior angles are less than or equal to 180 degrees.

Theorem 2.1.5. Hyperplane separation theorem [36] *Let A and B be two disjoint nonempty convex subsets of $\mathbb{R}^n$. Then there exist a nonzero vector v and a real number c such that*

$$\langle x, v\rangle \geq c \qquad \text{and} \qquad \langle y, v\rangle \leq c$$

for all x in A and y in B; i.e., the hyperplane $\langle \cdot, v\rangle = c$, with v the normal vector, separates A and B.

Definition 2.1.6. Sigmoid functions Define $\sigma(s)$ as a continuous monotone function on $\mathbb{R}$, differentiable at $s = 0$, satisfying $\sigma(s)s > 0$ for $s \neq 0$, $|\sigma(s)| \leq 1$ and $\frac{\mathrm{d}\sigma}{\mathrm{d}s}\big|_{s=0} > 0$.

Theorem 2.1.7. Lyapunov stability [37] *Let $x = 0$ be an equilibrium point for $\dot{x} = f(x)$. Let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that*

$$\begin{aligned} V(0) = 0 \quad \textit{and} \quad V(x) > 0, \quad \forall\, x \neq 0 \\ ||x|| \to \infty \Rightarrow V(x) \to \infty \\ \dot{V}(x) < 0, \quad \forall\, x \neq 0, \end{aligned}$$

then $x = 0$ is globally asymptotically stable (GAS).

Theorem 2.1.8. Matrosov's Theorem [38] *Consider a nonlinear, time-invariant system described by the equation $\dot{x} = f(x)$, where $x(t) \in \mathbb{R}^n$, and $f : \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitz continuous. Assume $x = 0$ is an equilibrium point, i.e. $f(0) = 0$. Consider the following conditions.*

- *(M1) There exists a differentiable, positive definite and radially unbounded function $V_0 : \mathbb{R}^n \to \mathbb{R}_+$ such that $\dot{V}_0 \leq 0$.*
- *(M2) There exist two differentiable functions $V_1 : \mathbb{R}^n \to \mathbb{R}$ and $V_2 : \mathbb{R}^n \to \mathbb{R}$, two continuous, positive semi-definite functions $Y_1 : \mathbb{R}^n \to \mathbb{R}$ and $Y_2 : \mathbb{R}^n \to \mathbb{R}$ and a continuous function $F_2 : \mathbb{R} \to \mathbb{R}$, satisfying $F_2(0) = 0$, such that*

$$\begin{aligned} \dot{V}_1 &\leq -Y_1 \\ \dot{V}_2 &\leq -Y_2 + F_2(Y_1). \end{aligned}$$

- *(M3) There exists a positive definite function $\omega : \mathbb{R}^n \to \mathbb{R}$ such that $Y_1(x) + Y_2(x) \geq \omega(x)$.*

Then the following hold. (M1), (M2) and (M3) imply that the equilibrium $x = 0$ of system $\dot{x} = f(x)$ is globally uniformly asymptotically stable (GUAS). Moreover, (M2) and (M3) imply that all bounded trajectories of the system converge to the equilibrium $x = 0$.

Lemma 2.1.9. Controllability [37] *Consider the LTI system $\dot{x} = Ax + Bu$ with $x \in \mathbb{R}^n$, for which the controllability matrix is given by*

$$R = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}.$$

The system is controllable if the controllability matrix R has full fow rank, i.e. $rank(R) = n$.

Theorem 2.1.10. Routh-Hurwitz Theorem [39] *Let $f(z) : \mathbb{C} \to \mathbb{C}$ be a polynomial of degree n with no roots on the imaginary axis. Let us define $P_0(y)$ (a polynomial of degree n) and $P_1(y)$ (a nonzero polynomial of degree strictly less than n) by $f(iy) = P_0(y) + iP_1(y)$, respectively the real and imaginary parts of f on the imaginary line. Furthermore, define p as the number of roots of $f(z)$ in the left half-plane; q the number of roots of $f(z)$ in the right half-plane; $\Delta \arg f(iy)$ the variation of the argument of $f(iy)$ when y runs from $-\infty$ to $+\infty$; $w(x)$ is the number of variations of the generalised Sturm chain obtained from $P_0(y)$ and $P_1(y)$ by applying the Euclidean algorithm; $I_{-\infty}^{+\infty} r$ is the Cauchy index of the rational function r over the real line.*

With the notations introduced above, the Routh-Hurwitz theorem states that:

$$p - q = \frac{1}{\pi}\Delta \arg f(iy) = \begin{cases} +I_{-\infty}^{+\infty} \frac{P_0(y)}{P_1(y)} & \text{for odd degree} \\ -I_{-\infty}^{+\infty} \frac{P_1(y)}{P_0(y)} & \text{for even degree} \end{cases} = w(+\infty) - w(-\infty).$$

Corollary 2.1.11. *The third-order polynomial $P(s) = s^3 + a_2 s^2 + a_1 s + a_0$ has all roots in the open left half plane if and only if $a_2 > 0$, $a_0 > 0$ and $a_2 a_1 > a_0$.*

2.2 Dynamics

In order to control or prescribe feasible paths for a tractor semi-trailer, its dynamic behaviour and response need to be understood. Consequently, such a model can be manipulated such that they behave as desired. To this end, kinematic models are developed that describe the evolution in time of the pose of the truck. Note that the pose is defined as the full vehicle state, including positions in the world frame. Based on the truck topology of Figure 1.2, we define the coordinates describing the pose of the truck as in Figure 2.1. In this section, a single track vehicle model is derived, which serves as a building block for the truck model. Moreover, the most attractive control and projection point, around which the motion equations are defined, is discussed in detail.

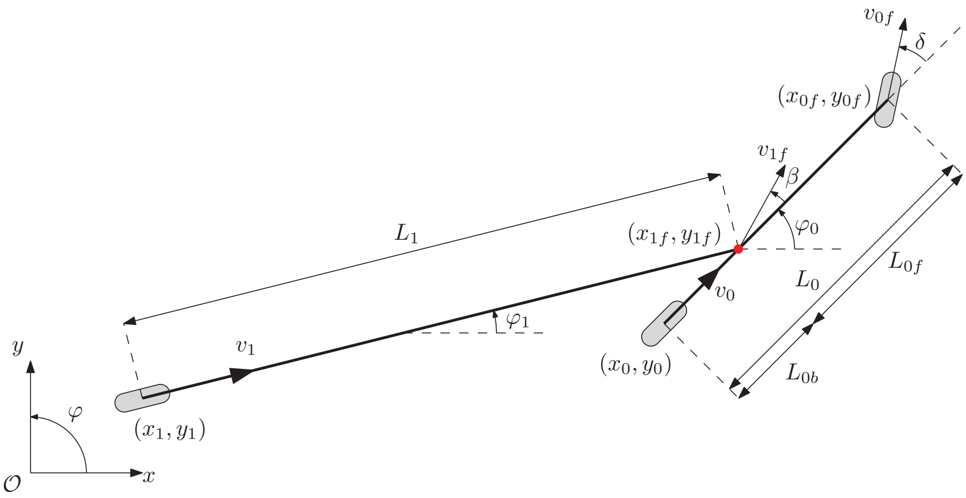


Figure 2.1: Coordinate convention tractor semi-trailer combination.

Single track vehicle model

A single track model, often referred to as a bicycle model, is schematically represented in Figure 2.2. Here, the dimensions and properties of the tractor are taken as an example, but notice that the trailer behaves analogously. The single track vehicle model is often used as a simple representation of dynamic vehicle behaviour when tyre side-slip angles are included, as well as kinematic behaviour at low speeds. In this project, due to the low speeds involved and the assumption of the no side-slip constraint at any of the tractor and trailer axles, the kinematic version is applicable. First of all, it consists of the translational part, given by

$$\begin{aligned} \dot{x}_0(t) &= v_0(t)\cos\varphi_0(t) \\ \dot{y}_0(t) &= v_0(t)\sin\varphi_0(t), \end{aligned} \tag{2.1}$$

with $x_0(t)$ and $y_0(t)$ the coordinates of the tractor rear, $v_0(t)$ the longitudinal velocity of the tractor and $\varphi_0(t)$ the orientation of the tractor with respect to the global inertial frame. These equations describe the movement of the trailer rear in the global frame, for which we know that its only velocity component is in the longitudinal direction, indicated by $\varphi_0(t)$. Regarding the rotational part, the dynamics are given by

$$\begin{aligned} \dot{\varphi}_0(t) &= \frac{v_{0f}(t)}{L_0}\sin\delta(t) = \frac{v_{1f}(t)}{L_{0b}}\sin\beta(t) \\ &= \frac{v_0(t)}{L_0}\tan\delta(t) = \frac{v_0(t)}{L_{0b}}\tan\beta(t), \end{aligned} \tag{2.2}$$

where use is made of the relationships $v_0(t) = v_{0f}(t)\cos\delta(t)$ and $v_0(t) = v_{1f}(t)\cos\beta(t)$. Here, $v_{0f}(t)$ and $v_{1f}(t)$ are the velocities at the tractor front wheels and the kingpin joint, respectively. Furthermore, we have the length between the tractor axles L_0 and between rear axle and kingpin joint L_{0b}. Notice that from the equations above, we can relate the steering angle $\delta(t)$ to the side-slip angle β, through

$$\delta(t) = \tan^{-1}\left(\frac{L_0}{L_{0b}}\tan\beta(t)\right). \tag{2.3}$$

This means that these angles can always be statically related to one another, which allows input constraints to be transferred as well. Moreover, this entails that both the planning and control problems could be solved either in terms of $\delta(t)$, or in $\beta(t)$, as they are interchangeable through (2.3).

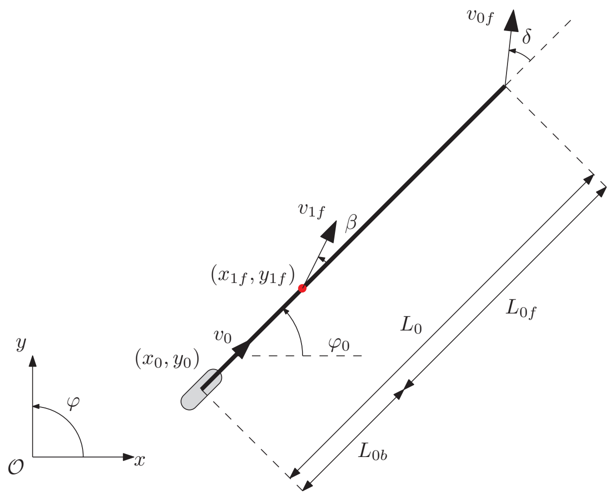


Figure 2.2: Single track vehicle model convention.

In order to proceed towards the kinematics of the entire tractor semi-trailer system, we derive the kinematics of the trailer analogously to those of the tractor. We then obtain

$$\begin{aligned}
\dot{x}_1(t) &= v_1(t)\cos\varphi_1(t)\\
\dot{y}_1(t) &= v_1(t)\sin\varphi_1(t)\\
\dot{\varphi}_1(t) &= \frac{v_{1f}(t)}{L_1}\sin(\beta(t)+\varphi_0(t)-\varphi_1(t))\\
&= \frac{v_1(t)}{L_1}\tan(\beta(t)+\varphi_0(t)-\varphi_1(t)),
\end{aligned} \tag{2.4}$$

where use is made of the relationship between

$$v_{1f}(t) = \frac{v_1(t)}{\cos(\beta(t)+\varphi_0(t)-\varphi_1(t))}. \tag{2.5}$$

This leaves six dynamic equations describing the evolution of the truck. To reach a minimal description, we can omit two of these coordinates through the holonomic constraints imposed by the kingpin joint, shown in Figure 2.1. The choice of which coordinates to eliminate from the model is non-trivial and based on the choice of the control point.

Control point

We have derived the kinematics describing the behaviour of a single track vehicle model, the tractor and semi-trailer. The two holonomic constraints originated from the connection at the kingpin joint are given by

$$\begin{aligned}
x_1(t) + L_1\cos\varphi_1(t) &= x_0(t) + L_{0b}\cos\varphi_0(t)\\
y_1(t) + L_1\sin\varphi_1(t) &= y_0(t) + L_{0b}\sin\varphi_0(t),
\end{aligned} \tag{2.6}$$

allowing us to eliminate two dynamical equations and still uniquely prescribe the pose of the truck. Not all arbitrary sets of four coordinates yield this unique definition, however. Consider for example the set of four position coordinates describing the tractor and trailer rear axles, which still permit two symmetrical configurations, together forming a kite shape. Instead, we take the coordinates at either truck or trailer rear, accompanied by the orientations of both components. Either of these choices yields a unique representation of the system, shown below as

$$\begin{aligned}
\dot{x}_0(t) &= v_0(t)\cos\varphi_0(t) \\
\dot{y}_0(t) &= v_0(t)\sin\varphi_0(t) \\
\dot{\varphi}_0(t) &= \frac{v_0(t)}{L_{0b}}\tan\beta(t) \\
\dot{\varphi}_1(t) &= \frac{v_1(t)}{L_1}\tan(\beta(t)+\varphi_0(t)-\varphi_1(t))
\end{aligned} \tag{2.7}$$

with the focus on the tractor rear axle or

$$\begin{aligned}
\dot{x}_1(t) &= v_1(t)\cos\varphi_1(t) \\
\dot{y}_1(t) &= v_1(t)\sin\varphi_1(t) \\
\dot{\varphi}_1(t) &= \frac{v_1(t)}{L_1}\tan(\beta(t)+\varphi_0(t)-\varphi_1(t)) \\
\dot{\varphi}_0(t) &= \frac{v_0(t)}{L_{0b}}\tan\beta(t),
\end{aligned} \tag{2.8}$$

which takes the trailer rear as the control point. Here, the choice is made to continue with the latter set of equations, where the trailer rear is the control point and all velocities are to be expressed in $v_1(t)$. Reasons for this choice are, first of all, that the most challenging scenario for a tractor semi-trailer combination is reversing. Here, the $\varphi_1(t)$-dynamics would become unstable with respect to the kingpin joint (substituting $v_1(t) < 0$ in the φ_1-dynamics of (2.8)), as we would expect for a reversing trailer. Manipulating the trailer orientation or position requires relatively large steering efforts, which can not directly be controlled by the steering angle. Besides, taking only the position of the tractor rear and the orientations of the tractor and trailer leads to significant error propagation and extreme sensitivity to disturbances. Instead, by taking the trailer rear as the control point, we obtain a more straightforward structure. Here, the idea is to design a 'virtual' control law for the trailer rear, meaning a desired input from the kingpin joint. The tractor is then simply required to attain this virtual control, leading to the complete system stabilising around the reference. The resulting controllers are referred to as virtual controllers, as they do not directly act on the input of the system. Secondly, the requirements on docking precision are imposed onto the trailer rear. The main purpose of the planning and control tasks considered in this project are docking the trailer at a docking station. The margins for successful docking are small enough to warrant the control point being placed as close as possible to the point of interest.

Tractor semi-trailer kinematic model

As mentioned in the introduction, the two inputs to this system are the steering angle $\delta(t)$ and the forward velocity of the tractor $v_0(t)$. Additionally, we have stated that the pose of the truck is uniquely defined at each point in time by the four coordinates contained in the tuple $q(t) = (x_1(t), y_1(t), \varphi_1(t), \varphi_0(t))^T$. Consider their dynamics (2.8), where we recognise that both $v_1(t)$ and $v_0(t)$ appear. Provided we can find a relationship between the two, we can express the velocity optimisation problem in either velocity as well, similar to the logic used between $\delta(t)$ and $\beta(t)$. The most intuitive relation would be to equate the respective velocities at the kingpin joint, according to

$$\begin{aligned}
v_{1f}(t) &= \frac{v_0(t)}{\cos\beta(t)} \\
v_{1f}(t) &= \frac{v_1(t)}{\cos(\beta(t)+\varphi_0(t)-\varphi_1(t))},
\end{aligned} \tag{2.9}$$

which can be rewritten as

$$v_0(t) = \frac{v_1(t)\cos\beta(t)}{\cos(\beta(t)+\varphi_0(t)-\varphi_1(t))}. \tag{2.10}$$

Unlike (2.3), this is not a static relationship, meaning that it is affected by the current pose of the vehicle. Moreover, we recognise a singularity at

$$\cos(\beta(t) + \varphi_0(t) - \varphi_1(t)) = 0, \tag{2.11}$$

which corresponds by means of (2.10) to the situation where the trailer axle is stationary, while the tractor is driving circles around it. This is an important practical state constraint to take into account when designing feasible reference trajectories, as is done in the path planning chapter.

In conclusion, we have obtained the dynamics of the tractor semi-trailer system, expressed in the trailer axle coordinates, given by

$$\begin{aligned}
\dot{x}_1(t) &= v_1(t)\cos\varphi_1(t) \\
\dot{y}_1(t) &= v_1(t)\sin\varphi_1(t) \\
\dot{\varphi}_1(t) &= \frac{v_1(t)}{L_1}\tan(\beta(t) + \varphi_0(t) - \varphi_1(t)) \\
\dot{\varphi}_0(t) &= \frac{v_1(t)}{L_{0b}\cos(\beta(t) + \varphi_0(t) - \varphi_1(t))}\sin\beta(t),
\end{aligned} \tag{2.12}$$

all expressed with inputs $\beta(t)$ and $v_1(t)$. Note that both of these inputs have to be transformed to the actual system inputs $\delta(t)$ and $v_0(t)$ to yield a feasible controller.

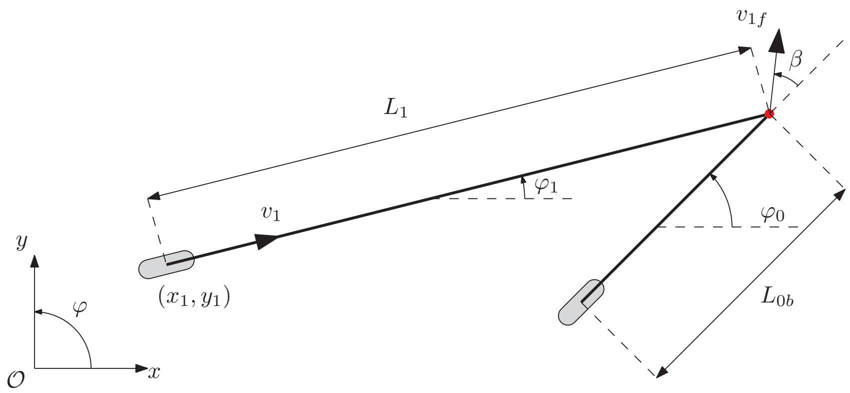


Figure 2.3: Tractor semi-trailer kinematic model representation.

2.3 Conclusion

In this chapter, the required foundations from mathematical and control theory used in this thesis have been introduced. Additionally, a representation of the dynamic behaviour of a tractor semi-trailer combination has been derived, from which relations between actual and virtual steering angles and velocities followed. We now consider the system as displayed in Figure 2.3, consisting of two trailers controlled by the steering angle at the kingpin joint. This dynamical system serves as the starting point of the path planning and tracking control chapters.

Chapter 3

Path planning

This chapter explains the need for a reliable and feasible path planner for navigation around distribution centres and proposes a planning strategy. First of all, section 3.1 introduces the goals, limitations and proposed strategy, including a short introduction of the used high-level planner. Next, the path planning problem is formulated in section 3.2 as a nonlinear optimisation problem. Section 3.3 deals with two implementable methods for obstacle avoidance, before section 3.4 goes in-depth to solving the formulated planning problem. Finally, section 3.5 concludes the chapter.

3.1 Goals and strategy

A control strategy for manoeuvring trucks around distribution centres is the nonlinear control problem to steer a dynamic system from an arbitrary initial condition towards a desired final pose, satisfying its kinematic constraints. The behaviour of the truck is of the class of mobile robots with trailer, in this case connected by off-axled hitching, meaning that $L_{0b} \neq 0$. No controllers have been encountered providing point-to-point control for this type of nonlinear systems, and use is often made of reference paths. Specifically, the controller is designed to suppress and minimise its errors around this reference path. This chapter deals with the challenge of designing a feasible reference path for tractor semi-trailer combinations with the tracking control application in mind. Being feasible indicates that the path should comply with the input and state constraints on the truck, such as articulation and steering angles. Moreover, the path should stay clear of any static obstacles around the distribution centre, to avoid collisions in the case of tracking errors. Dynamic obstacles are not taken into account, which remains for future work. The simplest approach to deal with them would be to wait until they have come to a full stop and then plan a new path around them. In general, satisfying the limitations mentioned above would allow infinite feasible paths from the initial to the final pose. These options are weighted over travel time, tyre wear, smoothness and predictability. This means that one (or multiple) paths are optimal in the sense of a weighted cost function. Intuitively, these preferences can be achieved by minimising the number of switches between forward and backward driving, path length and curvature changes. Furthermore, the velocity profile should be designed such that it maximises the velocity within acceptable bounds on velocity, longitudinal and lateral acceleration, again to be chosen by the customer.

General strategy

As mentioned in the introduction of the report, warehouse layouts are predictable, well-mapped and confined areas, making them suitable for automation. Their layouts are generally simple, with the path towards the dock allowing the truck to drive forward past the dock, before reversing towards the desired final pose. This method of splitting up the planning process into forward and backward driving segments has been successfully applied [28],[11]. Five challenging scenarios have been selected as scaled benchmark tests for the path planner, shown in Figure 3.1 and in Appendix B [29]. Here, the required trajectories are all dictated by surrounding obstacles and a suitable switching point has to be found by the planner.

In order to simplify the construction of a path planner, we first focus on the behaviour of the trailer. By fully defining the motion of the trailer, the next step is to obtain the required pose of the tractor, from which the system inputs $\delta(t)$ and $v_0(t)$ can be inferred. Additionally, we choose to solve the lateral and longitudinal planning separately, by parametrising the dynamics in spatial domain $s(t)$ instead of in time domain t. The last part of the strategy deals with the combination of a rough, general planner with a more accurate and extensive one.

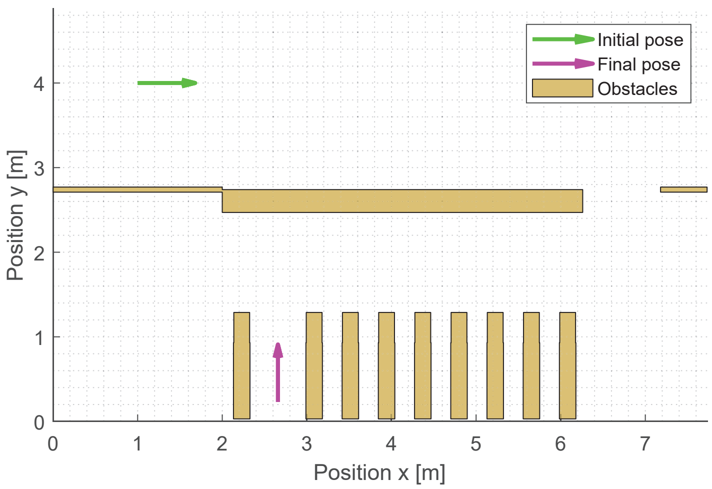


Figure 3.1: TruckLab benchmark scenario for path planner (1:13).

Two-stage strategy

Previous contributions to the TU/e research have proposed to make use of a two-stage planner for the path in spatial domain [29],[40]. This would allow a coarse, fast planner to very quickly give insight into general path characteristics. It could for instance identify how to drive around obstacles, what the approximate total length would be and how many switches between forward and backward driving are required. This information could be used as a starting point for the more precise, but significantly slower planner that would synthesise the eventual smooth path. This strategy is often referred to as high- and low-level planning [40].

The requirements of the high-level (HL) planner are that it should find a path, whenever it exists, from initial to final pose, expressed as a sequence of poses. Moreover, it should remain clear of obstacles, take the most limiting kinodynamic constraints of the truck into account and be able to compute quickly. Within the project, such an HL-planner has previously been developed based on a hybrid A-star algorithm [29]. It assumes a simplified version of the kinematic vehicle model such that the tractor is replaced by the front wheel of a car-like single track model, as in 2.2. Then the rear axle of the truck is assumed to be equal to the rear axle of the virtual car and the kingpin joint becomes the location of the steered wheels of the virtual car. The pose of a single track kinematic car model is completely determined by the tuple $(x_1(k), y_1(k), \varphi_1(k))$, of which the dynamics are discretised and evaluated at time step k, where the pose is transmitted through time by

$$\begin{aligned} x_1(k+1) &= x_1(k) + \Delta t v_1(k)\cos\varphi_1(k) \\ y_1(k+1) &= y_1(k) + \Delta t v_1(k)\sin\varphi_1(k) \\ \varphi_1(k+1) &= \varphi_1(k) + \Delta t\frac{v_1(k)}{L_1}\tan(\varphi_0(k) - \varphi_1(k)), \end{aligned} \tag{3.1}$$

with Δt the time increment, $k \in \mathbb{N}$ the time index and φ_0 the heading of the tractor, in this case the virtual steering angle. The φ_1-dynamics are based on the kinematic relations of section 2.2. With the dynamics established, it is assumed that the vehicle can only make six distinct inputs at each time k, and that these inputs are maintained until $k+1$ (ZOH-assumption). These inputs are driving forward or backward with $v_1 = 1$ with maximum left steer $\varphi_0 = \varphi_{0,\max}$, maximum right steer $\varphi_0 = -\varphi_{0,\max}$ or zero steer $\varphi_0 = 0$. Then from any point in the (x, y)-plane, the car can travel to six nearby poses, for each of which the feasibility concerning obstacles is checked. With these dynamics and input choices defined, a hybrid A-star algorithm can be applied to find the shortest path between any initial and final condition. A-star is an extension of Dijkstra's algorithm, where a heuristic is used to estimate the cost to the final state, often referred to as the cost-to-go [19]. It creates a tree of branches of possible paths from the initial pose, taking into account the cost of the path it has already travelled, as well as the expected cost to the final state. As a result, it performs a much more informed search than Dijkstra's algorithm and maintains its optimality property provided that the heuristic does not overestimate the actual cost from any position to the final state. In the considered HL-planner, the Euclidean distance between start and end is considered, which is less than or equal to any path travelling to this final pose. The algorithm continues until all branches have either converged to the final state or have a higher cost than one feasible path, at which point it is certain that they can not be optimal. Then, the cheapest feasible path is selected, and used as the basis for the low-level planner detailed in the remainder of this chapter. In the situation where no feasible path exists, a maximum iteration counter is exceeded and the planner concludes that it could not find a path within an acceptable time.

The output of the high-level planner is a list containing the tuple $(x_1(k), y_1(k), \varphi_1(k))$ of all points along the path, as well as the required inputs $(v_1(k), \varphi_0(k))$ at each point. When applied to the TruckLab benchmark scenario of Figure 3.1, the resulting output of the HL-planner is shown in Figure 3.2, where it avoids all obstacles and finds a path connecting the initial and final conditions. Disadvantages of the path are the fact that the required inputs are extremely sharp, either having maximum left, zero or maximum right steer. Moreover, the steering is considered to be done by the tractor orientation $\varphi_0(k)$, which can not directly be controlled. As a result, some combinations of inputs can not be realised by the tractor, but might still be present in the path.

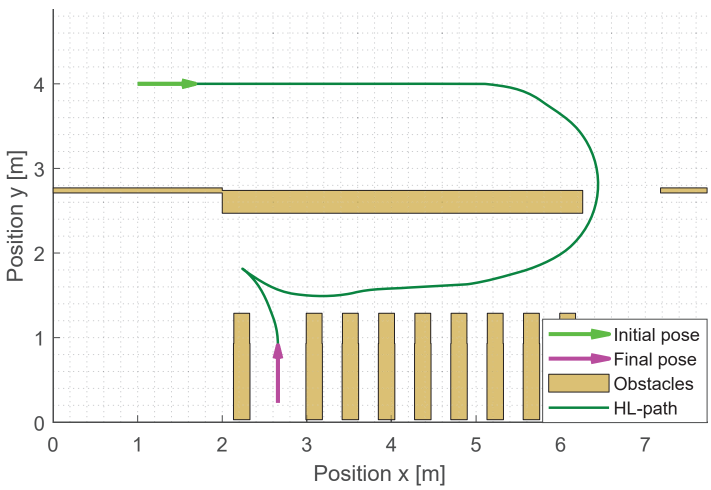


Figure 3.2: HL path for TruckLab scenario.

Based on the high-level path output, we seek to extract as much information out of it as possible, to be used in the low-level planner. First of all, the path is divided into different segments to ease the computations of the low-level planner. The need for this segmentation procedure and how the breakpoints are determined is detailed in section 3.2. Secondly, by investigating the signs of input $v_1(k)$, the switching points between forward and backward driving are obtained. Thirdly, the HL-path provides information on the approximate path length. By summing all intermediate Euclidean distances between the points on the path, the total path length is obtained. Finally, the curvature of the path can be approximated by sampling or averaging over a small domain on the path. This indicates whether a left or right turn is required to traverse the segment, as well as on the sharpness of the turn. In short, based on the HL-path we can obtain breakpoints, switching points, path length and curvature indications, together shaping the initial guess of the low-level path. More detail on the choice of breakpoints can be found in section 3.2, while in section 3.4 the implementation of initial length and curvature is considered.

Spatial-temporal strategy

For the low level planner we once again consider the dynamics of the trailer in continuous time, as derived in section 2.2. It is given by

$$\begin{aligned} \dot{x}_1(t) &= v_1(t)\cos\varphi_1(t), \\ \dot{y}_1(t) &= v_1(t)\sin\varphi_1(t), \\ \dot{\varphi}_1(t) &= \frac{v_1(t)}{L_1}\tan\theta(t), \end{aligned} \tag{3.2}$$

with $\theta(t) = \beta(t) + \varphi_0(t) - \varphi_1(t)$ the virtual steering angle of the trailer, where we recognise that both virtual inputs $v_1(t)$ and $\beta(t)$ appear. Note that these are not actually the inputs of the vehicle and are considered 'virtual' as a consequence. In order to independently solve for both virtual inputs, we recognise that we can rewrite the dynamics into spatial domain by applying $v_1(t) = \frac{\mathrm{d}s(t)}{\mathrm{d}t}$, yielding

$$\begin{aligned} x_1'(s(t)) &= \cos\varphi_1(s(t)), \\ y_1'(s(t)) &= \sin\varphi_1(s(t)), \\ \varphi_1'(s(t)) &= \frac{1}{L_1}\tan\theta(s(t)) = \kappa(s(t)) \end{aligned} \tag{3.3}$$

where coordinate $s(t)$ denotes the travelled distance and $\kappa(s(t))$ is the curvature. The design of the function $s(t)$ indicates the speed with which the truck drives, which is to be optimised using velocity optimisation in section 4.4. This observation, together with the dynamics in (3.3), means that the movement of the trailer in spatial domain is completely determined by the initial conditions $x_1(0)$, $y_1(0)$ and $\varphi_1(0)$ and the curvature function $\kappa(s)$. Since the initial conditions are known, the path to be designed is fully determined by the curvature function, where the explicit dependency of s on time t is dropped for clarity. This is therefore the starting point of the optimisation problem, that is to find the function $\kappa(s)$ yielding the most desirable path, whilst satisfying all requirements and limitations imposed by the path planning problem.

3.2 Path planning problem formulation

With the strategy of the path planning problem introduced, the next step is to formulate the optimisation problem of the low-level planner. The design and implementation of this low-level planner is the contribution of this thesis on the topic of path planning. In this section, attention is first given to the choice of the breakpoints in the segmentation process, before deriving the kinematics of the entire truck. Consequently, the requirements on continuity at the breakpoints are derived, with which the entire low-level path planning optimisation problem can be formulated.

Segmentation

As discussed in section 3.1, the high-level path can be divided into an arbitrary positive number of segments. The breakpoints separating these segments should be chosen such that they minimally limit the pose of the truck, as well as divide the path into segments of approximately equal complexity. In practice, when a truck traverses an arbitrary distribution centre environment, it tends to follow roads (or markings), alternated by sharp steering manoeuvres and crossroads. As a result, placing the breakpoints at the ends of straights encapsulates this alternating behaviour between different parts of the path in a way that minimises the variance of the pose. If the path is to be divided into different segments and the start and end of each segment are enforced to be equal to the breakpoints, the orientation of the truck and trailer at that breakpoint should not be a burden. With the choice of ends of straights, therefore, it comes as no surprise that the tractor and trailer are in line, with small intermediate angle between the two. Mathematically, this follows from the $\varphi_1'(s)$-dynamics of (3.3) with $\beta(s) = 0$ and constant $\varphi_0(s)$. For $s \to \infty$, the trailer orientation converges to $\lim_{s\to\infty} \varphi_1(s) - \varphi_0(s) = 0$. Placing the breakpoint at the end of a straight therefore minimises the articulation angle between tractor and trailer, giving the most freedom for the low-level planner to design the turns. After a breakpoint, the next segment starts with a steering manoeuvre, followed by a straighter part again. Figure 3.3 shows the resulting breakpoints for the TruckLab scenario, where segments of at least 1 m length are required. We recognise that segments of approximately equal length are obtained and that the resulting breakpoints are preferred at ends of straights. Due to the shape of the HL-path, not more segments and accompanying breakpoints are required. Due to the breakpoints lying at the end of a straight, the orientation of the truck is as predictable as possible, namely close to a straight line. The velocity switching points, indicated in red in Figure 3.3, can be trivially extracted from the HL-path by considering the choice of velocity at each time step k.

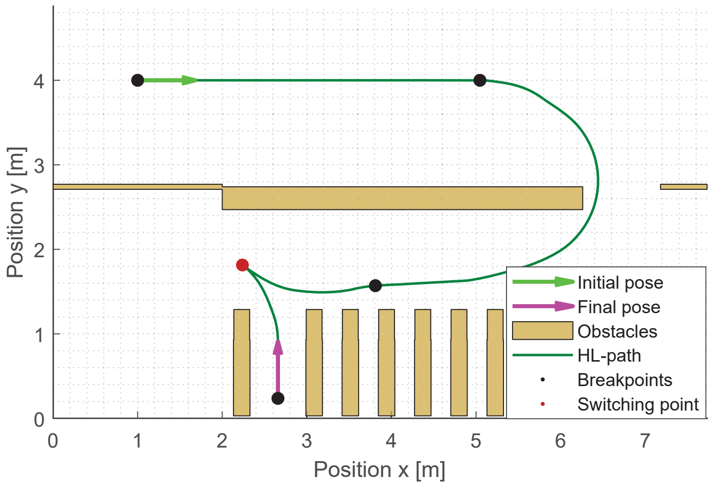


Figure 3.3: Segmentation for HL path TruckLab scenario.

Full truck dynamics

From section 2.2, we know the dynamics of the entire truck, with the trailer rear as control point and expressed in $v_1(t)$, to be that of (2.12), with $\beta(t)$ the side slip angle observed at the kingpin joint. For sake of brevity, we can consider $\theta(t) = \beta(t) + \varphi_0(t) - \varphi_1(t)$ as the trailer virtual steering angle. Recall that $\beta(t)$ and $\delta(t)$ are directly related through the known nonlinear geometric relation (2.3), so finding the required $\beta(t)$ for the above system yields the solution for $\delta(t)$. In order for the path to correctly incorporate penalties and constraints on the input, we require a description of this input as a function of time, which is sought in the following derivation. Starting off, we transition from time domain dynamics to the spatial domain, using $v_1(t) = \frac{\mathrm{d}s}{\mathrm{d}t}$, resulting in

$$\begin{aligned}
x_1'(s) &= \cos\varphi_1(s) \\
y_1'(s) &= \sin\varphi_1(s) \\
\varphi_1'(s) &= \frac{1}{L_1}\tan\theta(s) = \kappa(s) \\
\varphi_0'(s) &= \frac{1}{L_{0b}\cos\theta(s)}\sin\beta(s) \\
&= \frac{1}{L_{0b}}\sqrt{1 + L_1^2\kappa^2(s)}\,\sin\beta(s),
\end{aligned} \tag{3.4}$$

where prime indicates a derivative with respect to s and we have once again assumed positive $v_1(t)$ [41],[26]. The latter assumption entails that $\cos\theta(s) > 0$ for all s, since only then both tractor and trailer are moving in forward direction. Specifically, when $\cos\theta(s) = 0$, an infinite curvature $\kappa(s)$ is obtained and the system is in a singularity. Using the curvature definition of (3.4), we have expressed the $\varphi_0(s)$-dynamics in terms of $\kappa(s)$, as detailed in Appendix B, further eliminating $\theta(s)$ from the dynamics. As mentioned in the strategy section previously, we seek to design the curvature function $\kappa(s)$ as a suitable path. Then, we convert the resulting motion of the trailer to the inputs at the tractor. We therefore consider the curvature function $\kappa(s)$ as input of this system, which together with $\varphi_1(0)$ defines $\varphi_1(s)$. Provided some $x_1(0)$ and $y_1(0)$, the positions $x_1(s)$ and $y_1(s)$ also follow for all s. From this point onwards, we also drop the dependency on travelled distance s for clarity, since all variables except lengths L_{0b} and L_1 are functions of s. Consequently, the position of the kingpin joint is described by

$$x_{1f} = x_1 + L_1 \cos\varphi_1 \qquad y_{1f} = y_1 + L_1 \sin\varphi_1, \tag{3.5}$$

of which the first and second derivatives with respect to travelled distance s, substituting the expressions in (3.4), are given by

$$\begin{aligned} x'_{1f} &= \cos\varphi_1 - L_1\kappa\sin\varphi_1 & y'_{1f} &= \sin\varphi_1 + L_1\kappa\cos\varphi_1 \\ x''_{1f} &= -\kappa\sin\varphi_1 - L_1\kappa'\sin\varphi_1 - L_1\kappa^2\cos\varphi_1 & y''_{1f} &= \kappa\cos\varphi_1 + L_1\kappa'\cos\varphi_1 - L_1\kappa^2\sin\varphi_1. \end{aligned} \tag{3.6}$$

With the movement of the kingpin joint around the (x, y)-plane, we can write virtual steering angle θ of the trailer, as well as the change in angle θ', as

$$\begin{aligned} \theta &= \operatorname{atan2}(y'_{1f}, x'_{1f}) - \varphi_1 \\ \theta' &= \frac{x'_{1f}y''_{1f} - x''_{1f}y'_{1f}}{x'^2_{1f} + y'^2_{1f}} - \kappa \\ \theta' &= \frac{L_1\kappa'}{1 + L_1^2\kappa^2}. \end{aligned} \tag{3.7}$$

The last result is obtained by substituting (3.6) into (3.7) and rewriting the resulting expression. Using the dynamics of $\varphi'_1(s)$ in (3.4), we have an alternative and simpler way of deriving θ and θ' which does not require calculations on the kingpin joint, namely

$$\begin{aligned} \kappa &= \frac{1}{L_1}\tan\theta \\ \theta &= \tan^{-1}(L_1\kappa) \\ \theta' &= \frac{L_1\kappa'}{1 + L_1^2\kappa^2}. \end{aligned} \tag{3.8}$$

Geometrically, we find the relationship $\theta = \beta + \varphi_0 - \varphi_1$, from which we can derive a dynamical expression for $\beta(s)$, starting by differentiation and substituting φ'_0-dynamics, as in

$$\begin{aligned} \theta' &= \beta' + \varphi'_0 - \varphi'_1 \\ \beta' &= -\frac{1}{L_{0b}}\sqrt{1 + L_1^2\kappa^2}\sin\beta + \theta' + \kappa. \end{aligned} \tag{3.9}$$

The resulting expression can be interpreted as a stable autonomous part, disturbed by the inputs θ' and κ, which are a direct result of required curvature at the trailer rear. What we have achieved is that any continuously differentiable curvature function $\kappa(s)$, together with the initial state $(x_1(0), y_1(0), \varphi_1(0), \varphi_0(0))$, fully determines the state of the entire vehicle as a function of distance. Moreover, we have obtained the dynamical relationship for the virtual steering input β as a function of the states of the vehicle. The latter can be used for the objective function and constraints on the input.

Continuity requirements

With the strategy of splitting the path into different segments, a risk of discontinuities at the breakpoints has been created. If each segment is to be optimised separately, alignment on all system states is required at breakpoints, as well as $\mathcal{C}^1$ continuity for the input δ, as by Definition 2.1.2. These constraints can be simplified to be expressed closer to the optimisation function: curvature κ, further simplifying the optimisation problem.

Starting off, the geometrical relationship between actual and virtual steering angles of (2.3), shows that continuity of β implies continuity at the actual input. Consequently, by means of $\theta = \beta + \varphi_0 - \varphi_1$, we can replace the continuity requirement on φ_0 by one on θ, since we had previously assumed continuous β and φ_1. Furthermore, (3.9) indicates that for equal κ, β and θ', an equal value for β' is obtained. Continuing, (3.8) allows the requirement on θ' to be replaced by one on κ', since we have already assumed alignment of κ. Graphically, Figure 3.4 shows the relationships between the six variables on top, on which continuity is in principle required, and a sufficient set of variables for which continuity implies that the original state and input are continuous. The highlighted variables are the furthest simplification towards input constraints on the optimisation function that the proposed description allows. This simplifies the resulting optimisation problem as not all system states and input δ need to be reconstructed during the computation.

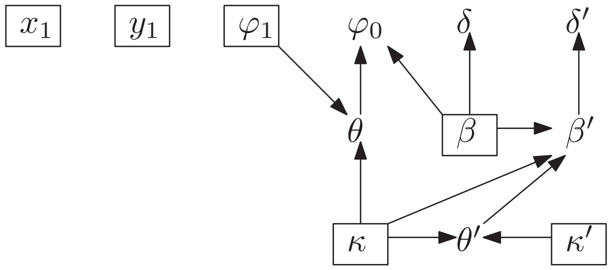


Figure 3.4: Relationship tree for continuity requirements.

In short, we have six variables on which we require alignment on a switching point between different path segments, as well as throughout each segment, namely x_1, y_1, φ_1, κ, κ' and β. Notice that five of these six constraints act on the trailer axle and one describing the tractor side slip at the kingpin joint. Furthermore, three of them have been expressed explicitly in the optimisation function κ, namely φ_1, being defined as the integral of κ, κ itself and its derivative κ', simplifying the kinematics that have to be evaluated during optimisation. The remaining issue is still that the final expression in (3.9) is an ODE, meaning that it has to be solved online and that it is dependent on the initial value $\beta(0)$.

The observation that solving an ODE is required to obtain the required input of the truck to follow the path parametrised by $\kappa(s)$, means that the proposed convention with the trailer axle as control point is not feedback linearisable. Feedback linearisability entails the presence of a finite set of outputs whose derivatives fully determine the states and inputs, without the need for an integration procedure [42]. Note that κ does not suffice as it, together with its derivatives, is unable to uniquely identify β. Conversely, input curvature κ and its derivatives can not be used to fully constrain the movement of the entire truck. In literature, it has been shown that the truck semi-trailer combination can be made feedback linearisable by proper choice of outputs [42]. However, the required relations to reconstruct the system states based on this output function are too complicated to include in the optimisation problem.

Optimisation problem

With the strategy and dynamics of the tractor semi-trailer combination introduced, all building blocks of the optimisation problem are present. Integration of the dynamics of (3.4), together with (3.8) and (3.9), yields

$$\begin{aligned}
x_1(s) &= x_1(0) + \int_0^s \cos\varphi_1(s)\mathrm{d}s \\
y_1(s) &= y_1(0) + \int_0^s \sin\varphi_1(s)\mathrm{d}s \\
\varphi_1(s) &= \varphi(0) + \int_0^s \kappa(s)\mathrm{d}s \\
\beta(s) &= \beta(0) + \int_0^s \left[-\frac{1}{L_{0b}}\sqrt{1+L_1^2\kappa(s)^2}\sin\beta(s) + \frac{L_1\kappa'(s)}{1+L_1^2\kappa(s)^2} + \kappa(s)\right]\mathrm{d}s
\end{aligned} \tag{3.10}$$

where without loss of generality we assume $s(0) = 0$. The functions above are the basis of the kinematics reconstruction used in the optimisation procedure. Note that by the derivation of Figure 3.4, the tuple $q(s) = (x_1(s), y_1(s), \varphi_1(s), \beta(s))$ prescribes all points on the truck along the entire reference path. As a result, the procedure is to optimise the curvature function $\kappa(s)$ between the initial pose $q(0) = (x_1(0), y_1(0), \varphi_1(0), \beta(0))$ and the final pose $q(s_f) = (x(s_f), y(s_f), \varphi(s_f), \beta(s_f))$, where $s \in [0,\ s_f]$. Note that the total path distance s_f is an unknown variable and therefore included in the optimisation. The path planning optimisation problem for a single segment is then defined by

$$\begin{aligned}
\textbf{Find} \quad & \kappa(s), s_f \\
\textbf{to minimise} \quad & J(s_f, \kappa(s), \kappa'(s)) \\
\textbf{s.t.} \quad & (3.10) \\
& q(s) \in \mathcal{Q}_{\text{free}} \\
& q(0) = q_0 \qquad\qquad q(s_f) = q_f \\
& \kappa(0) = \kappa_0 \qquad\qquad \kappa(s_f) = \kappa_f \\
& \left.\frac{\mathrm{d}\kappa}{\mathrm{d}s}\right|_0 = \kappa_0' \qquad\qquad \left.\frac{\mathrm{d}\kappa}{\mathrm{d}s}\right|_{s_f} = \kappa_f' \\
& |\beta(s)| \le \beta_{\max} \qquad\qquad \left|\frac{\mathrm{d}\beta}{\mathrm{d}s}\right| \le \beta'_{\max} \\
& 0 \le s_f \le s_{\max} \\
& \forall\ s \in [0,\ s_f],
\end{aligned} \tag{3.11}$$

where $J(s_f, \kappa(s), \kappa'(s))$ is the cost function, q_0 and q_f are the initial and final desired poses determined by the breakpoints, $\mathcal{Q}_{\text{free}}$ the obstacle-free configuration space, κ_0, κ_f, κ_0' and κ_f' the initial and final curvatures and curvature derivatives, respectively. These, too, are defined by the state of the vehicle at the breakpoints, where the curvature is assumed to be equal to zero. Moreover, $\beta_{\max}$ and $\beta'_{\max}$ are the maximum virtual steering angle and the maximum steering speed. Finally, $s_{\max}$ is the maximum admissible distance for the to be optimised path.

The function to be minimised, $J(s_f, \kappa(s), \kappa'(s))$, is a penalty function including path length as well as the 2-norm of the curvature and its derivative, as in

$$J(s_f, \kappa(s), \kappa'(s)) = \underbrace{\alpha_1 s_f}_{J_1} + \underbrace{\alpha_2 \int_0^{s_f} \left|\left|\frac{\mathrm{d}\kappa(s)}{\mathrm{d}s}\right|\right|^2 ds}_{J_2} + \underbrace{(1 - \alpha_1 - \alpha_2) \int_0^{s_f} ||\kappa(s)||^2 ds}_{J_3}. \tag{3.12}$$

Here, $0 \le \alpha_1 \le 1$ and $0 \le \alpha_2 \le 1$ are tuning parameters with which the weights on the different penalties can be changed, satisfying $0 \le \alpha_1 + \alpha_2 \le 1$. This way, the qualitative properties of the path obtained from the optimisation problem (3.11) can be manipulated based on preference, since the solution is always optimal in the sense that $J(s_f, \kappa(s), \kappa'(s))$ is minimised. The purpose of J_1 is straightforward, increasing the gain leads to shorter paths, generally reducing travel time and saving energy. Secondly, the term J_2 affects steering intensity, meaning the rate of change of the steering angle, reducing tyre wear due to slip of the steered wheels, unpredictability and actuator strain. Thirdly, the component J_3 penalises the curvature itself, leading to a smoother path, more room for corrections as the feedforward steering angle will be limited and less tyre wear as the presence of multiple axes induces slip when cornering. It should be noted that the objective function is expressed in terms of curvature at trailer rear $\kappa(s)$, whereas the choice for the steering input $\delta(s)$ or virtual steering input $\beta(s)$ would have been more intuitive. Equation (3.9), however, shows that the primary contributor to changes in the steering angle is $\theta'(s)$, which is expressed in terms of curvature $\kappa(s)$ and curvature change $\kappa'(s)$. Using $\beta(s)$ itself is numerically unattractive, as it follows from a nonlinear ODE, which is why this simplification towards $\kappa(s)$ is made.

Another addition to the optimisation problem (3.11) is the requirement $q(s) \in \mathcal{Q}_{\text{free}}$, stating that the state of the truck should always be member of the obstacle-free configuration workspace. Define $\mathcal{O}$ as the united set of all obstacles, or $\mathcal{O} = \cup \mathcal{O}_i \quad \forall\ i \in \{1, 2, \ldots, n_{\mathcal{O}}\}$ with $\mathcal{O}_i \in \mathbb{R}^2$ the set of points spanning obstacle i and $n_{\mathcal{O}}$ the number of obstacles. Moreover, consider $\mathcal{A}(q(s)) \in \mathbb{R}^2$ as the set of points covered by the truck, as a function of the tuple $q(s)$, and $\mathcal{Q} \in \mathbb{R}^4$ as the entire configuration workspace. We can then construct the obstacle-free configuration space as

$$\mathcal{Q}_{\text{free}} = \{q(s) \in \mathcal{Q} | \mathcal{A}(q(s)) \cap \mathcal{O} = \emptyset\}. \tag{3.13}$$

Computing $\mathcal{Q}_{\text{free}}$, however, becomes computationally expensive for even moderately complex configuration spaces [18]. For a four-dimensional configuration space, especially for systems with revolute joints, this computation is next to impossible and not applicable to a distribution centre environment. Calculating the space beforehand is not feasible either, as the presence, position and dimension of other trucks around the centre leads to infinitely many options. Therefore, in the next section, two obstacle avoidance methods are introduced that circumvent the construction of an explicit representation of $\mathcal{Q}_{\text{free}}$.

3.3 Obstacle avoidance

To guarantee obstacle avoidance for all generated paths by the optimisation problem (3.11), without explicitly computing $\mathcal{Q}_{\text{free}}$, two methods are introduced and compared. Firstly, a bounding box with obstacle points method, which calculates the minimum distance of the truck along a path to each obstacle. As long as these minimum distances exceed a certain safety margin, a collision-free path can be concluded. Secondly, the Separating Axis Theorem (SAT) is applied to compare polygona l obstacles with a simplified footprint of the truck. Using this more involved method, the shortest distance is calculated as well, which again has to exceed some threshold. In this section, the working principles of both methods are introduced and compared on feasibility, computational requirements and conservatism.

Bounding box with obstacle points

The first proposed method is a very straightforward way to deal with obstacles. At its core, it only considers distances between points and lines, which are very easy and cheap to compute. Around distribution centres, encountered obstacles tend to be polygonal or spherical, and most of the times convex. As a result, by using lines and points, we can capture a large number of different obstacle configurations around a given planning task. The proposed method involves creating a convex bounding box around the current segment, within which the to be optimised path has to be placed. Provided that the truck stays a certain safety distance $d_{\mathcal{O}}$ away from the boundaries of the box, and no obstacles are present within, it can be concluded that the segment contained in that box is collision-free. In case there are obstacles within the box, they are defined as a finite number of points, to which a minimum distance $d_{\mathcal{O}}$ has to be kept at all times.

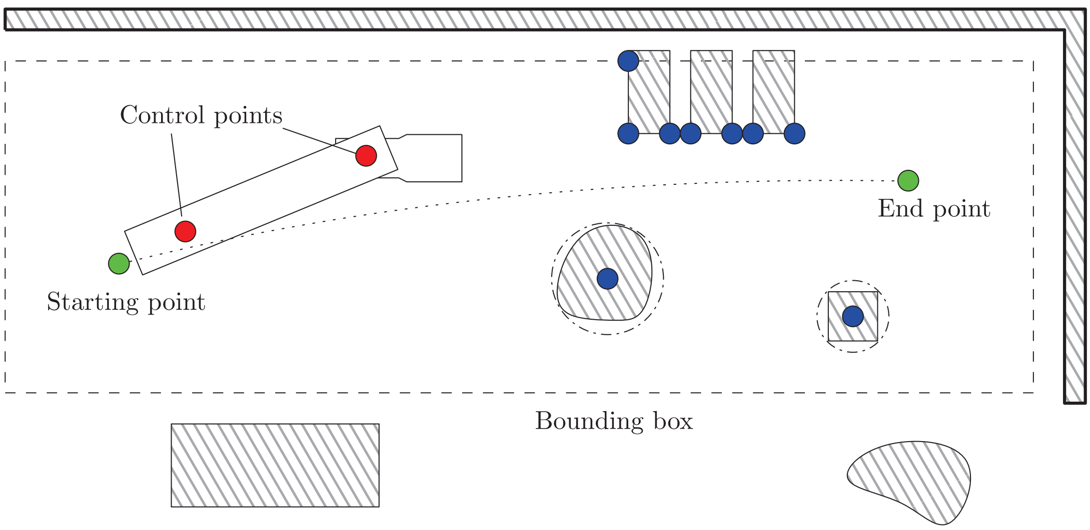


Figure 3.5: Bounding box with obstacles points visualisation.

Consider Figure 3.5, where the bounding box, indicated by the dashed line, has been applied to aid the obstacle avoidance approach. First of all, it has been selected such that it encloses the start and end points of the segment, both indicated in green. Secondly, it excludes a number of obstacles which can be disregarded in the planning optimisation as the path is not allowed to leave the box in the first place. This only leaves the obstacles within the bounding box as possible collisions of the path, for which the second part of the procedure is required. Here, we place obstacle points o_i at vertices or in the centre of approximately spherical objects, as indicated in blue. The top row of blue obstacle points is positioned at the vertices of the rectangular objects placed there and do not have an inherent radius. Below the dotted path, there are two more obstacle points, which are placed in the centres of obstacles. These encompass the entire object by introducing an obstacle radius $r_{\mathcal{O}_i}$ to them. Finally, we require the path of the truck to remain a certain safety distance $d_{\mathcal{O}}$ away from all obstacles, based on the control points and dimensions of the truck. Consider the control points at the trailer axle $p_1(s) = (x_1(s), y_1(s))$ and at the kingpin joint $p_{1f}(s) = (x_{1f}(s), y_{1f}(s))$. Additionally, consider the obstacle points $o_i = (x_{\mathcal{O}_i}, y_{\mathcal{O}_i})$ for all $i \in \{1, 2, \ldots, n_l\}$, with n_l the number of obstacle points. We then attempt to find conditions for obstacle avoidance for all travelled distance s. The lines spanning the bounding box $B(x, y)$ are defined as

$$b_j(x, y) = b_{jx}x + b_{jy}y + b_{jc}, \tag{3.14}$$

for all $j \in \{1, 2, \ldots, n_b\}$ with n_b the number of lines spanning the bounding box. For the case of Figure 3.5, this number $n_b = 4$. Define the minimum distance constraint between the truck and any edge of the bounding box as

$$\begin{aligned} \rho(p_1(s), b_j(x, y)) &= \min_{s} \frac{|b_{jx}x_1(s) + b_{jy}y_1(s) + b_{jc}|}{\sqrt{b_{jx}^2 + b_{jy}^2}} \geq d_{\mathcal{O}} \\ \rho(p_{1f}(s), b_j(x, y)) &= \min_{j,s} \frac{|b_{jx}x_{1f}(s) + b_{jy}y_{1f}(s) + b_{jc}|}{\sqrt{b_{jx}^2 + b_{jy}^2}} \geq d_{\mathcal{O}}, \end{aligned} \tag{3.15}$$

where $\rho(\cdot)$ indicates the Euclidean distance function between two objects, $s \in [0,\ s_f]$ and $j \in \{1, 2, \ldots, n_b\}$. In other words, we have two control points that are checked against n_b lines spanning the bounding box, yielding $2n_b$ inequality constraints. The minimum distance of all lines, along the entire segment s, has to be lower bounded by the safety distance $d_{\mathcal{O}}$, accounting for the dimensions of the truck. If these conditions are met, it is safe to say that the truck remains within the bounding box along the entire segment, excluding all obstacles outside of it.

Regarding the obstacles points, a similar approach is taken. The minimum distance between obstacle points and the control points on the truck is given by

$$\begin{aligned} \rho(p_1(s), o_k) &= \min_{s} ||p_1(s) - o_k||_2 - r_{\mathcal{O}_k} \geq d_{\mathcal{O}} \\ \rho(p_{1f}(s), o_k) &= \min_{s} ||p_{1f}(s) - o_k||_2 - r_{\mathcal{O}_k} \geq d_{\mathcal{O}}, \end{aligned} \tag{3.16}$$

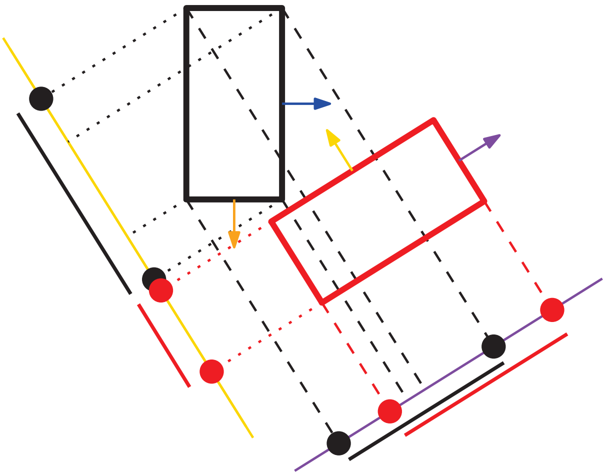
where we notice the additional distance $r_{\mathcal{O}_k}$ that has to be covered to clear an obstacle of said radius. The above inequality constraints have to be sufficed for all obstacle points k, as a result of which no collision between the path and obstacles can be concluded. This adds an additional $2n_k$ inequality constraints to the optimisation problem.

It should be noted that only comparing the rear trailer and kingpin joint to the surrounding map can still lead to collisions, as the truck could also hit obstacles between these points. Obviously, the distance $d_{\mathcal{O}}$ can be chosen high enough that all points on the truck are considered. This comes at the cost of conservatism, as a truck is not accurately approximated by two circles. As a result, the value of $d_{\mathcal{O}}$ has to be chosen such that it balances collision possibilities and conservatism, depending on the dimensions of the truck.

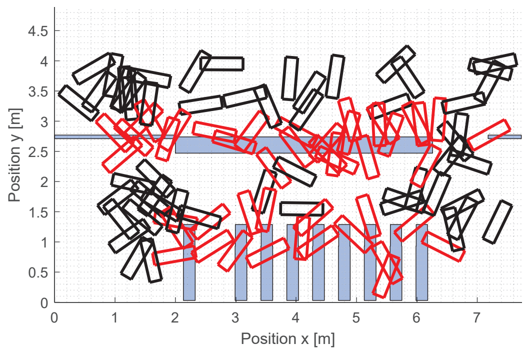
Separating Axis Theorem

From the observation that a truck is poorly approximated by circles, it follows as a logical next step to look for better simple shape approximations of the truck footprint $\mathcal{A}(q(s))$. Looking at Figure 1.2, a possible approximation would be to take one rectangle for the trailer, and one for the tractor. By adjusting the width of the tractor rectangle slightly, protruding elements such as steered wheels and rear-view mirrors can be included without introducing much conservatism at all. In addition to the truck, all obstacles expected around a distribution centre can be approximated by (unions of) convex polygons. This reduces the collision avoidance task to the problem of preventing intersections between the two rectangles describing the truck and the convex polygons defining the obstacles.

The problem of fast collision identification between convex polygons is an often encountered problem, which occurs for example in geometry, constrained optimisation problems and video game design [43]. One common solution is the hyperplane separation theorem 2.1.5 [36]. This theorem states that if there exists a hyperplane that separates two convex objects, they do not intersect. In two-dimensional studies, this theorem is often referred to as the separating axis theorem, stating that an axis is sought along which there is a gap between the projections of two convex objects upon that axis [36]. If such an axis is found, the search can be stopped and it can be concluded that no collision occurs. It also states that checking all axes normal to the edges of the objects suffices to conclude that collision does occur. Figure 3.6a shows an example where two identical rectangles are placed such that only one axis allows the conclusion that no collisions occur, namely the yellow axis of the right obstacle. If both objects are projected upon that axis, there is a gap to be observed in between, meaning that both objects are at least that distance apart. One of the axes along which no gap exists is shown in purple, where the ranges between the extreme projections of the red and black obstacles overlap.



(a) Separating axis theorem example along two axes, indicated in purple and yellow. The projections of the red and black obstacles are indicated by the coloured dots on the axes.



(b) Random collision check for 100 rectangles on the TruckLab scenario. Red rectangles indicated collisions with the environment while black ones are collision-free.

Figure 3.6: Separating axis theorem examples.

As mentioned above, the SAT works by looping over all relevant axes of the obstacles and the truck. The choice to approximate the truck by two rectangles leads to only four axes having to be checked coming from the truck orientation, namely $\vec{e}_x^{\,0}$, $\vec{e}_y^{\,0}$, $\vec{e}_x^{\,1}$ and $\vec{e}_y^{\,1}$. The obstacles, however, could in general have arbitrary many axes along which they are oriented. In distribution centre environments, the number of obstacles itself tends to be limited, as well as the orientations at which they are placed. Buildings, roads, fences and parked trucks are often positioned perpendicularly, or with a constant angle with respect to each other. As a result, the total amount of axes along which collisions have to be checked is limited as well. Consider for example the scenario shown in Figure 3.5, where most normals to obstacles are along the horizontal and vertical directions. The key strength of this algorithm is that, whenever the truck does not collide with its environment, one evaluated axis is enough to make that conclusion. Moreover, due to the narrow shape of a truck, the lateral direction of the trailer, $\vec{e}_y^{\,1}$, most often directly proves no collision, making it the most efficient starting axis.

The projections are calculated by taking the inner product between the normal vectors to each edge and the vertices of both objects currently under consideration. When this is done, the maximum and minimum projections of one object along the considered axis are evaluated and compared to those of the other object. If, in the case of the purple axis in Figure 3.6, the ranges between maximum and minimum projections overlap, we proceed to another axis. Only when all axes show overlapping behaviour, certainty of collision can be concluded. Once another axis, in this case the yellow one, shows a gap between the extreme projections of both obstacles, the search is stopped and a gap distance is concluded. Note that if the search is stopped after finding a gap, a larger gap might exist along a yet unexamined axis. The above procedure is described in pseudo-code in Algorithm 1. Figure 3.6b shows the results of 100 randomly oriented rectangles around the TruckLab scenario, with the red colour indicating collisions with the environment.

```
Data: Vertices of all obstacles
for all obstacles including truck do
    Compute normals to edges;
    Normalise identified normals;
    /* Normalisation is not required for the result of the algorithm, but assumed
       because it conserves the dimensions of the obstacle onto the projection.   */
end
Omit duplicate normal directions;
for all unique normals, starting at e_y^1 do
    for all obstacles including truck do
        Calculate inner products of current obstacle with current normal;
        Identify minimum and maximum projections;
    end
    Compute projection ranges between minimum and maximum projection;
    Compare projections ranges of truck to those of the obstacles;
    if no overlap then
        Conclude no collision;
        return the gap distance between the projection ranges (G_d > 0);
    end
end
Conclude collision;
return The gap distance between the projection ranges (G_d < 0);
```

Algorithm 1: Separating Axis Theorem algorithm.

Recalling from (3.11) that we require $q(s) \in \mathcal{Q}_{\text{free}}\ \forall s \in [0,\ s_f]$, indicating that infinitely many truck poses $q(s)$ have to be checked to conclude that the entire path segment is collision-free. As a result, a procedure is required to minimise the number of required tests, while still maintaining the no-collision guarantee. Consider the gap distance found by Algorithm 1, which provides an indication of a safe driving distance $\delta s(s)$. As mentioned before, the maximum gap distance along all axes is in principle unknown, but could always be computed if needed. Using the computed minimum distance between the truck and nearest obstacle, a safe driving distance is sought, meaning that no collision checks are required on the interval between s and $s + \delta s$. Then, by starting at the initial condition, which is known to be collision-free by virtue of the high-level path, the path can incrementally be checked for collisions. Finding the maximum safe driving distance based on a gap distance is no simple task, however. Suppose the truck was described by a simple point mass model, in which case the minimum safe driving distance would be the largest observed gap distance. In case the track was assumed to be straight, the same conclusion applies. However, for the truck, this computation is much more involved. Here, we consider the truck as the union of two rectangular blocks, placed along an arbitrary reference path, as shown in Figure 3.7. A very conservative but guaranteed safe driving distance would follow as

$$\delta s(s) = G_d(s) - (\text{circumscribed circle} - \text{inscribed circle}), \tag{3.17}$$

with the circumscribed and inscribed circle shown in Figure 3.7. The difference in radius of these circles relates to the symmetry of the truck with respect to the control point at the trailer rear, which the trailer and tractor combination is clearly lacking. Due to the large length to width ratio of the truck, especially with respect to the trailer rear control point, this approach often yields negative results, meaning that it can not guarantee any safe driving distance at all, due to large variations in the curvature possibly being present. In the proposed method, however, an approximation based on the assumption that the path is relatively straight for small driving distances is made, allowing

$$\delta s(s) = \min(\delta s_{\max}, k_{\text{SAT}} \cdot G_d(s)), \tag{3.18}$$

with a maximum driving distance $\delta s_{\max}$ to satisfy the assumption that the path remains relatively straight and the safety factor $k_{\text{SAT}} \leq 1$ to account for possible curvature changes. The safety factor represents a trade-off between computational demands and the risk of collisions in extreme cases. Using the latter option for $\delta s(s)$, the pose of the truck can be evaluated at a finite number of positions along the possible path segment, allowing the optimisation problem to avoid paths containing collisions.

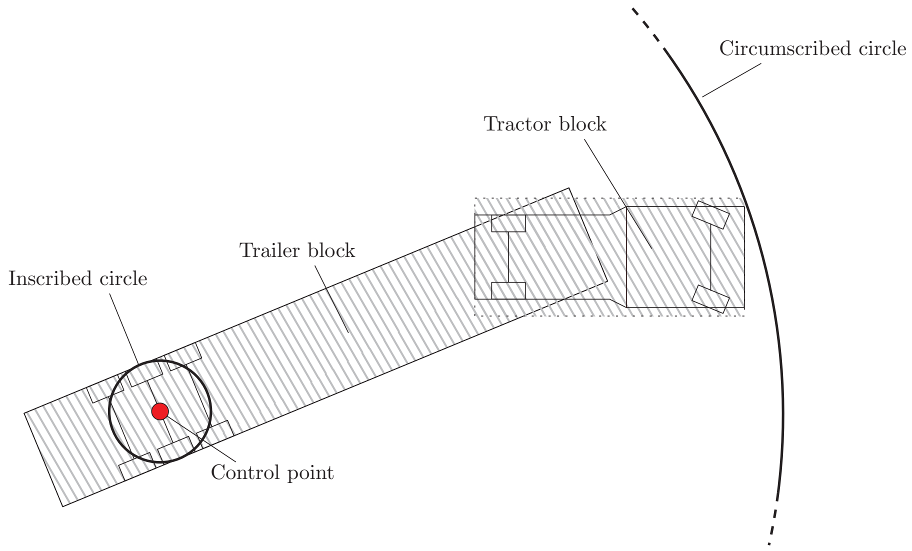


Figure 3.7: Convex area approximation and inscribed and circumscribed circles of the truck.

In conclusion, the Separating Axis Theorem is a much more robust and less conservative method to identify collisions along reference paths than the bounding box method. It efficiently and accurately approximates the shapes of both the truck and of the surroundings and determines whether or not collisions occur swiftly. The smallest distance to a collision of the currently evaluated path is then returned to the optimisation problem, which has to be positive.

3.4 Solving the optimisation problem

Reconsidering the optimisation problem introduced in (3.11), it is clear that the entire path for a given initial and final pose is determined by the curvature function $\kappa(s)$. In this section, the approaches towards determining the optimal curvature function will be detailed. Firstly, a parametrisation of the curvature is proposed, based on splines expressed at curve knots. Secondly, for these curve knots, suitable initial values are sought to speed up the optimisation process and avoid local minima. Finally, the solver is introduced with which the nonlinear optimisation problem is solved and possibilities for speeding up the optimisation are introduced.

Spline approximation

Solving an optimisation problem over a function space is in principle an infinite-dimensional problem. Instead, we seek to approximate the function $\kappa(s)$ over a single segment by a finite number of coefficients and basis functions, requiring only the coefficients to be optimised. This solution leads to the field of polynomial approximation, where arbitrary resolution can be obtained by increasing the polynomial order. In practice, splines, which are relatively low order polynomials connected at their ends, are often preferred over higher order polynomials due to numerical considerations. The continuity requirements analysis of section 3.2 showed that constraints on $\kappa(s)$, $\kappa'(s)$ and $\beta(s)$ are required at the starts and ends of each segment. Due to $\beta(0)$ relying on $\kappa(0)$ and $\kappa'(0)$ by virtue of (3.9) and (3.8), only five constraints remain. Preventing the optimisation problem from being overconstrained, we need to introduce at least as many degrees of freedom as constraints, such that all constraints can be respected. As a result, the lowest order spline that can be applied in this method is a quintic spline space, described by

$$\kappa_i(s) = \alpha_{0,i} + \alpha_{1,i}(s - s_i) + \alpha_{2,i}(s - s_i)^2 + \alpha_{3,i}(s - s_i)^3 + \alpha_{4,i}(s - s_i)^4 + \alpha_{5,i}(s - s_i)^5 \qquad s \in [s_i, \quad s_i + s_{fi}], \tag{3.19}$$

with $\alpha_{j,i}$ being the j-th coefficient of the i-th spline and s_i the starting distance of the i-th spline. Note that this notation assumes n_s spline functions per path segment, which is a tradeoff between accuracy and numerical complexity. The choice of s_i at the start of each spline entails that $s = s_1 = 0$ at the first spline of each segment. Recall that all segments are optimised over separately, each containing n_s spline functions. At the start of spline function κ_i, the s coordinate is equal to s_i, as in Figure 3.8. Furthermore, the derivative of the curvature function (3.19) with respect to s is obtained from simple differentiation and is given by

$$\begin{aligned}\kappa_i'(s) = \alpha_{1,i} + 2\alpha_{2,i}(s-s_i) + 3\alpha_{3,i}(s-s_i)^2 + 4\alpha_{4,i}(s-s_i)^3 + 5\alpha_{5,i}(s-s_i)^4 \\ s \in [s_i, \quad s_i + s_{fi}].\end{aligned} \tag{3.20}$$

These curvature functions are fully defined by their parameters, which are collected in the coefficient vector as

$$p_i = \begin{bmatrix}\alpha_{0,i} & \alpha_{1,i} & \alpha_{2,i} & \alpha_{3,i} & \alpha_{4,i} & \alpha_{5,i} & s_{fi}\end{bmatrix}^T \in \mathbb{R}^7, \tag{3.21}$$

where we have six spline coefficients $\alpha_{j,i}$ and the spline length s_{fi} as optimisation variables.

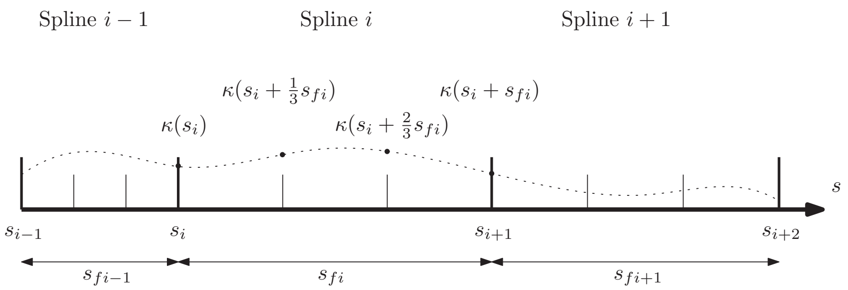


Figure 3.8: Schematic representation of parametrisation of spline function i.

The nonlinear optimisation problem for a single segment then boils down to finding the optimal value p_i^* for each of the n_s spline functions contained in the optimisation problem (3.11) for the associated segment. From this optimisation vector and the corresponding curvature function, all references on all vehicle states can be inferred. Even though multiple options are mathematically valid for inferring these references, for instance driving forward and backwards or angle changes of π, it is trivial to consider only desired paths where no internal collisions occur.

Curve knot spacing

The proposed parametrisation has the significant drawback that its coefficients $\alpha_{j,i}$ do not have a physical interpretation. This entails that it is not trivial to assign constraints to them, nor to make sensible estimates of their values to support optimisation. Therefore, we seek to reparametrise the coefficient vector into a more intuitive form, to which constraints can directly be applied. Consider the reparametrisation of p_i into

$$\begin{aligned}p_i^* &= \begin{bmatrix}\kappa(s_i) & \kappa'(s_i) & \kappa\left(s_i + \frac{1}{3}s_{fi}\right) & \kappa\left(s_i + \frac{2}{3}s_{fi}\right) & \kappa\left(s_i + s_{fi}\right) & \kappa'\left(s_i + s_{fi}\right) & s_{fi}\end{bmatrix}^T \\ &= T(s_{fi})p_i,\end{aligned} \tag{3.22}$$

requiring the transformation matrix

$$T(s_{fi}) = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 1 & \frac{1}{3}s_{fi} & \left(\frac{1}{3}s_{fi}\right)^2 & \left(\frac{1}{3}s_{fi}\right)^3 & \left(\frac{1}{3}s_{fi}\right)^4 & \left(\frac{1}{3}s_{fi}\right)^5 & 0\\ 1 & \frac{2}{3}s_{fi} & \left(\frac{2}{3}s_{fi}\right)^2 & \left(\frac{2}{3}s_{fi}\right)^3 & \left(\frac{2}{3}s_{fi}\right)^4 & \left(\frac{2}{3}s_{fi}\right)^5 & 0\\ 1 & s_{fi} & s_{fi}^2 & s_{fi}^3 & s_{fi}^4 & s_{fi}^5 & 0\\ 0 & 1 & 2s_{fi} & 3s_{fi}^2 & 4s_{fi}^3 & 5s_{fi}^4 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}. \tag{3.23}$$

In the new vector p_i^*, the curvature and derivative of curvature at the beginning and the end of each spline can be recognised. Since these directly appear in the optimisation problem formulated in (3.11), this allows for the direct application of constraints on the parameter vector. Moreover, it contains the curvature at two intermediate points along the spline function, which is a much more intuitive way to approach a path than polynomial coefficients. In essence, the transformation $T(s_{fi})$ scales the coefficients in p_i, which have very different units and magnitudes, into physical properties of the path in p_i^*. Note that this transformation is a nonlinear transformation between the vectors p_i and p_i^*, as the variable s_{fi} is a priori unknown and in itself an optimisation variable. The knots at which the curvature is now directly prescribed are not to be confused with the knots used in B-splines, which serve a completely different purpose [44].

As previously mentioned, this nonlinear coordinate transformation in the optimisation variables allows the application of linear constraints into the new variables p_i^*. From (3.11), we recall that $\kappa(0) = \kappa_0$, $\kappa(s_f) = \kappa_f$, $\kappa'(0) = \kappa'_0$ and $\kappa'(s_f) = \kappa'_f$, which in terms of the new parameter vector becomes

$$\begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1^* \\ p_{n_s}^* \end{bmatrix} = \begin{bmatrix} \kappa_0 \\ \kappa'_0 \\ \kappa_f \\ \kappa'_f \end{bmatrix}. \tag{3.24}$$

In addition to constraints on the segment ends, we also require the splines to be interconnected smoothly, as discussed in section 3.2. Here, it was concluded that continuity on all states is guaranteed by continuity on $x_1(s)$, $y_1(s)$, $\varphi_1(s)$, $\kappa(s)$, $\kappa'(s)$ and $\beta(s)$. Recalling (3.10), we recognise that only $\kappa(s)$ and $\kappa'(s)$ are not enforced through their initial conditions. Therefore, we require defining them in terms of the optimisation vectors p_i^* and p_{i+1}^*, according to

$$\begin{gathered} \kappa_i(s_{fi} + s_i) = \kappa_{i+1}(s_{i+1}) \\ \kappa'_i(s_{fi} + s_i) = \kappa'_{i+1}(s_{i+1}) \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} p_i^* = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} p_{i+1}^* \qquad \forall\ i \in \{1, 2, \dots, n_s\}. \end{gathered} \tag{3.25}$$

In addition to these linear equality constraints on the curvature functions, there exists also an inequality constraint on the total path length. This constraint on the path length is simply given by

$$0 \leq s_f = \sum_{i=1}^{n_s} s_{fi} \leq s_{\max}, \tag{3.26}$$

which is the sum of all spline function lengths.

The proposed parametrisation of spline function i, is schematically visualised in Figure 3.8. Here, the intermediate distances between the different spline functions s_{fi} are clearly visible, as well as the curve knots at which the curvature is prescribed. Note that any distribution of the intermediate curve knots is mathematically equivalent, while the choice for an equidistant distribution is most intuitive. The application of the constraints is also clearly visible, namely alignment on curvature and its derivative at all spline function ends s_i. This way, a twice differentiable curvature spline is obtained, which means that all system states are also continuous, on the basis of section 3.2.

Initial estimates

Properly initialising a nonlinear optimisation problem such as (3.11) can greatly benefit computational speed and convergence properties. Intuitively, if the optimisation problem starts in a small neighbourhood around the global optimum, it should be able to converge to said optimum swiftly. Obviously, the global optimum is unknown before the optimisation problem, but the information incorporated in the HL-path is available. Of the HL-path, it is known that it contains the optimal solution for a particular discretisation size for a path travelled by a bicycle model. The primary task of the low-level planner is to infer continuous states of the entire tractor semi-trailer model along the entire path and ensure that it satisfies state and input constraints. To this end, it requires a continuous curvature function, twice differentiable along the entire path and constraints on the entire tractor semi-trailer state. Regardless, the optimal path of $(x_1(s), y_1(s))$ will generally be close for both planners, meaning that more information than just breakpoints can be inferred from the HL-path. In section 3.1, mention was made of obtaining path length and curvature approximations, which will both be dealt with in this section.

Starting with the length of each segment, and consequently of each spline function, we simply measure the length of the HL-path between each of the breakpoints. This yields, for each segment, the value $s_f = \sum_{i=1}^{n_s} s_{fi}$, with n_s the number of spline functions within that segment. As an initial guess, we then encode

$$s_{fi} = \frac{s_f}{n_s} \quad \forall\, i \in \{1, 2, \dots, n_s\}, \tag{3.27}$$

making all spline functions equal in length from the start. Since the choice has been made to optimise over each segment independently, we ensure that they align on a common value. This value has been chosen equal to zero, which corresponds to the choice of breakpoints at the ends of straights, where the curvature is indeed close to zero. A small downside of this choice is that some freedom is lost in the optimisation process since the HL-path fully prescribes the pose at the breakpoints. The choice of alignment at zero curvature at the end of straights implies that κ_0, κ'_0, κ_f and κ'_f of (3.24) are equal to zero, as well as the corresponding initial guesses of p_1^* and $p_{n_s}^*$. Recalling Figure 3.8 and (3.22), we still require the value of the curvature $\kappa(s)$ at the two remaining intermediate positions. Moreover, for p_2^* up to $p_{n_s-1}^*$, we required four guesses of the curvature $\kappa(s)$ and two of the derivative of curvature $\kappa(s)$ at the spline function ends. Due to the structure of the curve knot spacing technique, these can all be sampled based on s_{fi} and s_i. Instead of sampling directly from the HL-path at a single point, the average of the surrounding path is taken in such a way that contributions of the entire HL-path are included. For segment i, depicted in Figure 3.8, the initial estimate of $\kappa(s_i + \frac{1}{3}s_{fi})$ is taken as the arithmetic mean of

$$\kappa(s) \text{ for } s \in \left[s_i + \tfrac{1}{6}s_{fi} \quad s_i + \tfrac{1}{2}s_{fi}\right]. \tag{3.28}$$

Applying the method above yields a decent estimate of the expected optimisation parameters. Consider as an example the TruckLab scenario, consisting of the three segments shown in Figure 3.3. In order to balance the design freedom against the computational complexity of increasing the number of spline coefficients, it has been found that using two connected splines within each segment provides good planning results. This yields a total of fourteen coefficients that, provided known initial conditions, fully define the path along each segment. Next, we approximately infer the length of each segment and the curvature at each curve knot as mentioned, the result of which is listed in the initialisation column in Table 3.1. The first segment is fully straight, meaning that the only nonzero entries of p_i^* are the lengths, which are simply defined according to (3.27). For segments two, however, the sharp curvature to the right, signed negative, is clearly visible. Moreover, we recognise the constraints defined in (3.24) and (3.25).

In addition to examining the numerical values of the initialisation process, the resulting path can be plotted against the HL-path, as done in Figure 3.9. Here, it is clear that the initial path contains the most important characteristics of the optimal final path. It is not yet necessarily feasible, which is most apparent at the end of the second segment, where a large jump is observed towards the start of segment 3. This is a result of a slightly conservative approach to the curvature estimation, meaning that the optimisation process always approaches the global optimum from the side corresponding to the straightest resulting path. However, the initial path does not consider obstacles, meaning that it could contain collisions. Since it is designed to approximate the HL-path, which does take obstacles into account, it gives a clear preference on how to traverse between obstacles.

Nonlinear optimisation problem

Solving the nonlinear optimisation problem proposed in (3.11) is performed using MATLAB R2020a, with the `fmincon.m` function of the MATLAB optimisation toolbox. It allows the inclusion of the objective function, linear and nonlinear constraints and initial search conditions, all of which are used for the path planner. Using an interior point algorithm, the global minimiser to the (multivariate) objective function is sought that still satisfies all constraints. As mentioned before, the segments are not dependent on the results of one another, meaning that they are optimised separately. This allows parallelisation of the computation, greatly speeding up the path planning process.

Table 3.1: Initialisation and optimisation results for TruckLab scenario.

	Initial			**Optimal**		
Segment	1	2	3	1	2	3
$\kappa_1(0)$	0	0	0	0	0	0
$\kappa_1'(0)$	0	0	0	0	0	0
$\kappa_1(\frac{1}{3}s_{f1})$	0	-0.5968	-0.4939	0	-1.0208	-0.6000
$\kappa_1(\frac{2}{3}s_{f1})$	0	-0.9055	-0.7408	0	-0.7913	0.3839
$\kappa_1(s_{f1})$	0	-0.9055	0	0	-0.4300	0.4143
$\kappa_1'(s_{f1})$	0	0	0	0	0.9585	0.1635
s_{f1}	2.0225	2.6250	1.6550	2.0215	4.1000	2.6747
$\kappa_2(0)$	0	-0.9055	0	0	-0.4300	0
$\kappa_2'(0)$	0	0	0	0	0.9585	0
$\kappa_2(\frac{1}{3}s_{f2})$	0	-0.3219	0.2264	0	-0.1188	0.5642
$\kappa_2(\frac{2}{3}s_{f2})$	0	-0.1543	0.9055	0	0.0112	0.9493
$\kappa_2(s_{f2})$	0	0	0	0	0	0.4143
$\kappa_2'(s_{f2})$	0	0	0	0	0	0.1635
s_{f2}	2.0225	2.6250	1.6999	2.0235	1.7111	2.7879

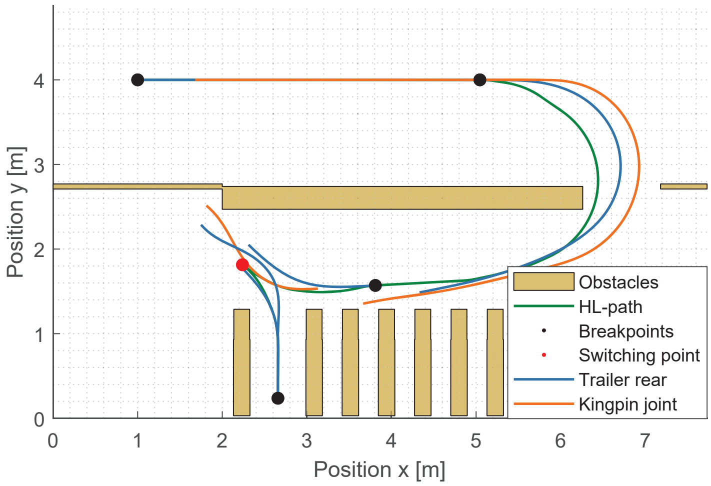


Figure 3.9: Graphical representation of initialised path.

The results of the optimisation for the TruckLab scenario with the scaled trucks of Appendix A are shown in Table 3.1, as well as graphically in Figure 3.10. Here, the bounding box obstacle avoidance method is used, as well as the gains of the objective function $\alpha_1 = 0.001$ and $\alpha_2 = 0.029$. We notice that the resulting path connects smoothly at the segment breakpoints and at the path ends. Moreover, the resulting path forms an intuitive shape overall. The numerical results show that for the first segment, the initial guess is already the global optimum, since this segment is a perfectly straight line. For segment two, the described initialisation procedure correctly estimates the sign and magnitude of the curvature. The optimisation result, however, does introduce a significant length disparity between both spline segments, which also leads to the curvature values having different interpretations, as they are curve knots at differently spaced intervals. The third segment deviates the most from the initial estimates, since it has the most complicated and unpredictable shape. Here, we recognise that both splines are defined in forwards driving direction and meet at the switching point, where we find equal curvature and curvature derivative.

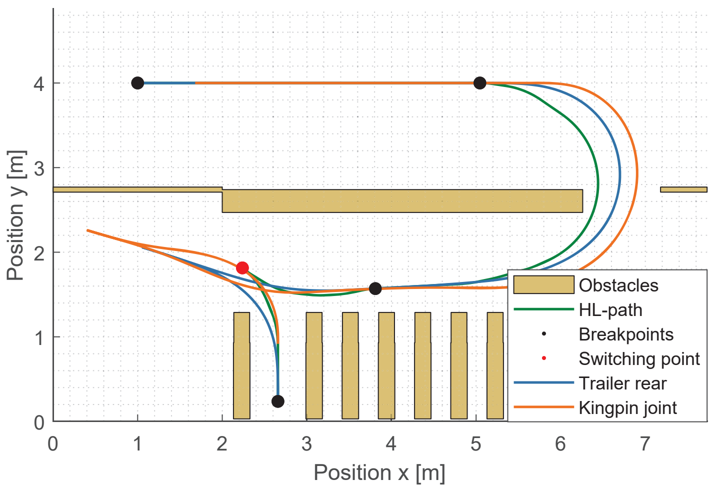


Figure 3.10: TruckLab scenario planned path.

One such reference path is shown in Figure 3.10 and the other benchmark tests in Appendix B. For the considered benchmark tests, the performance of the planner is satisfactory and satisfies all imposed constraints. However, when increasing $d_{\mathcal{O}}$, the planner sometimes fails to find a feasible path when it should still exist. This means that the proposed planner is not complete, primarily due to the limited amount of design freedom resulting from taking only two spline functions per path segment. Regarding computation time, the path shown in Figure 3.10 takes $\approx$ 60 s on an Intel Core i7-4710MQ CPU, preventing online recalculation in case of large tracking errors. The computation time is further increased when using the SAT method, which is numerically expensive to compute at multiple points along the path. These numerical difficulties also show in the robustness in finding a feasible path, which is also decreased for the SAT method, meaning that even more space is required to yield good results.

3.5 Observations and conclusions

In this chapter, the path planning strategy for tractor semi-trailer combinations around warehouses is introduced. It involves a two-stage planning scheme, where a simple and fast high-level planner is used to yield an approximate solution. Based on this path, independent segments are identified for which a nonlinear optimisation problem is solved, seeking the optimal path parametrised by its curvature function. On this function, linear and nonlinear constraints are imposed, ensuring smoothness, state and input constraints and obstacle avoidance. From this curvature function, the full state of the tractor semi-trailer combination is inferred at any point along the path, serving as the reference to the path tracking controller.

The resulting path consists of segments optimal in an arbitrarily weighted objective function for the chosen approximation, satisfies kinodynamic and input constraints, is once continuously differentiable at the input δ and avoids collisions with surrounding obstacles. Besides, due to the transition into distance domain, the planning problem has been made such that the curvature and velocity are optimised over separately. With the proposed search direction initialisation procedure, on the basis of the HL-path, a first estimate lying reasonably close to the optimum is obtained, speeding up the computation. Furthermore, parallelisation allows for an even more efficient implementation, where all segments are evaluated simultaneously.

On the other hand, drawbacks of the proposed method are that it is not complete, meaning that a feasible path may not always be found when it exists. This problem is accentuated when the Separating Axis Theorem method is used for obstacle avoidance, due to the numerical complexities involved. The separating axis method, however, is the only one that provides guarantees that collisions are avoided. For the bounding box method, significant levels of additional safety distance have to be introduced, further shrinking the feasible domain. Additionally, the computation time of the low-level planner prevents online recalculation of the path, so only the predetermined path can be used for the truck to follow. Finally, the obtained path is not necessarily the global optimum, as the breakpoints are determined by the HL-planner. Even though it is optimal in its own condition, it does not take all kinodynamic constraints nor feasible inputs into account.

Chapter 4

Path Tracking Control

The second essential part for implementing autonomous driving of trucks is to obtain a robust and reliable control strategy. With the reference path established and the entire pose of the truck assumed known, the controller is responsible for steering the truck towards the reference and rejecting disturbances. This chapter details the goal of tracking and specifies the considered system in section 4.1. Next, a simplified system is considered to get familiar with the system and nonlinear control in section 4.2. Furthermore, in section 4.3 the controller for the tractor semi-trailer system is introduced and conditions for its stability are proposed. The velocity control problem is tackled in section 4.4. Finally, section 4.5 concludes the chapter, showing simulations, validating the controller and investigating its robustness. The symbols used to describe the state of the vehicle in this chapter refer to the actual measured vehicle state, while the reference is indicated by subscript r, i.e. vehicle state $q(t)$ and reference vehicle state $q_r(t)$.

4.1 Tracking problem definition

In order to synthesise a controller for the tractor semi-trailer system, the control objective must be defined. There are several options to define a tracking problem, being reference tracking, trajectory tracking and path tracking. Since the time component of the reference vehicle has not been established, due to the assumption of stationary obstacles, it serves no clear purpose to define the error dynamics in time domain as $e(t) = q(t) - q_r(t)$. Moreover, tracking a reference defined in time leads to loss of uniformity, as well as practical limitations such as a tendency to cut corners [27] or drive unnecessarily slow. Another option is trajectory tracking, where the coordinates $x_r(t)$ and $y_r(t)$ are known functions of time and serve as the reference point for the ego vehicle. This method also involves both longitudinal and lateral control tasks simultaneously. Instead, we choose path tracking, where we assume that the path is completely independent of time. Placing the vehicle in the vicinity of the path implies some projection onto the reference path, obtaining the reference state $q_r(s_r(s))$ at that position. This full reference state can be used, regardless of the magnitude of the velocity at which the considered vehicle is driving. The additional degree of freedom obtained with this method is the mapping $s_r(q(s(t)))$, or $s_r(s)$, relating any pose of the truck to any distance along the reference. The error coordinates are therefore defined along the lines of $e(s) = q(s) - q(s_r(s))$. This choice ensures that the path is always tracked with the most accuracy possible, meaning that less conservatism has to be included regarding distance to obstacles. Moreover, there is no time component associated with the path, meaning that the considered vehicle should always drive at the optimal velocity with regards to the path, instead of trying to catch up or coast down. Optimality in this sense means maximising the driving speed subject to powertrain, safety and comfort constraints.

System description

The representation of the to be controlled system has been introduced in section 2.2. Consider the tractor semi-trailer dynamics

$$\begin{aligned} \dot{x}_0(t) &= v_0(t)\cos\varphi_0(t) \\ \dot{y}_0(t) &= v_0(t)\sin\varphi_0(t) \\ \dot{\varphi}_0(t) &= \frac{v_{1f}(t)}{L_{0b}}\sin\beta(t) \end{aligned} \tag{4.1}$$

$$\begin{aligned} \dot{x}_1(t) &= v_1(t)\cos\varphi_1(t) \\ \dot{y}_1(t) &= v_1(t)\sin\varphi_1(t) \\ \dot{\varphi}_1(t) &= \frac{v_{1f}(t)}{L_1}\sin\theta(t) = v_1(t)\kappa(t), \end{aligned} \tag{4.2}$$

as well as the location of the kingpin joint

$$\begin{aligned} x_{1f}(t) &= x_0(t) + L_{0b}\cos\varphi_0(t) \\ y_{1f}(t) &= y_0(t) + L_{0b}\sin\varphi_0(t) \end{aligned} \qquad (4.3) \qquad\qquad \begin{aligned} x_{1f}(t) &= x_1(t) + L_1\cos\varphi_1(t) \\ y_{1f}(t) &= y_1(t) + L_1\sin\varphi_1(t). \end{aligned} \qquad (4.4)$$

Differentiating (4.3) and (4.4), substituting the dynamics, multiplying with $\sin\varphi_0(t)$ and $\cos\varphi_0(t)$, respectively, equating both the $x_{1f}(t)$ coordinates and $y_{1f}(t)$ coordinates, subtracting the equation in y from the equation in x, yields

$$L_{0b}\dot{\varphi}_0(t) = -v_1(t)\sin[\varphi_0(t) - \varphi_1(t)] + L_1 v_1(t)\kappa(t)\cos[\varphi_0(t) - \varphi_1(t)]. \tag{4.5}$$

We now have an expression for $\dot{\varphi}_0(t)$ which does not include $\beta(t)$, and is expressed in trailer coordinates and itself only. Defining $\psi(t) = \varphi_0(t) - \varphi_1(t)$ yields the dynamics of the system

$$\begin{aligned} \dot{x}_1(t) &= v_1(t)\cos\varphi_1(t) \\ \dot{y}_1(t) &= v_1(t)\sin\varphi_1(t) \\ \dot{\varphi}_1(t) &= v_1(t)\kappa(t) \\ \dot{\psi}(t) &= \frac{1}{L_{0b}}\left[-v_1(t)\sin\psi(t) + L_1 v_1(t)\kappa(t)\cos\psi(t)\right] - v_1(t)\kappa(t). \end{aligned} \tag{4.6}$$

We once again proceed to split up lateral and longitudinal control of the truck, through a distinction between spatial and temporal dynamics. This distinction allows the separate solving of the steering angle and the optimal velocity. Conversion into spatial domain using $v_1(t) = \pm\frac{\mathrm{d}s(t)}{\mathrm{d}t}$ and $\kappa(t)$ yields for forward and backward driving, respectively,

$$\begin{aligned} x_1'(s) &= \pm\cos\varphi_1(s) \\ y_1'(s) &= \pm\sin\varphi_1(s) \\ \varphi_1'(s) &= \pm\frac{1}{L_1}\tan(\beta(s) + \psi(s)) = \kappa(s) \\ \psi'(s) &= \kappa(s)\left[\frac{L_1}{L_{0b}}\cos\psi(s) - 1\right] \mp \frac{1}{L_{0b}}\sin\psi(s), \end{aligned} \tag{4.7}$$

which is independent of the magnitude of the velocity [41],[26]. Unlike path planning, we can not assume that forward driving is always maintained, so we have to specifically consider reversing. As a result, the differences come from substituting a negative velocity in the dynamics (4.6). Moreover, we define $\dot{s}(t) \geq 0$, meaning that the travelled distance will always increase or remain level. Due to the change in travel direction, the sign of $\kappa(s)$ also changes, cancelling the minus sign in the φ_1-dynamics. By means of the definition of $\kappa(s)$ in (4.7), we can relate the curvature at the trailer rear to the tractor virtual steering angle, according to

$$\beta(s) = \pm\tan^{-1}(L_1\kappa(s)) - \psi(s). \tag{4.8}$$

This observation allows a controller synthesis on the basis of (4.7), using $\kappa(s)$ as the input. From this controller in terms of $\kappa(s)$, the required virtual steering angle $\beta(s)$ can be inferred, from which eventually we know the actual steering angle input $\delta(s)$.

Regarding reference dynamics, the same starting point is chosen as in (4.6), namely

$$\begin{aligned} \dot{x}_{1r}(t) &= v_{1r}(t)\cos\varphi_{1r}(t) \\ \dot{y}_{1r}(t) &= v_{1r}(t)\sin\varphi_{1r}(t) \\ \dot{\varphi}_{1r}(t) &= v_{1r}(t)\kappa_r(t) \\ \dot{\psi}_r(t) &= \frac{1}{L_{0b}}\left[-v_{1r}(t)\sin\psi_r(t) + L_1 v_{1r}(t)\kappa_r(t)\cos\psi_r(t)\right] - v_{1r}(t)\kappa_r(t). \end{aligned} \tag{4.9}$$

Conversion into the spatial coordinate of the truck, by means of $v_{1r}(t) = \pm\frac{\mathrm{d}s_r(s)}{\mathrm{d}s}\frac{\mathrm{d}s(t)}{\mathrm{d}t} = \pm s_r'(s)\frac{\mathrm{d}s(t)}{\mathrm{d}t}$, yields

$$
\begin{aligned}
x'_{1r}(s_r(s)) &= \pm s'_r(s)\cos\varphi_{1r}(s_r(s))\\
y'_{1r}(s_r(s)) &= \pm s'_r(s)\sin\varphi_{1r}(s_r(s))\\
\varphi'_{1r}(s_r(s)) &= s'_r(s)\kappa_r(s_r(s))\\
\psi'_r(s_r(s)) &= s'_r(s)\kappa_r(s_r(s))\left[\frac{L_1}{L_{0b}}\cos\psi_r(s_r(s))-1\right] \mp \frac{1}{L_{0b}}s'_r(s)\sin\psi_r(s_r(s)).
\end{aligned}
\tag{4.10}
$$

Here, we notice the dependency of the reference coordinates on $s_r(s)$, being the travelled distance along the reference. In general, the mapping $s_r(s)$ can be any arbitrary projection of the vehicle onto the reference path. In fact, for error dynamics, we require the relative rates of change $s'_r(s)$ of the distance $s_r(s)$ with respect to s instead of the current mapping. By choice of the error coordinates and associated mapping this rate of change is implied, as shown in the next paragraph.

Error coordinates and projection

The next step is deriving error coordinates, describing the relative difference between the vehicle and the reference positions. Stabilisation of these error coordinates will result in $\lim_{s\to\infty} q(s) - q_r(s_r(s)) = 0$, which is the desired vehicle behaviour. The position coordinates (x, y) are dependent on the choice of inertial frame, leading to the dependency of the error coordinates on this frame. The benefits of taking a frame of reference independent of an arbitrary inertial frame have long been understood [27]. This leads to two possible options, expressing the error dynamics in the frame of the vehicle $\varphi_1(s)$ or that of the reference path $\varphi_{1r}(s_r(s))$, shown in Figure 4.1. In either case, we can significantly simplify the dynamics by taking the mapping $s_r(s)$ such that we can eliminate one error coordinate. This simplifies the dynamics and controller synthesis, at the cost of using the remaining degree of freedom mentioned earlier. Another advantage is that we obtain a unique surjective mapping between vehicle position $(x_1(s), y_1(s))$ and reference point $(x_{1r}(s_r(s)), y_{1r}(s_r(s)))$, meaning that the mapping is time-independent. Any position of the truck maps to a unique reference point, provided that the lateral errors are not too extreme. More elaborate conditions are presented near the end of this section.

Intuitively, taking the shortest distance approach between vehicle and reference path seems the most straightforward projection option. Alternatively, a perpendicular mapping to either the vehicle or reference is also a known choice [26]. The shortest distance projection, which is equivalent to taking the projection perpendicular to the reference, fails for extreme lateral errors as the projection might not be unique. On the other hand, the projection perpendicular to the vehicle fails for extreme heading errors as it can occur that no projection is obtained at all. The resulting error dynamics are significantly simpler in the former case, meaning that we choose the shortest distance projection as our mapping $s_r(s)$. This corresponds to expressing the error in the frame of the projection onto the reference, or of $q_r(s)$, as shown in Figure 4.1. In section 2.2, it has been argued that the best choice of control point is the trailer axles, meaning that the projection is taken from the point $(x_1(s), y_1(s))$ onto $(x_{1r}(s_r(s)), y_{1r}(s_r(s)))$, with $\varphi_{1r}(s_r(s))$ as the orientation of the frame of the errors.

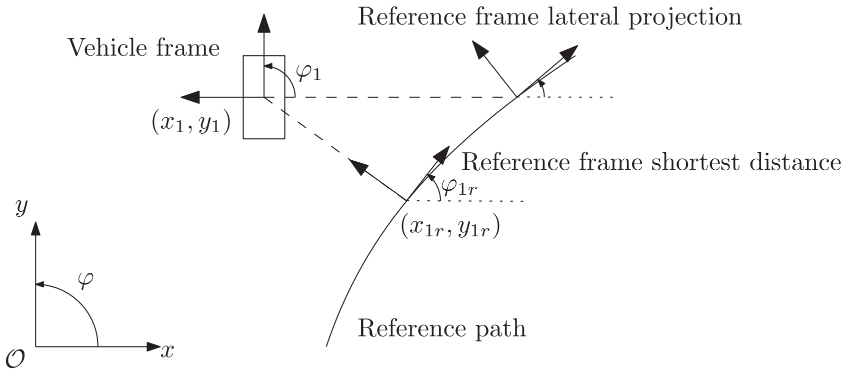


Figure 4.1: Representation of reference projection options with possible error frames.

So far, we have assumed that a mapping $s_r(s)$ from the vehicle to the reference path always exists. In reality, for finite paths and arbitrary initial conditions, this is not the case. For instance, driving slightly too far along a path results in overshooting the stopping point, meaning that the closest point on the reference is slightly behind the vehicle. This is problematic since this means that the position of the truck is not unique, as the projection is based on the assumption that the projection is always perpendicular to the track. One possible option to deal with these situations is to allow arbitrary elongations of the reference path at either end, with corresponding reference states steering back towards the original end of the path. These elongations are not part of the actual reference track, but when the projection falls onto them, they guide the vehicle back to the actual switching or stopping point. This way, the control strategy becomes much more robust to extreme initial conditions and misjudged stopping distances.

From this point onwards, the explicit dependency on reference and travelled distances $s_r(s)$ and s is omitted for sake of brevity. Defining the tracking errors as

$$\begin{aligned} x_e &= (x - x_r)\cos\varphi_{1r} + (y - y_r)\sin\varphi_{1r} \\ y_e &= -(x - x_r)\sin\varphi_{1r} + (y - y_r)\cos\varphi_{1r} \\ \varphi_e &= \varphi_1 - \varphi_{1r} \\ \psi_e &= \psi - \psi_r, \end{aligned} \tag{4.11}$$

yields after differentiation and rearranging the error dynamics

$$\begin{aligned} x_e' &= y_e s_r' \kappa_r \mp s_r' \pm \cos\varphi_e \\ y_e' &= -x_e s_r' \kappa_r \pm \sin\varphi_{1e} \\ \varphi_e' &= \kappa - s_r' \kappa_r \\ \psi_e &= \kappa\left[\frac{L_1}{L_{0b}}\cos(\psi_e + \psi_r) - 1\right] - s_r'\kappa_r\left[\frac{L_1}{L_{0b}}\cos\psi_r - 1\right] \mp \frac{1}{L_{0b}}\sin(\psi_e + \psi_r) \pm \frac{1}{L_{0b}} s_r' \sin\psi_r. \end{aligned} \tag{4.12}$$

Due to the choice of projection, arranging our error frame such that the entire distance between reference and vehicle becomes the y_e-coordinate, we obtain $x_e \equiv 0$. Substituting this in the dynamics above yields

$$\begin{aligned} y_e' &= \pm\sin\varphi_e \\ \varphi_e' &= \kappa - s_r'\kappa_r \\ \psi_e' &= \kappa\left[\frac{L_1}{L_{0b}}\cos(\psi_e + \psi_r) - 1\right] - s_r'\kappa_r\left[\frac{L_1}{L_{0b}}\cos\psi_r - 1\right] \mp \frac{1}{L_{0b}}\sin(\psi_e + \psi_r) \pm \frac{1}{L_{0b}} s_r' \sin\psi_r, \end{aligned} \tag{4.13}$$

as well as the projection rate of change

$$s_r' = \frac{\cos\varphi_e}{1 \mp y_e\kappa_r}. \tag{4.14}$$

This projection rate of change expression indicates the limitations on the maximum acceptable lateral error. For the projection to maintain unique mappings, $1 \mp y_e\kappa_r > 0$ must be satisfied at all times. Practically, this means that the trailer rear may never be at the centre of the inscribed corner of the reference path, since this would mean that the projection change would be undefined. Note that the maximum lateral error at the outside of a reference turn is unbounded.

The dynamics (4.13) are the starting point for the controller design, which controls the input is κ. In short, the control task is to design a controller for the system (4.13), such that it stabilises the error dynamics around the origin. In turn, this implies that the state of the vehicle coincides completely with the state of the reference, as well as the trailer rear coinciding completely with the projection point. Demanding stabilisation of the error system towards the origin implies the use of perfect feedforward, which is the only remaining input signal once this stabilisation has been completed.

4.2 Trailer system

As a first step towards finding a controller for the dynamics in (4.13), a simplified system is considered. Omitting the ψ_e-dynamics yields the dynamics of a car-like vehicle. In this case, this dynamics would refer to the behaviour of the trailer with the assumption of the tractor as the steering wheel, but it could easily be transformed into isolated truck dynamics. We make this simplification to get familiar with nonlinear analysis, controller design and stability proofs, all of which will be useful for the complete system. Consider the system dynamics

$$\begin{aligned} y_e' &= \pm \sin \varphi_e \\ \varphi_e' &= \kappa - s_r' \kappa_r \end{aligned} \tag{4.15}$$

with s_r' according to (4.14), describing the error coordinates of the trailer only.

Lyapunov-based synthesis

We seek to find a stabilising controller using Lyapunov synthesis, for which we need to find a suitable Lyapunov function candidate. In order to obtain a bounded term $c_y \sigma(y_e)$ in the controller, we require it in the first term of V', which we have to integrate for the original Lyapunov function. Therefore, we take the Lyapunov function candidate

$$V = S(y_e) + \frac{1}{c_y}(1 - \cos \varphi_e), \tag{4.16}$$

with $S(\tau) = \int \sigma(\tau) \mathrm{d}\tau$ the integral of the sigmoid function $\sigma(y_e)$ of Definition 2.1.6 and $c_y > 0$ an arbitrary constant. The purpose of taking this integral of the sigmoid function and the $1 - \cos \varphi_e$ term becomes clear when the controller is proposed. Differentiating (4.16) with respect to travelled distance s yields

$$V' = \sigma(y_e) \left[\pm \sin \varphi_e \right] + \frac{1}{c_y} \sin \varphi_e \left[\kappa - s_r' \kappa_r \right], \tag{4.17}$$

for which a suitable controller is given by

$$\begin{aligned} \kappa &= s_r' \kappa_r \mp c_y \sigma(y_e) - c_\varphi \sin \varphi_e \\ &= \frac{\cos \varphi_e}{1 \mp y_e \kappa_r} \kappa_r \mp c_y \sigma(y_e) - c_\varphi \sin \varphi_e, \end{aligned} \tag{4.18}$$

with $c_\varphi > 0$, rendering the Lyapunov function candidate equal to

$$V' = -\frac{c_\varphi}{c_y} \sin^2 \varphi_e, \tag{4.19}$$

which is clearly negative semi-definite. Therefore, we can conclude on the basis of Theorem 2.1.7 that the errors remain bounded and the system is at least Lyapunov stable. Further analysis is required to prove asymptotic convergence towards the origin, which is shown in the next paragraph. In this controller, we recognise that $1 \mp y_e \kappa_r > 0$ is a hard requirement, as this directly stems from the chosen projection. As a result, the above inequality should be guaranteed by proper choice of c_φ and c_y, for which conditions are derived later in this section. Notice that the magnitude of the feedback terms $c_y \sigma(y_e)$ and $c_\varphi \sin \varphi_e$ is now bounded by the controller gains c_y and c_φ, respectively. This illustrates the purpose of using a sigmoid function for the lateral error y_e, namely to saturate the steering angle at extreme lateral errors. Regarding the sigmoid function, as an example, the functions

$$\sigma_1(y_e) = \tanh y_e \qquad\qquad \sigma_2(y_e) = \frac{y_e}{\sqrt{1 + y_e^2}} \tag{4.20}$$

both suffice the requirements of Definition 2.1.6, making them suitable to use in feedback. Their primitives, to be used in the Lyapunov function (4.16), are

$$S_1(y_e) = \ln \cosh y_e \qquad\qquad S_2(y_e) = \sqrt{1 + y_e^2} - 1, \tag{4.21}$$

respectively.

Uniform asymptotic stability

In order to prove GUAS of the nonlinear time-varying controller proposed previously, we require the use of Matrosov's theorem 2.1.8. Consider the closed-loop dynamics

$$\begin{aligned} \tilde{y}_e' &= \pm \sin \tilde{\varphi}_e \\ \tilde{\varphi}_e' &= \mp c_y \sigma(\tilde{y}_e) - c_\varphi \sin \tilde{\varphi}_e, \end{aligned} \tag{4.22}$$

where the controller κ of (4.18) has been substituted. From the controller synthesis, we have that

$$V_1 = S(\tilde{y}_e) + \frac{1}{c_y}(1 - \cos \tilde{\varphi}_e) \tag{4.23}$$

and its spatial derivative

$$V_1' = -\frac{c_\varphi}{c_y} \sin^2 \tilde{\varphi}_e = Y_1(\tilde{\varphi}_e) \tag{4.24}$$

as the first Lyapunov function, which is clearly negative semi-definite. Choosing the second function

$$V_2 = -\tilde{\varphi}_e \tilde{\varphi}_e', \tag{4.25}$$

yields, after differentiation and substitution of $\tilde{\varphi}_e = 0$ the spatial derivative

$$\begin{aligned} V_2' &= -\tilde{\varphi}_e'^2 - \tilde{\varphi}_e \tilde{\varphi}_e'' \\ &= -c_y^2 \sigma(\tilde{y}_e)^2 + F_2(||\tilde{\varphi}_e||) \\ &= Y_2(\tilde{y}_e, \tilde{\varphi}_e), \end{aligned} \tag{4.26}$$

where $F_2(||\tilde{\varphi}_e||)$ is a function of the trailer heading error satisfying $F_2(0) = 0$. Then, we establish that $Y_1(\tilde{\varphi}_e) = 0$ implies that $Y_2(\tilde{y}_e, \tilde{\varphi}_e) \leq 0$ and that $Y_1(\tilde{\varphi}_e) = Y_2(\tilde{y}_e, \tilde{\varphi}_e) = 0$ implies that $\tilde{\varphi}_e = \tilde{y}_e = 0$, with which GUAS for the controlled system (4.22) has been proven. Effectively, this means that either $Y_1(\tilde{\varphi}_e)$ is decreasing, leading towards the origin, or that $Y_1(\tilde{\varphi}_e)$ is stationary and $Y_2(\tilde{y}_e, \tilde{\varphi}_e)$ is not increasing. If $Y_2(\tilde{y}_e, \tilde{\varphi}_e)$ is indeed decreasing, the system once again moves towards the origin, while if both are stationary and no longer decrease, it has been shown that we must be at the origin of the error system.

State constraints

If we want to guarantee that the errors remain bounded by some maximum value, which is required to keep the projection continuous, as shown in (4.14), one option is to look at the Lyapunov functions. Consider the Lyapunov function candidate proposed in (4.16), evaluated for some initial pose of the reference and robot. We then notice that by $V' \leq 0$, the value of $V(s) \leq V(0)$ for all travelled distance s. As a result, we can describe maximum possible values for both states, as a function of the initial poses. Take the requirements $1 \mp y_e \kappa_r^{\max} > 0$ for some maximum reference curvature $\kappa_r^{\max}$ and $\cos \varphi_e > 0$ to guarantee the reference point moving forward at all times. The former can be simplified to

$$|y_e| < \frac{1}{\kappa_r^{\max}}, \tag{4.27}$$

which holds in general for both driving directions and arbitrary reference curvature satisfying $|\kappa_r| < \kappa_r^{\max}$. Furthermore, assume known initial $y_e(0)$ and $\varphi_e(0)$, for which we know that an initial value of the Lyapunov function provides an upper bound to both system states, according to

$$\begin{aligned} V(0) = S(y_e(0)) + \frac{1}{c_y}(1 - \cos \varphi_e(0)) &\geq S(y_e(s)) < S\left(\frac{1}{\kappa_r^{\max}}\right) \\ V(0) = S(y_e(0)) + \frac{1}{c_y}(1 - \cos \varphi_e(0)) &\geq \frac{1}{c_y}(1 - \cos \varphi_e(s)) < \frac{1}{c_y}. \end{aligned} \tag{4.28}$$

In the most extreme case, all of the Lyapunov function value is condensed into a single state, so the leftmost inequalities above are valid in the sense of equality. If we then apply the maximum values that the states $y_e(s)$ and $\varphi_e(s)$ may attain, we obtain two inequalities, from which we can deduce requirements on c_y, the lateral feedback gain. Rewriting the equations above yields

$$\begin{aligned} c_y &> \frac{1-\cos\varphi_e(0)}{S\left(\frac{1}{\kappa_r^{\max}}\right) - S(y_e(0))} \\ c_y &< \frac{\cos\varphi_e(0)}{S(y_e(0))}, \end{aligned} \tag{4.29}$$

where it is interesting to note that this yields an upper and lower bound for the lateral gain c_y, independent of the heading gain c_φ. This is a result of only c_y appearing in the Lyapunov function (4.16). Moreover, this means that for some set of initial values and $\kappa_r^{\max}$, we obtain a range of possible values for c_y, whereas for other options we either have only one or zero feasible options. Equating both requirements provides maximum space to the initial conditions, yielding

$$\cos\varphi_e(0) > \frac{S\left(y_e(0)\right)}{S\left(\frac{1}{\kappa_r^{\max}}\right)}. \tag{4.30}$$

Only for initial conditions satisfying the above inequality can we guarantee, albeit conservatively, that the state constraints are not violated by proper choice of c_y. This inequality can be interpreted as an upper bound on the allowable initial heading error $\varphi_e(0)$. Depending on the amount of leeway remaining in the $1 \mp y_e\kappa_r > 0$ constraint, the acceptable initial heading error varies. In case either initial error is chosen as zero, one trivially finds the other constraint as the limiting factor.

Additionally, it is interesting to examine the case of extreme lateral errors. When the truck returns to the reference path, it should at most attain a heading error of 90 deg, i.e. $\cos\varphi_e \geq 0$. This corresponds to the situation where the lateral error is of such magnitude that the truck drives straight back towards the reference path, perpendicular to it. In order to guarantee that the controller (4.18) tends to zero for $\cos\varphi_e = 0$ and extreme y_e, we require

$$\begin{aligned} \kappa &= \frac{\cos\varphi_e}{1 \mp y_e\kappa_r} \mp c_y\sigma(y_e) - c_\varphi \sin\varphi_e \\ &= \mp c_y\sigma(y_e) - c_\varphi \sin\varphi_e \end{aligned} \tag{4.31}$$

to be nonnegative for $y_e << 0$ (with $\sin\varphi_e = 1$) and nonpositive for $y_e >> 0$ (with $\sin\varphi_e = -1$). This can be achieved due to the sigmoid function used on the lateral error, specifically by taking

$$c_\varphi \geq \pm c_y. \tag{4.32}$$

If implemented, this prevents the truck from returning to the reference path driving in circles and moving back along the reference. Instead, the truck will maintain driving perpendicular to the reference path, until the lateral error is diminished sufficiently and the truck merges with the reference. When close enough, the heading term of the controller begins to dominate the lateral term until both become zero.

4.3 Full truck system

Extending the considered system can be done in multiple ways. We first consider a cascaded control structure, before we attempt to look at the system (4.13) in its entirety and apply the same methods as in section 4.2. Consequently, we simplify the system by looking at time-invariant reference signals, as well as linearisation about these references. For these simplified systems, controllers are synthesised and conditions for stability are provided. Finally, the obtained insights are combined in the design of a complete controller, which is to be tested in a simulation environment.

Constant curvature

Finding a controller that directly stabilises the nonlinear system is attempted in two different ways. First, we take the curvature of the trailer system and use it as the desired curvature which needs to be realised by the tractor. This refers to a cascaded approach, where a desired controller of a subsystem is the reference for the next subsystem. Secondly, we mimic the method of section 4.2 and use a direct Lyapunov synthesis for finding a controller. Both of these methods are shown in Appendix C

For the above approaches, we have not been able to find a stabilising controller for the nonlinear time-varying system directly. Therefore, we attempt to make simplifications by first considering the time-invariant case with references of constant curvature, before looking at the linearisation of the dynamics. Notice that for a reference of constant curvature, $\psi_r' = 0$ holds, from which we obtain

$$\kappa_r \left[\frac{L_1}{L_{0b}} \cos\psi_r - 1 \right] \mp \frac{1}{L_{0b}} \sin\psi_r = 0, \tag{4.33}$$

from which we can either isolate κ_r or ψ_r. Due to the dynamics (4.13) containing only one instance of κ_r, we substitute

$$\kappa_r = \frac{\pm\sin\psi_r}{L_1\cos\psi_r - L_{0b}}, \tag{4.34}$$

yielding the nonlinear time-invariant dynamics

$$\begin{aligned} y_e' &= \pm\sin\varphi_e \\ \varphi_e' &= \kappa \mp s_r' \frac{\sin\psi_r}{L_1\cos\psi_r - L_{0b}} \\ \psi_e' &= \kappa \left[\frac{L_1}{L_{0b}} \cos(\psi_e + \psi_r) - 1 \right] \mp \frac{1}{L_{0b}} \sin(\psi_e + \psi_r), \end{aligned} \tag{4.35}$$

assuming that $|\psi_r| < \cos^{-1}\left(\frac{L_{0b}}{L_1}\right)$. For these time-invariant dynamics, a Lyapunov-based controller synthesis using the Lyapunov function candidate

$$V = S(y_e) + \frac{1}{c_1}(1 - \cos\varphi_e) + \frac{1}{c_2}(1 - \cos\psi_e) \tag{4.36}$$

has not provided any results, for the same reasons described in Appendix C. Even for the simple case of tracking a straight line, or $\kappa_r = \psi_r = 0$, the dynamics

$$\begin{aligned} y_e' &= \pm\sin\varphi_e \\ \varphi_e' &= \kappa \\ \psi_e' &= \kappa \left[\frac{L_1}{L_{0b}} \cos\psi_e - 1 \right] \mp \frac{1}{L_{0b}} \sin\psi_e \end{aligned} \tag{4.37}$$

still do not provide a straightforward Lyapunov-based synthesis using (4.36). This is a result of the complexity of the control task, which requires the simultaneous stabilisation of two trailers, both of which are directly affected by the input κ.

Linearisation

Another approach to simplify the dynamics of (4.13) is to linearise it around the reference $y_e = \varphi_e = \psi_e = 0$ and $\kappa = s_r'\kappa_r = \kappa_r$, yielding

$$\begin{aligned} \bar{y}_e' &= \pm\bar{\varphi}_e \\ \bar{\varphi}_e' &= \mp\kappa_r^2 \bar{y}_e + \bar{\kappa} \\ \bar{\psi}_e' &= \left[\frac{1}{L_{0b}} \sin\psi_r \kappa_r \pm \kappa_r^2 \left(1 - \frac{L_1}{L_{0b}} \cos\psi_r \right) \right] \bar{y}_e \pm \frac{L_1 - L_{0b}\cos\psi_r}{L_{0b}(L_{0b} - L_1\cos\psi_r)} \bar{\psi}_e + \left[\frac{L_1}{L_{0b}} \cos\psi_r - 1 \right] \bar{\kappa}. \end{aligned} \tag{4.38}$$

By applying Lemma 2.1.9, we find that the determinant of the corresponding controllability matrix is always strictly nonzero, meaning that the origin of this system is controllable. Substituting the constant curvature assumption (4.34) yields the LPV dynamics

$$\begin{aligned} \bar{y}_e' &= \pm\bar{\varphi}_e \\ \bar{\varphi}_e' &= \frac{\mp\sin^2\psi_r}{(L_{0b}-L_1\cos\psi_r)^2}\bar{y}_e + \bar{\kappa} \\ \bar{\psi}_e' &= \pm\frac{L_1 - L_{0b}\cos\psi_r}{L_{0b}(L_{0b}-L_1\cos\psi_r)}\bar{\psi}_e + \left[\frac{L_1}{L_{0b}}\cos\psi_r - 1\right]\bar{\kappa}, \end{aligned} \tag{4.39}$$

for which we seek to find a stabilising $\bar{\kappa}$, suitable for a feasible domain on ψ_r. Assuming any state feedback of the form

$$\bar{\kappa} = -c_y y_e - c_\varphi \varphi_e - c_\psi \psi_e, \tag{4.40}$$

yielding a closed loop system characterised by the polynomial $P = \lambda^3 + a_2\lambda^2 + a_1\lambda + a_0$, we require the roots to be in the left half of the complex plane by Theorem 2.1.10. By means of the Routh-Hurwitz criterion of Corollary 2.1.11, we obtain the necessary and sufficient conditions

$$a_2 > 0 \qquad a_0 > 0 \qquad a_2 a_1 - a_0 > 0, \tag{4.41}$$

to guarantee roots with strictly negative real part. For our system, taking into account that $a_1 > 0$ is implied, these become

$$\begin{aligned} a_2 &= -\frac{L_{0b}^2 c_\psi - L_{0b}^2 c_\varphi \pm L_1 \mp L_{0b}\cos\psi_r + L_1^2 c_\psi \cos^2\psi_r + L_1 L_{0b} c_\varphi \cos\psi_r - 2L_1 L_{0b} c_\psi \cos\psi_r}{L_{0b}(L_{0b}-L_1\cos\psi_r)} > 0 \\ a_1 &= \pm c_y \pm \frac{L_1^2\cos\psi_r - L_{0b}(L_1 c_\varphi \pm \cos^2\psi_r + L_1 c_\varphi \cos^2\psi_r \mp 1) + L_{0b}^2 c_\varphi \cos\psi_r}{L_{0b}(L_{0b}-L_1\cos\psi_r)^2} > 0 \\ a_0 &= \frac{1}{L_{0b}} c_y \cos\psi_r - \frac{1}{L_{0b}(L_{0b}-L_1\cos\psi_r)^3}\Big[\sin^2\psi_r(\pm L_1 + L_{0b}^2 c_\psi \mp L_{0b}\cos\psi_r + L_1^3 c_y \cos^2\psi_r \\ &\quad + L_1^2 c_\psi \cos^2\psi_r + L_1 L_{0b}^2 c_y - 2L_1 L_{0b} c_\psi \cos\psi_r - 2L_1^2 L_{0b} c_y \cos\psi_r)\Big] > 0 \\ a_2 a_1 - a_0 &= \frac{\pm 1}{L_{0b}^2(L_{0b}-L_1\cos\psi_r)^2}\Big[\pm L_1^2 c_\varphi \pm L_{0b}^2 c_\varphi - L_1 L_{0b}^2 c_\varphi^2 + L_{0b}^3 c_\varphi^2 \cos\psi_r + L_{0b}^4 c_y c_\varphi - L_{0b}^4 c_y c_\psi \\ &\quad + L_1^3 c_\varphi c_\psi \cos^2\psi_r + L_1 L_{0b}^2 c_\varphi c_\psi \mp 2L_1 L_{0b} c_\varphi \cos\psi_r - L_1 L_{0b}^2 c_\varphi^2 \cos^2\psi_r - L_{0b}^3 c_\varphi c_\psi \cos\psi_r + L_1^2 L_{0b} c_\varphi^2 \cos\psi_r \\ &\quad + 2L_1 L_{0b}^2 c_\varphi c_\psi \cos^2\psi_r - L_1^2 L_{0b} c_\varphi c_\psi \cos^3\psi_r - L_1^3 L_{0b} c_y c_\psi \cos^3\psi_r + L_1^2 L_{0b}^2 c_y c_\varphi \cos^2\psi_r \\ &\quad - 3L_1^2 L_{0b}^2 c_y c_\psi \cos^2\psi_r - 2L_1 L_{0b}^3 c_y c_\varphi \cos\psi_r + 3L_1 L_{0b}^3 c_y c_\psi \cos\psi_r - 2L_1^2 L_{0b} c_\varphi c_\psi \cos\psi_r\Big] > 0 \end{aligned} \tag{4.42}$$

for forward and backward driving, respectively. Isolating from these inequalities the gains c_y, c_φ and c_ψ, we find that they have to fulfil the conditions

$$\begin{aligned} c_y &> 0 \\ c_\varphi &\gtrless -c_y L_{0b} \\ c_\psi &> \frac{1}{(L_{0b}-L_1\cos\psi_r)^2(L_1 c_\varphi - L_{0b}^2 c_y - L_{0b} c_\varphi \cos\psi_r + L_1 L_{0b} c_y \cos\psi_r)}\Big[L_1 L_{0b}^2 c_\varphi \cos^2\psi_r (c_\varphi - L_1 c_y) \\ &\quad + L_{0b} c_\varphi \cos\psi_r (c_\varphi L_1^2 \pm 2c_y L_1 L_{0b}^2 + 2L_1 \mp c_\varphi L_{0b}^2) - c_\varphi(\pm L_1^2 - c_\varphi L_1 L_{0b}^2 + c_y L_{0b}^4 \pm L_{0b}^2)\Big]. \end{aligned} \tag{4.43}$$

The above conditions reveal some insight into the workings of the feedback controller and what the effect of each gain is. Starting with c_y, which requires a positive gain on the lateral error, which is responsible for suppressing the trailer lateral error. Consequently, we have the heading gain c_φ, of which the stability condition is driving direction dependent. For forward driving, we would expect a positive gain, which is not necessarily required. Due to the inherent stable nature of trailers during forward motion, it is permitted to have it slightly working against the lateral gain, as long as it remains sufficiently small. For backward driving, the situation is switched as a result of the sign change first observed in (4.7). Also for the nonlinear controller derived for the isolated trailer system in (4.18), we noticed this sign difference when reversing. In fact, due to the unstable nature in this case, the heading gain needs to dominate this effect and is therefore bounded from below. Regarding the articulation angle gain c_ψ, we find a lower bound, depending on the reference track. Its responsibility is to keep the tractor in line, and prevent it from returning to its stable undesired equilibrium. Looking at the system from the perspective of Figure 2.3, the gain c_ψ keeps the trailer close to the reference, which is clearly more involved in the case of backward driving. Moreover, for increasing reference curvature, the reference articulation angle increases as well, meaning that the tractor rear is further away from its stable equilibrium. This results in a significant increase in required c_ψ, to counteract this instability and maintain stable reversing. As a consequence, limiting the maximum allowable reference curvature allows for a smaller articulation angle gain c_ψ, improving the error attenuation performance of the controller.

Considering the maximum steering angle of the truck of approximately 30 deg and the vehicle dimensions listed in section 2.2, we find a maximum constant articulation angle of approximately $|\psi_r| \leq 1$, satisfying the assumption that $|\psi_r| < \cos^{-1}\left(\frac{L_{0b}}{L_1}\right)$. In a practical scenario, the maximum curvature of the reference path could be used as the determining factor in that case, as not every reference path contains the tightest possible turns. Consequently, we have conditions for local asymptotic stability around the reference for the admissible range of constant curvature references, or circles. In order to verify whether the proposed controller of (4.40) also works for time-varying references, simulations are performed in section 4.5. Here, the limitations of the controller obtained from the linear system are put to the test as well.

Controller extensions

Based on the linear state-feedback controller proposed in (4.40), we can attempt to apply some of the lessons learnt in section 4.2. As long as it is ensured that the linearisation of the controller around the origin of the error system remains identical to the linear state feedback controller, all results and conclusions obtained for the linear controller above remain valid. Recalling the two proposed controllers of (4.18) and (4.40), we find

$$\kappa_{\text{trailer}} = s_r'\kappa_r \mp c_y\sigma(y_e) - c_\varphi \sin\varphi_e \tag{4.44}$$

as the controller derived for the trailer system and

$$\kappa = s_r'\kappa_r - c_y y_e - c_\varphi\varphi_e - c_\psi\psi_e \tag{4.45}$$

as the state feedback controller for the linearised system. Apart from the sign switch at the lateral feedback term, the principal differences lie in the saturation and sine functions. Applying the same logic as in section 4.2, the saturation function will help in preventing excessive steering angles for extreme lateral errors. There are multiple options for feasible sigmoid functions for this application, provided that they satisfy Definition 2.1.6. Furthermore, the sine function more accurately cancels the lateral y_e-dynamics, as well as deals with the angle φ_e correctly. The wrapping of angles, as in $\varphi_e = \varphi_e + 2k\pi$, now leads to correct representation in the controller. Finally, we have the articulation angle feedback term $c_\psi\psi_e$, for which we do not have an example. Since its main function is ensuring the stability of the ψ_e coordinate, stabilising the tractor around the reference, it can be strengthened by taking the tangent function $\tan\psi_e$ instead. This implies that for extreme errors, the associated feedback term tends to infinity. Had it not been for the input constraints, this would have lead to the guarantee that $|\psi_e(s)| < \pi/2$ for all s. Again, this also incorporates the domain of angles properly, meaning that $\tan\psi_e = \tan(\psi_e + 2\pi)$. Combining the above statements and observations yields the nonlinear controller

$$\kappa = s_r'\kappa_r - c_y\sigma(y_e) - c_\varphi\sin\varphi_e - c_\psi\tan\psi_e, \tag{4.46}$$

of which the linearisation remains identical to the prior choice, hence local stability is preserved for gains satisfying (4.43). It would also be possible to flip the sign of the lateral feedback term and maintain a constant condition on c_φ, but this serves no real purpose. The results from section 4.2, regarding conditions on the gains to satisfy state constraints and input constraints at extreme errors, no longer hold. They can, however, be used as guidelines when tuning gains.

Converting this controller back to the δ controller that is required for the eventual hardware implementation requires (2.3), (4.8) and (4.46). Rewriting yields

$$\begin{aligned} \delta &= \tan^{-1}\left(\frac{L_0}{L_{0b}}\tan\left(\pm\tan^{-1}(L_1\kappa)-\psi\right)\right) \\ &= \tan^{-1}\left(\frac{L_0}{L_{0b}}\cdot\frac{\pm L_1\kappa-\tan\psi}{1\pm L_1\kappa\tan\psi}\right) \end{aligned} \tag{4.47}$$

with the curvature controller using the square root sigmoid function, given by

$$\kappa = \frac{\cos\varphi_e}{1\mp y_e\kappa_r}\kappa_r - c_y\frac{y_e}{\sqrt{1+y_e^2}} - c_\varphi\sin\varphi_e - c_\psi\tan\psi_e. \tag{4.48}$$

From a physical perspective, the behaviour of controller (4.48) makes sense. First of all, we have a feedforward term based on the reference curvature and projection speed s_r'. For small tracking errors, this projection speed tends to one, meaning that the reference curvature is used as direct feedforward. In the case of errors, the feedforward term is supplemented by a nonlinear state feedback part. Consider for instance the case when driving backwards and close to a straight reference path, a deviation of the rear of the trailer to the left ($y_e > 0$) requires steering to the left, as does a clockwise deviation of the orientation of the trailer ($\pi > \varphi_e > 0$). If the truck itself has a clockwise orientation deviation ($\frac{1}{2}\pi < \psi_e < 0$) steering to the right is required. For forward driving, the situation is precisely reversed, due to the $\pm$ signs in the inverse kinematics (4.47).

Regarding tuning intuition, we have the lateral gain c_y taking the role of a proportional gain on the error. Increasing its value decreases rise time at the cost of overshoot. Secondly, the trailer heading gain c_φ acts as a differential term to the system behaviour. Due to the y_e-dynamics, whose derivative is given by $\pm\sin\varphi_e$, we recognise that increasing the lateral gain adds damping to the controlled system. Generally, this leads to slightly increased rise times, but reduced overshoot and especially reduced settling times. These two gains are the main tuning knobs for adapting the controller, since c_ψ is primarily introduced to maintain the stability of the articulation angle. A high gain ensures that the articulation angle remains close to the reference, which provides very little freedom for the system to suppress other errors. Therefore, this gain should be chosen as low as possible, while maintaining the conditions for stability of (4.43). Recall that the requirement on c_ψ is dependent on the gains c_y and c_φ, as well as on the maximum value of ψ_r considered path. Increasing c_ψ further than necessary increases rise time and settling time and has no noticeable effect on overshoot.

4.4 Velocity control

The paths and controllers derived so far have only considered the spatial domain, meaning that it was made completely independent of time. Employing the relationship $v_1(t) = \pm\frac{\mathrm{d}s(t)}{\mathrm{d}t}$ we converted the dynamics into spatial domain and eliminated the effect of the magnitude of the velocity. Still, a velocity controller is required to go back from the analysis and synthesis domain towards a practical setting. Since the spatial dynamics are not affected by the driven velocity, we assume that the travelled trajectory is known a priori. Specifically, the curvature of the trailer can be computed based on the controller and the initial conditions all the way until the final position. Admittedly, due to imperfections in modelling and noise, this path is not driven exactly, leading to small deviations in realised curvature. This can be solved by recomputing the predicted curvature ahead and associated desired velocity at each discrete point in time, or whenever some boundary is exceeded. Since there is no reference value associated with the velocity, as the control task is simply to drive the maximum admissible velocity, the velocity profile is the result of a constrained linear optimisation problem.

Optimisation problem formulation

Starting off, we establish the velocity of any point on the truck as $v(s(t)) = \frac{\mathrm{d}s(t)}{\mathrm{d}t}$, with $s(t)$ the travelled distance of the associated point. Restricting ourselves to the point of no side-slip, namely the rear axles of tractor and trailer, the velocities above are purely longitudinal velocities, equal to $v_0(s(t))$ and $v_1(s(t))$

of Figure 2.1. From this, we also obtain the accelerations $a_x(s(t)) = \frac{\mathrm{d}v(s(t))}{\mathrm{d}t} = \frac{\mathrm{d}^2 s(t)}{\mathrm{d}t^2}$ at both the tractor and trailer rear. Even though the lateral velocities are zero, the lateral accelerations are given by

$$a_{0y}(s(t)) = v_0^2(s(t))\kappa_0(s(t)) \qquad\qquad a_{1y}(s(t)) = v_1^2(s(t))\kappa_1(s(t)), \tag{4.49}$$

denoting the tractor and trailer, respectively, with $\kappa_i(s(t))$ the curvature of the path at the corresponding axle. Note that by the controller design, only $\kappa_1(s(t))$ is available, but that this can easily be rewritten into the curvature at the truck. For all relevant accelerations and velocities, we can define limits to shape the desired velocity profile. Define these limits as

$$\begin{aligned}
v_0^{\min} &\leq v_0(s(t)) \leq v_0^{\max} \\
a_{0x}^{+,\min} &\leq a_{0x}^{+}(s(t)) \leq a_{0x}^{+,\max} \\
a_{0x}^{-,\min} &\leq a_{0x}^{-}(s(t)) \leq a_{0x}^{-,\max} \\
|a_{0y}(s(t))| &\leq a_{0y}^{\max} \\
|a_{1y}(s(t))| &\leq a_{1y}^{\max},
\end{aligned} \tag{4.50}$$

to accommodate for maximum velocity with forward and backward driving, maximum acceleration and deceleration with both forward (a^+) and backward (a^-) driving and the maximum lateral acceleration of both the truck and trailer. With these, all friction, powertrain, comfort and safety requirements can be guaranteed. Note that the longitudinal velocity of the trailer is not listed, as the combination of trailer lateral acceleration and tractor velocity through (2.10) sufficiently bounds the trailer velocity. It could simply be added if need be, but is in general never the limiting constraint.

By expressing all constraints of (4.50) into the position of a single control point, we can once again significantly simplify the resulting optimisation problem. It turns out we have four (or eight) constraints on the tractor rear, where the input $v_0(t)$ applies, while we only have one (or two) on the trailer rear. The straightforward choice is therefore the tractor rear as the control point of the velocity optimisation problem. Expressing the known curvature at the trailer rear into the curvature at the tractor rear requires the dynamics of (2.12), yielding

$$\kappa_0(s(t)) = \frac{L_1\kappa_1(s(t)) \mp \tan\psi(s(t))}{L_{0b}\left[1 \pm L_1\kappa_1(s(t))\tan\psi(s(t))\right]}, \tag{4.51}$$

where use has been made of $\dot{\varphi}_0(s(t)) = v_0(s(t))\kappa_0(s(t))$. With the relationship between the velocities $v_0(s(t))$ and $v_1(s(t))$ of (2.10), we can express all constraints of (4.50) into the trailer axle. These constraints, combined with the aim to maximise the longitudinal velocity of the tractor, yields the desired linear optimisation problem.

Finding the optimal velocity

Knowing that the velocity optimisation problem is an optimisation problem with linear cost function and constraints, the minimiser is a so-called bang-bang solution. The solution will always satisfy at least one constraint in the sense of equality, meaning that we can always take the most stringent one. First of all, the constraints of (4.50) are compared at each distance s to obtain the pointwise maximum velocity. We do assume that all future path information, meaning $\kappa(s)$ and $q(s)$, is calculable and known. Moreover, note that in the velocity optimisation, the tractor rear is considered the control point, meaning that the travelled distance s indicates the travelled distance of the tractor. This step takes lateral acceleration limits and the maximum velocity into account, according to

$$v_{0,\text{point}}(s) = \min\left(v_0^{\max}, \sqrt{\frac{a_{0y}^{\max}}{\kappa_0(s)}}, \sqrt{\frac{a_{1y}^{\max}}{\kappa_1(s)}\frac{\cos\beta(s)}{\cos(\beta(s)+\psi(s))}}\right), \tag{4.52}$$

where the explicit dependency of $s(t)$ on time t has been omitted and $\beta(s)$ and $\psi(s)$ are obtained from the current input and vehicle state, respectively. Taking the minimum out of the three terms ensures that the corresponding constraints are met at each point along the path. A switching point can easily be included by setting $v_{0,\text{point}}(s_{\text{switching}}) = 0$, where $s_{\text{switching}}$ is the value of s at the switching point.

The calculated velocity profile is the ideal velocity without taking powertrain and braking capabilities into consideration. In order to do so, we discretise the path into equidistant, arbitrarily small portions, for which a constant acceleration is assumed. This allows the incorporation of the maximum longitudinal acceleration of the truck by iteratively moving forward from the initial velocity of $v_0(0)$. Considering the definitions of acceleration and velocity of rectilinear motion

$$a_{0x}(s) = \frac{\mathrm{d}v_0(s)}{\mathrm{d}t} \qquad\qquad v_0(s) = \frac{\mathrm{d}s}{\mathrm{d}t}, \tag{4.53}$$

we can eliminate $\mathrm{d}t$ and obtain $v_0(s)\mathrm{d}v = a_{0x}(s)\mathrm{d}s$. Integrating both sides from s to $s + \Delta s$, assuming constant acceleration for this interval, yields

$$\frac{1}{2}\left[v_0^2(s+\Delta s) - v_0^2(s)\right] = a_{0x}(s)\Delta s. \tag{4.54}$$

Isolating $v_0(s + \Delta s)$ results in the updating scheme for forward driving, given by

$$v_{0,\text{forward}}(s+\Delta s) = \min\left(v_{0,\text{point}}(s+\Delta s), \sqrt{v_0^2 + 2a_{0x}^{+,\max}\Delta s}\right), \tag{4.55}$$

where clearly the pointwise velocity may not be exceeded. Finally, we perform a very similar approach as the forward integration approach, but this time taking the maximum deceleration of the vehicle into account. The corresponding updating sequence starts at the known final velocity $v_0(s_f)$ and satisfies

$$v_{0,\text{backward}}(s-\Delta s) = \min\left(v_{0,\text{point}}(s-\Delta s), \sqrt{v_0^2 - 2a_{0x}^{+,\min}\Delta s}\right). \tag{4.56}$$

This backward optimisation procedure is the latest step in the implemented method, meaning that it yields $v_0(s)$. For backward driving, an analogous approach is performed, but with the adjusted longitudinal acceleration limits $a_{0x}^{-,\max}$ and $a_{0x}^{-,\min}$.

In short, this yields a feasible velocity profile, satisfying both the lateral and longitudinal constraints on the truck. It is optimal within the resolution chosen for Δs, assuming that infinite jerk can be applied to the system. Since it is a linear optimisation procedure, the optimum will always be on the boundary of the feasible domain, meaning that at least one constraint is met in the sense of equality. Most of the time, this will be the maximum acceleration or deceleration that is active. In the case of sharp turns, however, the pointwise velocity may be limiting for a certain distance, being the constraint met in equality in this case.

4.5 Validation and limitations

In the previous sections, methods have been introduced to design a controller on the basis of a complicated nonlinear system. A controller is proposed which is proven only locally stable around the reference track for the time-invariant case. To investigate its performance in the time-varying, nonlinear application it is designed for, we require a realistic simulation environment of the controller truck system. Along with the tracking results, attention is given to generating Lyapunov functions which (locally) provide stability guarantees to the system. Finally, the limitations of the controller regarding initial conditions and controller gains are investigated, before concluding the chapter.

Simulation results

The plant model of the simulation environment is based on the dynamics of (2.12), where the velocity conversion of (2.10) and steering angle conversion (2.3) are used. The considered truck parameters and dimensions are listed in A, which belong to the scaled version that is used in the TruckLab. The reference signals are either generated by an arbitrary $\kappa(s)$ function or by the output of the path planning algorithm, both of which imply the full state of the system provided known initial conditions. The controller (4.48) has been implemented, together with the inverse kinematics of (4.47) to obtain the tractor steering angle.

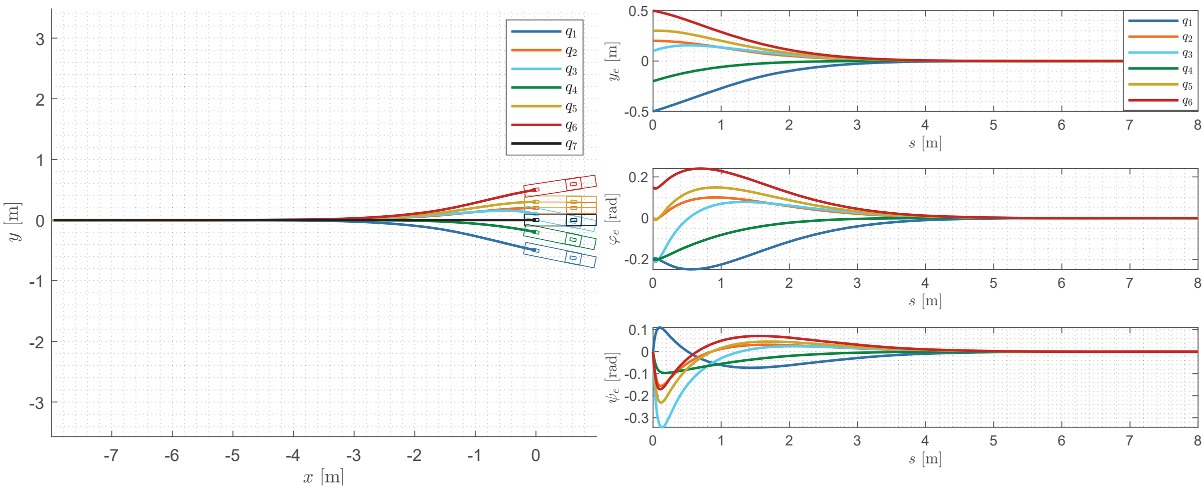
Testing the performance of the controller requires choosing suitable gains. Assuming some path without too sharp turns, we can apply the conditions of (4.43) to find a set of suitable gains. We choose the set

$$c_y = 2 \qquad c_\varphi = 3 \qquad c_\psi = 1 \tag{4.57}$$

for forward driving and

$$c_y = 2 \qquad c_\varphi = -3 \qquad c_\psi = 4 \tag{4.58}$$

for reversing motions. The difference in c_ψ is required to maintain articulation angle stability, where we recognise that the constraint in (4.43) is stricter for backward driving, resulting in a higher gain. First of all, consider the simplest possible reference path, a straight line, being defined by a constant reference curvature of zero. Taking several initial conditions around this reference yields the results in Figure C.2 for forward driving. Figure 4.2 shows the same initial conditions for backward driving. We recognise stable reference tracking for initial errors deviating significantly from the origin. The transient behaviour is relatively short and the closed-loop system exhibits no overshoot. The plots for input curvature and Lyapunov function traces are provided in Appendix C.



(a) Path overview.

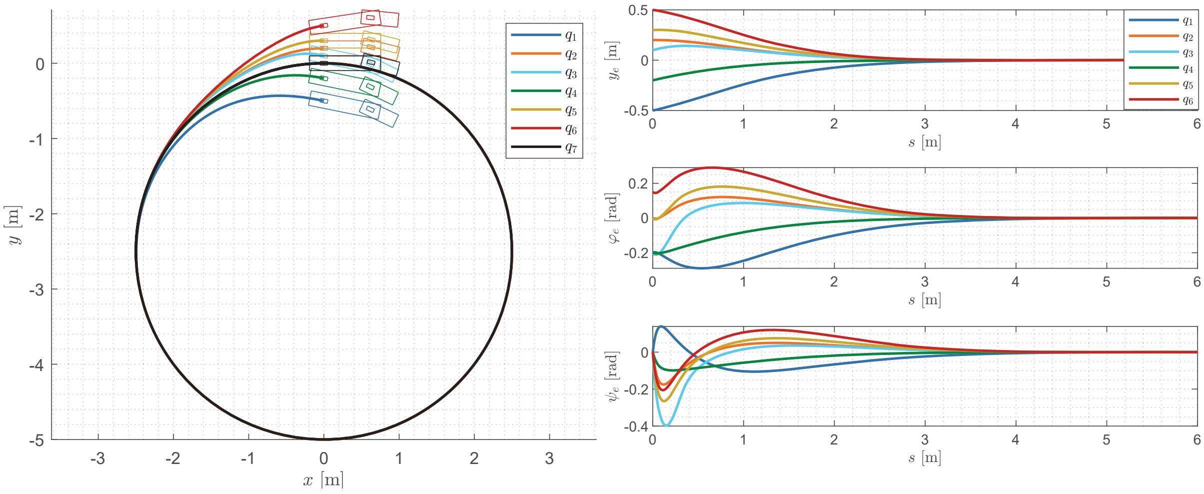
(b) Error traces.

Figure 4.2: Controller validation for backward driving on straight reference path.

In addition to straight-line references, we can also consider nonzero constant curvatures, yielding circular reference paths. Figures C.5 and 4.3 show simulation results for varying initial conditions for a constant curvature of $\kappa_r(t) = -0.4$. The supporting graphs for input curvature and Lyapunov function traces can be found in Appendix C. These simulations show that the qualitative behaviour is very similar for both driving directions, meaning that good error attenuation performance is obtained. Furthermore, perfect tracking is observed after stabilisation of the transient effects has been completed, as the feedforward is identical to the reference steering angle in this case.

Tracking a time-varying reference is more relevant to validate in simulation, as there is no proof that the controller yields a stable closed-loop system. Considering the TruckLab scenario of Figure 3.2, we obtain the tracking results shown in Figure 4.4, supported by Appendix C. Here, we recognise that the controller is capable of dealing with time-varying reference curvatures and significant errors simultaneously. The switching point is also dealt with correctly, inducing no further difficulties to the tracking performance.

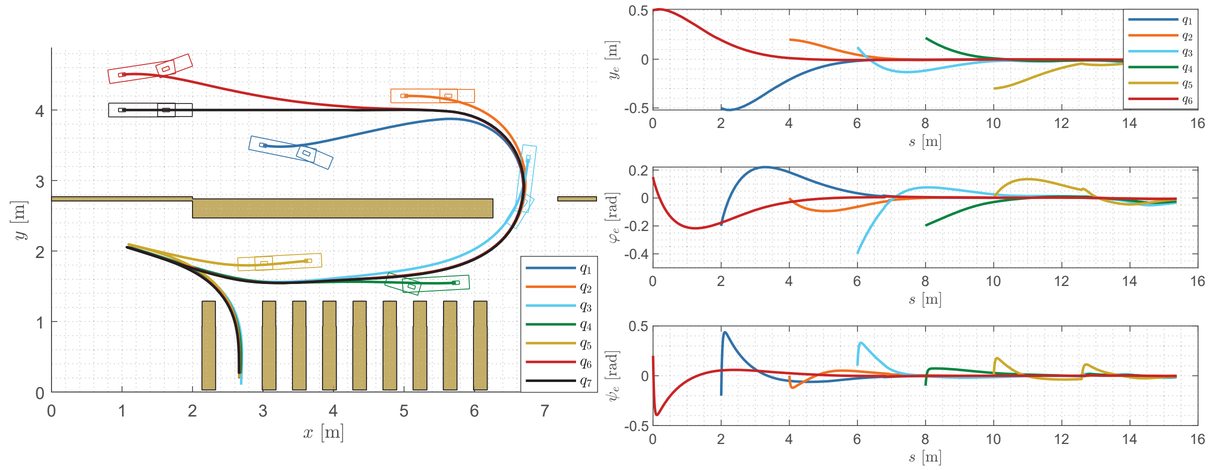
Regarding the driven velocity, where the method of section 4.4 was applied, we obtain the velocity trace shown in Figure 4.5 for the TruckLab scenario. Here, parameters are chosen such that the different regimes in the optimisation problem are displayed clearly. We clearly recognise the start and endpoints, as well as the switching point at approximately 13m. Moreover, the sections where the powertrain, the maximum velocity and the lateral accelerations are limiting can also be distinguished. This figure also shows the trailer velocity being significantly smaller than the tractor velocity in corners, which is to be expected and also follows from (2.10).



(a) Path overview.

(b) Error traces.

Figure 4.3: Controller validation for backward driving on circular reference path.



(a) Path overview.

(b) Error traces.

Figure 4.4: Controller validation for TruckLab scenario.

Lyapunov functions

In addition to simulation results supporting the idea that the proposed controller is stabilising and exhibits sufficient performance, we can also consider Lyapunov functions. Finding a function satisfying the properties of a Lyapunov function, introduced in Theorem 2.1.7, further supports the idea that the closed-loop system is stable. First of all, the linearisation allows a quadratic Lyapunov function of the form

$$V = e^T P(\psi_r) e, \tag{4.59}$$

with $e = (\bar{y}_e, \bar{\varphi}_e, \bar{\psi}_e)^T$ the error coordinates and $P(\psi_r) > 0$ a positive definite weighting matrix that is dependent on a constant reference articulation angle. By means of the Lyapunov equation

$$\tilde{A}^T(\psi_r)P(\psi_r) + P(\psi_r)\tilde{A}(\psi_r) = -Q, \tag{4.60}$$

where $\tilde{A}(\psi_r)$ is the closed-loop system matrix satisfying $e' = \tilde{A}(\psi_r)e$ and $Q > 0$ an arbitrary positive definite matrix, we can find a Lyapunov function to the closed loop system of $\tilde{A}(\psi_r)$. Taking $Q = I_3$ yields a Lyapunov function for both forward and backward driving for any given articulation angle, given

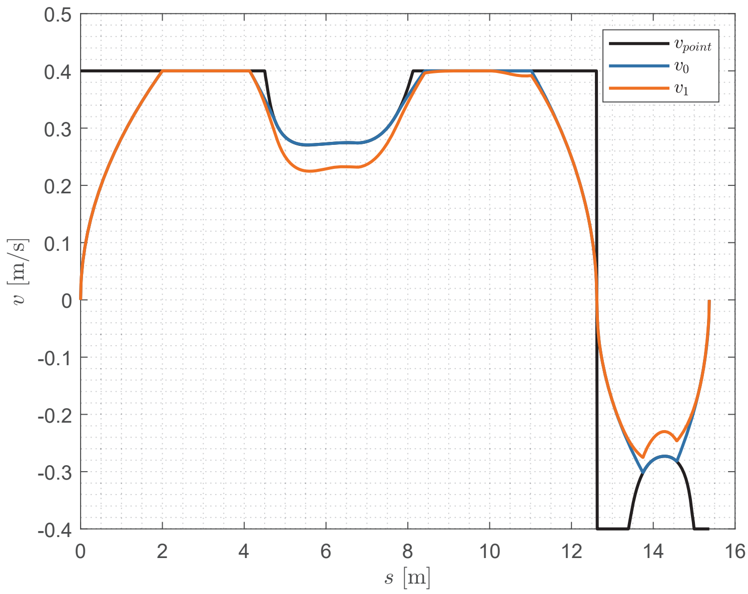


Figure 4.5: Velocity trace example for TruckLab scenario.

by (4.59). For the straight line $\psi_r = 0$ example, we obtain

$$P^+(0) = \begin{bmatrix} 1.58 & 0.75 & -0.04 \\ 0.75 & 1.17 & -0.07 \\ -0.04 & -0.07 & 0.05 \end{bmatrix} \qquad P^-(0) = \begin{bmatrix} 1.68 & -0.89 & 0.10 \\ -0.89 & 1.36 & -0.16 \\ 0.10 & -0.16 & 0.06 \end{bmatrix}, \tag{4.61}$$

for forward and backward driving, respectively. The Lyapunov function should always decrease when the error system is close to the origin, meaning when V is small. The resulting Lyapunov function traces for the TruckLab scenario are shown in Figure 4.6, where the Lyapunov function for $\psi_r = 0$ is displayed and switched according to driving direction.

In addition to a quadratic Lyapunov function at a given constant articulation angle ψ_r, we attempt to find a constant $P_\mathrm{c}(\psi_r)$ that holds for all admissible values of ψ_r. To this end, we define a polytopic system and solve the LMI problem

$$\begin{aligned} \textbf{Minimise} \quad & \tau \\ \textbf{s.t.} \quad & \tilde{A}(\psi_r)^T Q + Q\tilde{A}(\psi_r) < \tau I_3 \\ & \forall\ \psi_r \in [-\psi_r^{\max}, \psi_r^{\max}] \\ & Q = Q^T \\ & Q > I, \end{aligned} \tag{4.62}$$

where we can reconstruct $P_\mathrm{c}(\psi_r) = Q^{-1}$, yielding a common quadratic Lyapunov function (CQLF) for all admissible constant curvatures. For the set of gains proposed in (4.57) and (4.58), this LMI yields a feasible solution, meaning that the closed-loop linear system is indeed quadratically stable for constant curvature. The Lyapunov function (4.59) with the matrix

$$P_\mathrm{c}^+ = \begin{bmatrix} 51.9 & 61.5 & 49.1 \\ 61.5 & 88.4 & 70.8 \\ 49.1 & 70.8 & 69.9 \end{bmatrix} \qquad P_\mathrm{c}^- = \begin{bmatrix} 75.2 & -94.0 & 47.4 \\ -94.0 & 135.0 & -71.9 \\ 47.4 & -71.9 & 50.5 \end{bmatrix} \tag{4.63}$$

proves this, for forward and backward driving, respectively. Note that this does not imply that a time-varying reference signal is stabilised, even when containing only admissible reference articulation angles. Additionally, we find a different CQLF for both driving directions, as both the system dynamics and controller gains differ. Around a switching point, it should be ensured that no Zeno behaviour, meaning no uncontrolled switching between forward and backward driving, can occur. Zeno behaviour could prevent the Lyapunov functions from decreasing towards zero, as the system essentially becomes stuck at a switching point.

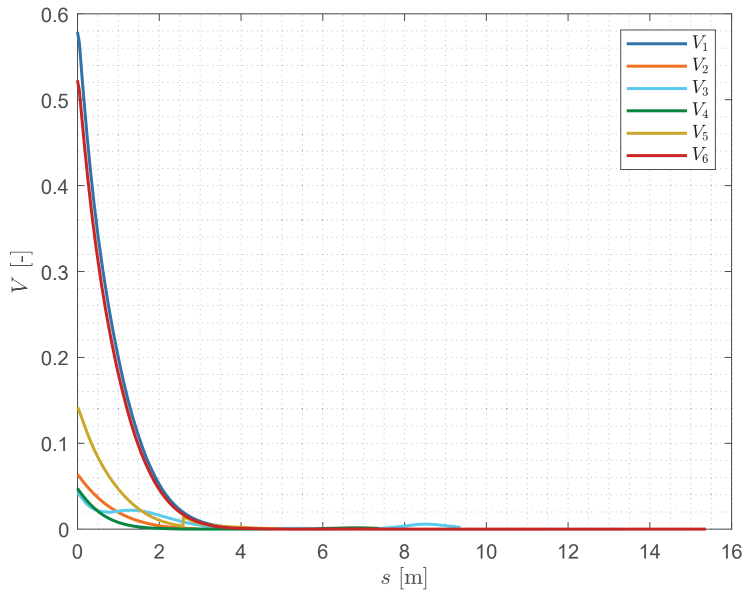


Figure 4.6: Lyapunov functions for TruckLab scenario corresponding to the initial conditions of Figure 4.4.

In addition to a quadratic Lyapunov function, we can reconsider Lyapunov functions of the form (4.36). Attempting to find gains c_1 and c_2 such that the resulting derivative is never increasing, runs into the issue shown in Figure 4.7. These are the three (unweighted) terms of the nonlinear Lyapunov function, where an initial condition of $\tilde{y}_e(0) = 0.5$ and $\tilde{\varphi}_e(0) = \tilde{\psi}_e(0) = 0$ is considered. Here, we recognise that in order to decrease the lateral error, the articulation error first has to increase. This means that no weights c_1 and c_2 exist for which the sum of the three terms never increases. Presumably, cross-terms are required to yield such a Lyapunov function, but this lies outside the scope of this thesis.

Controller limitations

Since the proposed controller of equations (4.47) and (4.48) is based on the linear system, there are supposedly initial conditions for which the controller fails to stabilise the system. Moreover, from the controller synthesis, we recall that the projection onto the reference path is defined provided that (4.14) remains bounded. To this end, the inequality $1 \mp \tilde{y}_e \kappa_r > 0$ must hold for all distance s. Additionally, the conversion (4.47) contains the system dimensions, meaning that incorrect estimation of these dimensions leads to an incorrect control law.

Executing simulations with some extreme initial errors, we obtain Figure 4.8. This result shows that the controller behaves properly, even in the case of extreme lateral and heading errors. The trajectory of the path corresponding to the largest initial lateral error shows some overshoot in x-direction, which is further increased for larger initial conditions. For initial errors at the right side of the track, the first turn is required to be taken tighter than the reference. In a practical setting, there is a limit to the amount of corner that can be cut, due to steering system and articulation angle limitations. Regarding the heading error, it is clear that all traces quickly converge to the desired reference path. Interestingly, for $\tilde{\varphi}_e(0) = \pi$ the controller does not manage to turn around the truck, as this initial condition coincides with the unstable equilibrium of the controlled system. On close inspection, we also recognise the reference path extension to guarantee a feasible mapping when driving behind the initial condition. Even though the

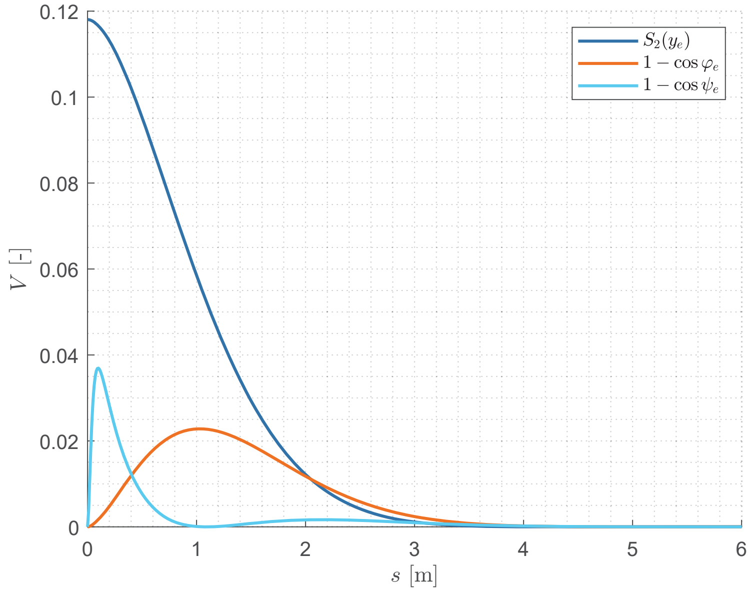
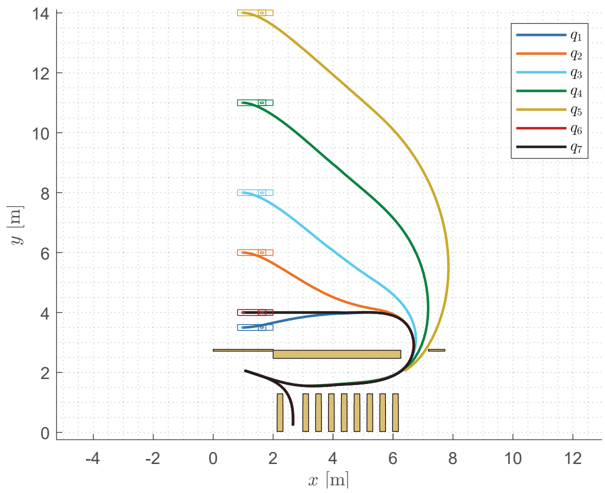
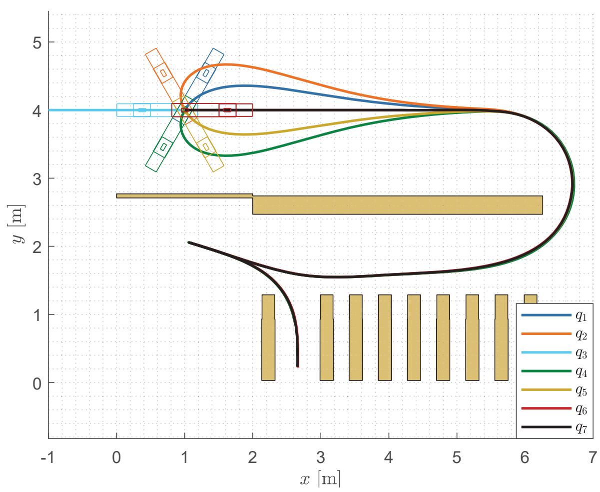


Figure 4.7: Nonlinear Lyapunov function terms.

reference path was not defined for $x < 1$, this extension still guides the truck back to the original reference path.



(a) Initial lateral error.



(b) Initial heading error.

Figure 4.8: Extreme initial condition simulations.

If we take an initial condition close to or past the limit on $1 \mp \tilde{y}_e \kappa_r > 0$, we find the results in Figure 4.9. The most extreme admissible initial condition is plotted here, which lies very close to the centre of the circle the reference path currently forms. Note that a curvature κ_r corresponds to a radius $\frac{1}{\kappa_r}$, where on the centre the shortest distance projection to the reference track may be undefined. Attempting to simulate an initial condition exceeding this restriction leads to undefined projections and chattering behaviour, meaning that this should be avoided at all times.

Modelling errors are expected to have a detrimental effect on the tracking performance of the controller, even once stabilisation of the initial errors has been completed. The most common modelling error would be the trailer length, as this can not be measured beforehand, is highly trailer-dependent and even varies with load. Different loading conditions affect the weight distribution, which influences the effective point

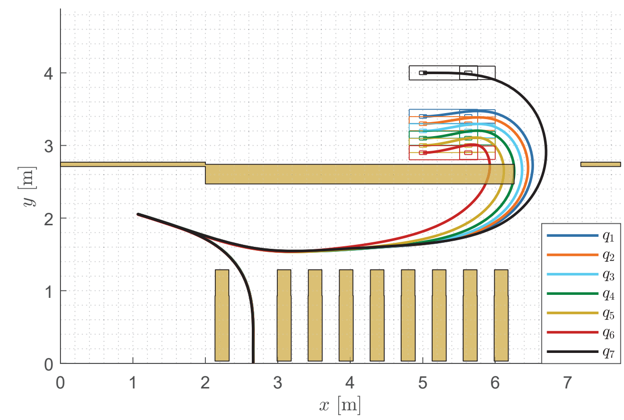


Figure 4.9: Extreme initial conditions for projection feasibility constraint.

of rotation of the trailer axles. Investigating a wrongly identified trailer length L_1 yields the results in Figure 4.10, where lengths between $0.5L_1$ (q_1) and $2L_1$ (q_6) were considered in the controller. The resulting initial error attenuation for the straight line is not too affected. The feedforward signal much more so, as the conversion from reference to eventual steering angle uses the inaccurate L_1 estimation. This observation emphasises the importance of an accurate online estimation of the effective trailer lengths, as well as the other relevant vehicle dimensions.

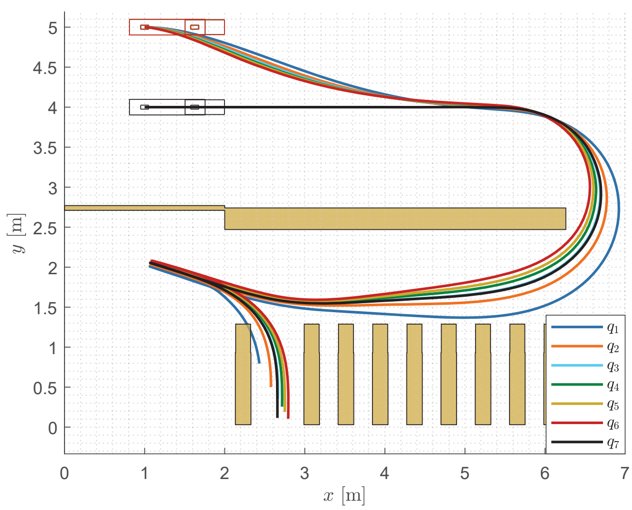


Figure 4.10: Simulation results using wrongly estimated trailer lengths L_1.

4.6 Observations and conclusions

In this chapter, a controller stabilising the tractor semi-trailer system around an arbitrary full-state reference has been proposed. To this end, error dynamics have been defined, using the shortest distance mapping between vehicle and reference path. These error dynamics are defined in spatial coordinates, meaning that they are independent of the magnitude of the velocity. In order to synthesise a controller, a simplified system was considered first, consisting of a trailer only. The lessons learnt and methods used here are then applied to the full system. However, simplifications were required to eventually obtain a locally stabilising controller, namely a constant reference curvature assumption and linearisation. These controllers have been expanded using nonlinear functions to extend their performance and area of applicability. The eventual controller performance has been validated in a simulation environment, where it was tested for multiple different references and extreme initial conditions. Regarding velocity optimisation, a simple linear optimisation problem was proposed. This optimisation problem maximised the driven velocity, subject to powertrain, safety and comfort constraints. A minor drawback is the computation involved with determining $\kappa(s)$ ahead of the vehicle, which is required to anticipate turns coming up. It should be recomputed every so often to compensate for model mismatches or estimation errors accumulating when driving.

The resulting controller is capable of stabilising the error dynamics in all realistic scenarios tested. Necessary and sufficient conditions for gain tuning have been derived, meaning that a lot of design freedom remains to influence the behaviour of the controller. Moreover, the controller is capable of perfect reference tracking, as it uses the reference steering angle as the feedforward signal. Regarding stability proofs, only local stability of the time-invariant system for arbitrary admissible curvatures has been proven. Any extensions towards obtaining global or uniform results have not been successful. Moreover, the convergence speed at which errors are attenuated is not quantitatively assessed. No guarantees have been found that prove that the error has been been brought sufficiently close to the origin for successful docking.

One drawback of the proposed control strategy is the sensitivity to estimation and modelling errors. The state and model of the system need to be mapped accurately for the assumptions made in the controller design to hold. Additionally, the system has not been proven stable for all feasible operating conditions, although simulations show that large initial errors and time-varying references are dealt with correctly. Furthermore, the choices of control point and projection are the most intuitive, yielding the simplest analysis and controller synthesis. The performance, however, might not be better than that of a controller for which the projection was not statically defined. Leaving the projection $s_r(s)$ as an additional controller input gives the controller more flexibility and possibly improves tracking performance.

Chapter 5

Conclusions and recommendations

The role of road freight transport on the Dutch and European roads is predicted to continue to grow. Most of this transport is performed by tractor semi-trailer combinations, known for their flexibility and efficiency. In the current society of rapid engineering progress and automation, numerous opportunities exist to improve the logistics sector. This thesis aims to contribute by specifically considering autonomously driving trucks around warehouses and distribution centres. These are low-speed, closed-off and predictable environments with few external influences. In this chapter, a summary of the thesis and its primary conclusions and recommendations for future research are stated.

In order to describe the motion and dynamic behaviour of tractor semi-trailers, a kinematic model has been derived to model and predict the states of the vehicle. Based on one of these models, a kinodynamic path planning algorithm is developed, aiming to yield a feasible and optimal path, taking vehicle constraints and external obstacles into account. It takes the outputs of a coarse high-level planner and uses that as initialisation to identify how obstacles ought to be passed and where the truck should turn around. Then, a nonlinear optimisation problem is formulated based on the curvature function, which is parametrised using quintic splines. With the coefficients of the spline as the output of the optimisation problem, the states of the truck in spatial domain are fully determined. The objective function that is to be minimised is a weighted sum of the path length, the bending energy (a measure of curvature) and steering intensity (a measure of derivative of curvature). Two options for taking constraints into account have been presented, one based on distance functions between simple shape approximations of the truck and its environment, and the other on the hyperplane separation theorem between arbitrary convex polygons.

Based on the same single track vehicle model as used in the path planning part of this thesis, a nonlinear reference tracking controller has been designed. To this end, error coordinates using the shortest distance projection have been chosen, such that stabilisation of the error coordinates implies perfect reference tracking. The projection serves as a unique mapping between vehicle state and reference state, simplifying the resulting expressions and ensuring vehicle behaviour independent of initialisation or re-calculation of reference paths. The error dynamics, which are defined in spatial domain, are dependent on the driving direction of the truck. For both directions, a control strategy has been derived and analysed for an isolated trailer system. The lessons learnt for this simple controller are applied to the entire vehicle system, where linearisation and constant curvature assumptions are required to sufficiently simplify the dynamics. Based on this linearisation, a state feedback controller is proposed, with sufficient and necessary conditions for gain tuning to warrant local asymptotic stability. Then, the dynamics are converted back to time domain through the optimal velocity profile along the entire path, taking powertrain, friction and comfort limitations into account. The controller and velocity profile have been applied to several reference paths, showing good and reliable reference tracking.

In conclusion, a pathfinding algorithm has been developed which respects general state and obstacle constraints. It employs a model of the entire vehicle, meaning that feasibility and continuity of all vehicle states are guaranteed. It does so in a rather conservative manner, however, leading to loss of 'completeness', since the planner may fail to find a feasible path when one exists. Moreover, the computational and numerical complexity of the optimisation problem prevents real-time application. The separating axis theorem method for collision avoidance specifically, works in an isolated environment but complicates the optimisation to such an extent that often no feasible solutions are found. In fact, neither obstacle avoidance method behaves completely as desired. The re-parametrisation of the spline coefficients using curve knot spacing acts as a scaling factor between the optimisation parameters and allows intuitive initialisation. It reduces the computational complexity of the optimisation problem, but unfortunately not enough to warrant online implementation. The planner is also reliant on decent output of the high-level planner for arbitrary paths, meaning that it can only function independently for simple paths of few turns.

The proposed control strategy is capable of perfect reference tracking, using the steering angle obtained from the reference vehicle as feedforward. Moreover, in close proximity to the reference, the controller has been proven locally asymptotically stable and conditions have been provided allowing tuning without compromising this stability. Using nonlinear functions, the controller has been improved to deal with the domain of angles, as well as saturate the input in case of extreme lateral errors. No stability guarantees have been derived for the general nonlinear system, however, which means that reference tracking with arbitrary errors and time-varying reference curvatures is not proven reliable. By means of simulation, it has been shown that for realistically feasible reference paths, the controller behaves well and provides excellent tracking performance. The control strategy provides an identical tuning method for forward and backward driving, allowing a relatively simple and intuitive setup and operation. The velocity optimisation is simple, but reliable and robust, although it requires simulation in future time to anticipate the path ahead.

Recommendations

Even though good results have been obtained in both path planning and tracking controller design, various improvements are possible. Below, a list of the most important recommendations for future work are provided.

- **Experimentation**
  The performance and robustness of the proposed planning and control strategies have only been tested in a simulation environment. Performing real-life experiments should validate the proposed methods and be a starting point for further improvements.
- **Online collision avoidance**
  In addition to tracking a collision-free path, online safety features need to be implemented to deal with situations in which extreme errors or dynamical objects would otherwise cause a collision.
- **Obstacle detection**
  Neither proposed obstacle detection method performs as desired. The bounding box method introduces significant conservatism in the solution, while the separating axis theorem method is numerically too harsh on the solver, meaning that a new method should be introduced.
- **State and system estimation**
  The reliance on accurate and reliable kinematic models and estimation is very significant. In both planning and control, a large sensitivity towards the system states and effective trailer lengths exists, meaning that a state and parameter accurate estimation procedure is required. One option is to implement an Extended Kalman Filter.
- **Path planner complexity**
  Before the proposed planner can be applied in a practical setting, its implementation needs to be revisited and its numerical complexity reduced. Currently, it is not robust, nor fast enough for practical implementation. Improving the implementation will also allow higher-order spline approximations, further increasing the freedom of the planner.
- **Stability proof**
  A global (uniform) stability proof of the controlled nonlinear system has not been done yet, which would be desired. Even though multiple simulations show good reference tracking, the controlled system intricacies have not been fully understood.
- **Error attenuation**
  No error attenuation guarantees have been found to prove that the controller with specific gains sufficiently rejects initial errors. Successful docking requires small lateral and heading errors, meaning that disturbances and errors should be attenuated swiftly.

Bibliography

[1] Imagocampagne transportsector. Can be found online at `www.adformatie.nl/carriere/imagocampagne-transportsector`, November 2009. (In Dutch).

[2] Centraal Bureau voor de Statistiek. Monitor topsectoren 2018. Can be found online at `www.cbs.nl/maatwerk/2018/41/monitor-topsectoren-2018`, Den Haag, the Netherlands, October 2018. (In Dutch).

[3] Centraal Bureau voor de Statistiek. Meer goederen vervoerd over de weg en over zee in 2018. Can be found online at `www.cbs.nl/nieuws/2019/29/meer-goederen-vervoerd-over-de-weg-en-over-zee-in-2018`, Den Haag, the Netherlands, July 2019. (In Dutch).

[4] Rabobank. Beroepsgoederenvervoer over de weg. Can be found online at `www.rabobank.nl/bedrijven/cijfers-en-trends/transport-en-logistiek/beroepsgoederenvervoer-over-de-weg`. (In Dutch).

[5] A. Tipping and P. Kauschke. Shifting patterns, the future of the logistics industry. PwC's future in sight series, Can be found online at `www.pwc.com/transport`, 2016.

[6] M. Overmeer and K. Verweij. Trends in xxl-distributiecentra in het pre- en post coronatijdperk. *Logistiek vastgoed 2020*, pages 76–77, 2020. (In Dutch).

[7] Transmetrics. Top 10 supply chain and logistics technology trends in 2021. Can be found online at `www.transmetrics.ai/blog/supply-chain-logistics-technology-trends`, February 2021.

[8] C. You, J. Lu, D. Filev, and P. Tsiotras. Autonomous planning and control for intelligent vehicles in traffic. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2339–2349, 2020.

[9] O. Gungor and I. Al-Qadi. All for one: Centralized optimization of truck platoons to improve roadway infrastructure sustainability. *Transportation Research Part C: Emerging Technologies*, 114:84–98, 2020.

[10] T. Hertogh. Automatic docking of articulated vehicles, 2017. DC 2017.030, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven University of Technology.

[11] I. Mashood Badshah. Automatic docking of a scaled tractor semi-trailer. Master's thesis, DC 2018.005, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands, January 2018.

[12] A. Sudhakaran. Autonomous parallel parking of a scaled tractor semi-trailer. Master's thesis, DC 2018.006, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands, January 2018.

[13] Y. Li, Q. Shi, and D. Qiu. Parameter identification of tractor-semitrailer model under steering and braking. *Mathematical Problems in Engineering vol. 2019, Article ID 1345364*, 2019.

[14] L. Xu, E. Tseng, T. Pilutti, and S. Schondorf. Yaw rate based trailer hitch angle estimation for trailer backup assist. In *WCX™ 17: SAE World Congress Experience*, Detroit, USA, March 2017. SAE International.

[15] M. Diachuk, S. Easa, and J. Bannis. Path and control planning for autonomous vehicles in restricted space and low speed. *MDPI - Infrastructures*, 42(5), 2020.

[16] B. Donald and P. Xavier. Kinodynamic motion planning. *Journal of the Association for Computing Machinery*, 40(5), 1993.

[17] S. Coenen. Motion planning for mobile robots — a guide. Master's thesis, CST 2012.108, Control Systems Technology Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands, November 2012.

[18] M. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley and Sons, Inc., Hoboken, New Jersey, USA, November 2005. 1st press, ISBN: 978-0-471-64990-8.

[19] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[20] F. Lamiraux, E. Ferre, and E. Vallee. Kinodynamic motion planning: connecting exploration trees using trajectory optimization methods. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 4, pages 3987–3992 Vol.4, 2004.

[21] J. Morales, J.L. Martínez, A. Mandow, and A.J. García-Cerezo. Steering the last trailer as a virtual tractor for reversing vehicles with passive on- and off-axle hitches. *IEEE Transactions on Industrial Electronics*, 60(12):5729–5736, 2013.

[22] C. Altafini, A. Speranzon, and B. Wahlberg. A feedback control scheme for reversing a truck and trailer vehicle. *IEEE Transations on Robotics and Automation*, 17(6):915–922, 2001.

[23] M. Werling and M. Heidingsfeld. Reversing the general one-trailer system: Asymptotic curvature stabilization and path tracking. *IEEE Transations on Intelligent Transportation Systems*, 15(2):627–636, 2014.

[24] N. T. Binh, N. A. Tung, D. P. Nam, and N. H. Quang. An adaptive backstepping trajectory tracking control of a tractor trailer wheeled mobile robot. *International Journal of Control, Automation and Systems*, 17(2):465–473, 2019.

[25] J. Yuan, F. Sun, and Y. Huang. Trajectory generation and tracking control for double-steering tractor–trailer mobile robots with on-axle hitching. *IEEE Transactions on Industrial Electronics*, 62(12):7665–7677, 2015.

[26] E. Lefeber, J. Ploeg, and H. Nijmeijer. A spatial approach to control of platooning vehicles: Separating path-following from tracking. In *Proceedings of the 20th World Congress of the International Federation of Automatic Control*, pages 15565–15570, Toulouse, France, 2017.

[27] Y. Kanayama, K. Yoshihiko, M. Fumio, and N. Tetsuo. A stable tracking control method for an autonomous mobile robot. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation; Cincinnati, OH, USA*, pages 384–389, May 1990.

[28] K. Kural. *Analysis of High Capacity Vehicles for Europe*. PhD thesis, Eindhoven University of Technology, Eindhoven, the Netherlands, August 2019. ISBN: 978-90-386-4853-8.

[29] N. Nair. Path planning for auto docking of articulated vehicles. Master's thesis, DC 2019.055, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands, June 2018.

[30] S. Rajagopalan. Matlab based control of a scaled tractor semi-trailer. Master's thesis, DC 2017.031, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands, March 2017.

[31] G. Smeulders. Development and testing of a scaled tractor for use in the trucklab. Master's thesis, DC 2020.050, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands, August 2020.

[32] Centraal Bureau voor de Statistiek. Aantal trekkers voor oplegger blijft stijgen. Can be found online at `www.cbs.nl/nieuws/2017/42/aantal-trekkers-voor-oplegger-blijft-stijgen`, Den Haag, the Netherlands, October 2017. (In Dutch).

[33] B. Barsky and A. DeRose. Geometric continuity of parametric curves. EECS Department, University of California, Berkeley, U.S.A., oct 1984.

[34] I. Barbalat. Systems des equations differentielles des oscillations nonlineaires. *Revue Roumaine de Mathematique Pures et Appliquees*, 4:267–270, 1959.

[35] J. Jimenez-Delgado, R. Segura, and F. Felto. Efficient collision detection between 2d polygons. *Journal of WSCG*, January 2004.

[36] S. Boyd and L Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[37] H. Khalil. *Nonlinear Systems*. Pearson Education Limited, Essex, United Kingdom, 2014. 3rd edition.

[38] V. Matrosov. On the stability of motion. *Journal of Applied Mathematics and Mechanics*, 26(5):1337–1353, 1962.

[39] E. Routh. A treatise on the stability of a given state of motion. Macmillan and co., London, United Kingdom, 1877.

[40] L. Hendrix. Implementation low-level path planner, November 2020. Internship report. Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven University of Technology.

[41] Y. Zhang, H. Chen, S. L. Waslander, J. Gong, G. Xiong, T. Yang, and K. Liu. Hybrid trajectory planning for autonomous driving in highly constrained environments. *IEEE Access*, 6:32800–32819, 2018.

[42] P. Rouchon, M. Fliess, J. Lévine, and P. Martin. Flatness, motion planning and trailer systems. In *IEEE Proceedings on the 32nd Conference on Decision and Control, San Antonio, Texas*, volume 8, pages 2700–2705, December 1993.

[43] B. H. Pourciau. Multiplier rules and the separation of convex sets. *Journal of Optimization Theory and Applications*, 40(3):321–331, July 1983.

[44] R. van Hoek, J. Ploeg, and H. Nijmeijer. Gap closing for cooperative driving in automated vehicles using b-splines for trajectory planning. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 370–375, Las Vegas (NV), U.S.A, 2020.

Appendix A

Preliminaries

The considered scaled trucks are developed within the TU/e research [31]. Their dimensions, as well as the full truck dimensions the scaled model is based on, are listed in Table A.1. The subscript L_f refers to the front of the truck, while L_r refers to the rear. Furthermore, subscript L_o indicates overhang, meaning the length of the truck past either the kingpin joint or the rear effective axle.

Table A.1: Full truck and scaled model dimensions [31].

Dimension	**Symbol**	**Full truck [m]**	**Scaled [m]**
Tractor			
Wheelbase	L_0	3.60	0.271
King pin - rear axle	L_{0b}	0.47	0.035
King pin - front axle	L_{0f}	3.13	0.236
Front overhang	L_{0fo}	1.37	0.103
Rear overhang	L_{0ro}	0.99	0.074
Total length	$L_0 + L_{0fo} + L_{0ro}$	5.96	0.448
King pin height	-	1.19	0.090
Total height	-	3.53	0.264
Total width	-	2.48	0.187
Trailer			
Wheelbase	L_1	7.62	0.573
Front overhang	L_{1fo}	1.68	0.126
Rear overhang	L_{1ro}	2.50	0.188
Total length	$L_1 + L_{1fo} + L_{1ro}$	13.68	1.028
King pin height	-	1.19	0.090
Total height	-	4.00	0.301
Total width	-	2.55	0.192
Axle spacing	-	1.31	0.099

Appendix B

Path planning

Derivations

In this section, the intermediate steps required for (3.4) are detailed. Recall

$$\varphi_0'(s) = \frac{1}{L_{0b}\cos\theta(s)}\sin\beta(s), \tag{B.1}$$

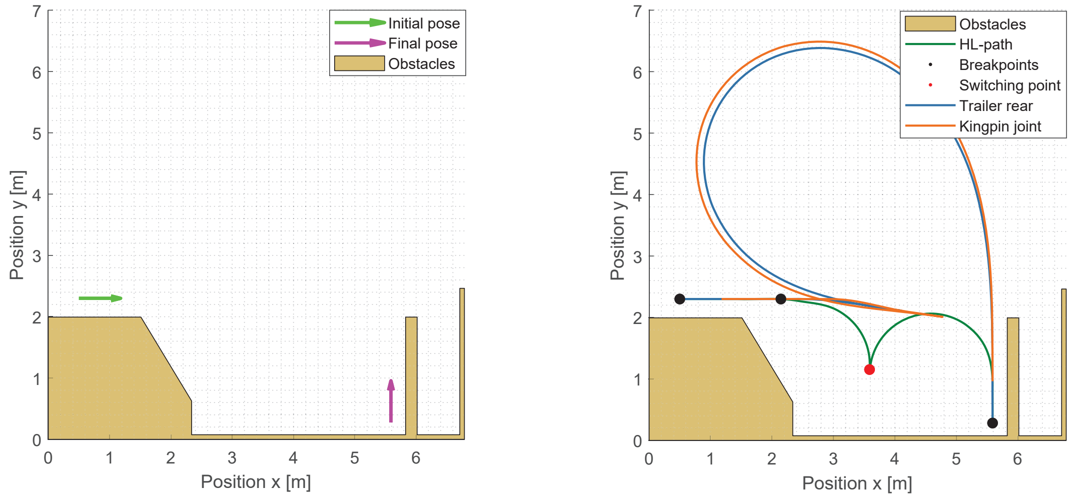
where we substitute by means of the φ_1-dynamics the relationship $\theta(s) = \tan^{-1}(L_1\kappa(s))$. Executing and simplifying yields

$$\begin{aligned}\varphi_0'(s) &= \frac{1}{L_{0b}\cos(\tan^{-1}(L_1\kappa(s)))}\sin\beta(s)\\ &= \frac{1}{L_{0b}}\sqrt{1 + L_1^2\kappa^2(s)}\sin\beta(s),\end{aligned} \tag{B.2}$$

where we have arrived at the desired result of (3.4).

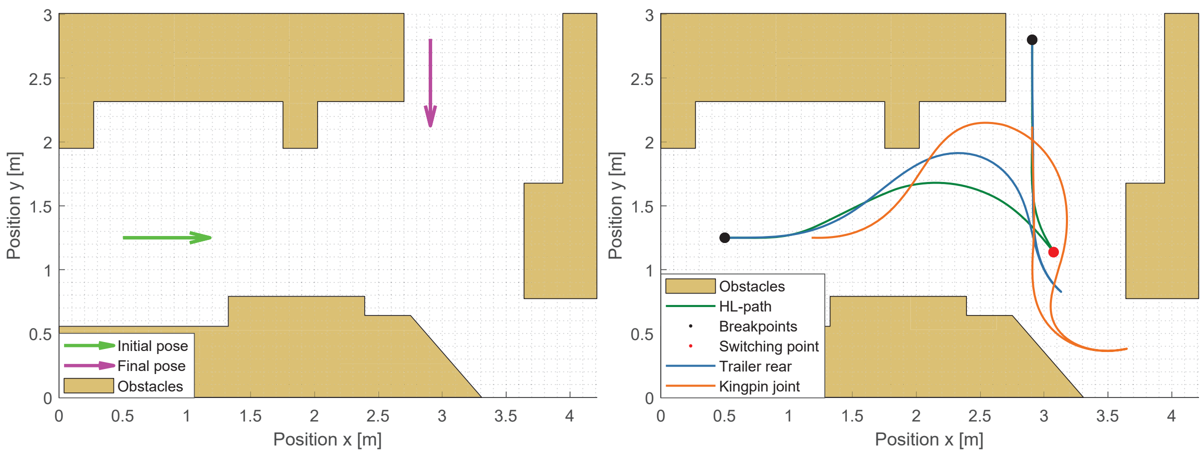
Planning results

The results of the path planner for the four remaining benchmark scenarios, designed by [29], are shown in figures B.1, B.2, B.3 and B.4. The choice of gains for the objective function is $\alpha_1 = 0.001$ and $\alpha_2 = 0.029$. Here, both the high-level path, as well as the low-level path designed in this thesis are shown and compared. The low-level path is indicated by two lines, denoting the movements of the trailer rear and kingpin joint. We observe that the low-level path is significantly longer than the HL-path, as it has to satisfy all kinodynamic constraints of the vehicle. Moreover, it stands out that for scenario 4, where the HL-planner actually fails to yield a feasible path, the proposed planner is still capable of planning.



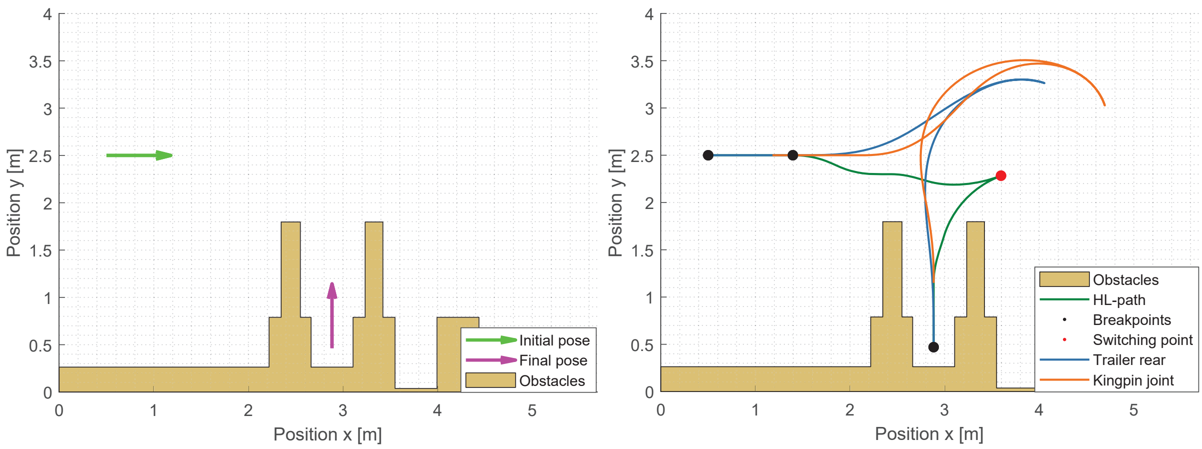
(a) Scenario. (b) Planned path.

Figure B.1: Benchmark scenario 1.



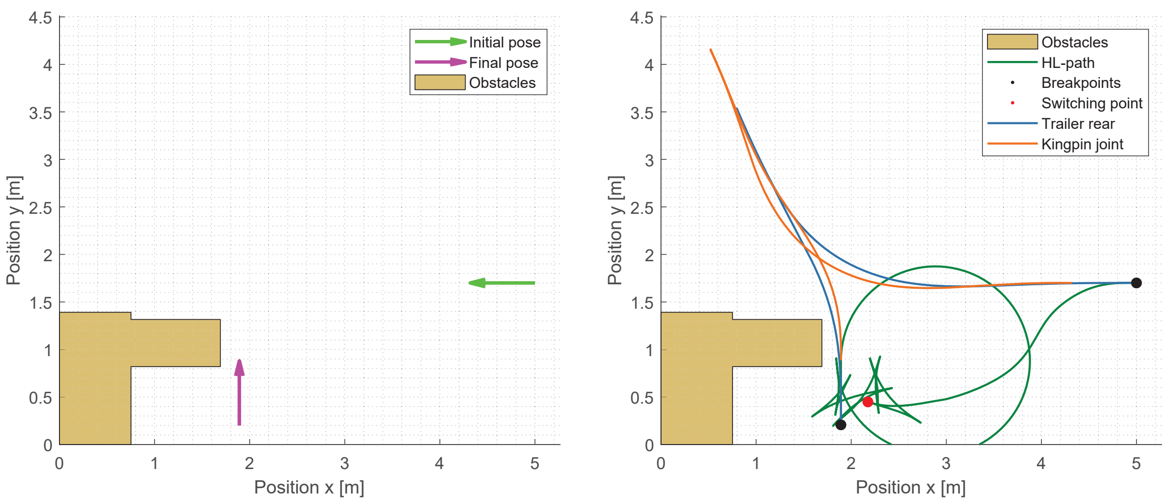
(a) Scenario. (b) Planned path.

Figure B.2: Benchmark scenario 2.



(a) Scenario. (b) Planned path.

Figure B.3: Benchmark scenario 3.



(a) Scenario. (b) Planned path.

Figure B.4: Benchmark scenario 4.

Appendix C

Tracking control

Nonlinear controller synthesis

In section 4.2, we have established that the controller in (4.18) stabilises the trailer error system (4.15) and that conditions can be found for which the state constraints and specific input constraints are respected. We then seek to find a controller that directly stabilises the nonlinear system, to which we describe two synthesis methods in the following sections.

Cascaded approach

Consider this controller κ as the desired curvature κ_d at the trailer rear, which is to be ensured by manoeuvring the tractor properly. We then convert this desired curvature κ_d to a desired virtual steering angle θ_d at the kingpin joint, according to

$$\theta_d = \pm \tan^{-1}(L_1 \kappa_d), \tag{C.1}$$

where the φ_1-dynamics of (4.7) and $\theta = \beta + \psi$ are used. Taking the error coordinate $\theta_e = \theta - \theta_d$ yields the very naive controller choice

$$\beta = \pm \tan^{-1}(L_1 \kappa_d) - \psi, \tag{C.2}$$

which does indeed stabilise $\theta_e = 0$ and hence ensures that $\kappa = \kappa_d$, but fails to stabilise the internal dynamics of the tractor. Taking for instance the articulation angle ψ, we find

$$\psi' = \kappa_d \left[\frac{L_1}{L_{0b}} \cos\psi - 1 \right] \mp \frac{1}{L_{0b}} \sin\psi, \tag{C.3}$$

where the virtual control law κ_d does not consider the ψ dynamics. Here, in the situation of a straight reference, when the stabilisation of the trailer has been completed, we can clearly see the remaining dynamics

$$\psi' = \mp \frac{1}{L_{0b}} \sin\psi \tag{C.4}$$

being unstable for backward driving. This is a direct result of the system properties shown in Figure 2.3, where the steering law has to stabilise both the tractor and trailer rear simultaneously when reversing. The controller β of (C.2) fails to account for the articulation angle ψ, meaning that both β and ψ have their stable equilibrium at π, which is clearly undesirable. Instead, we introduce another error coordinate $\beta_e = \beta - \beta_r$, yielding the system

$$\begin{aligned} y_e' &= \pm \sin\varphi_e \\ \varphi_e' &= \kappa - s_r' \kappa_r \\ \theta_e' &= \beta' + \psi' \mp \frac{L_1 \kappa_d'}{1 + L_1^2 \kappa_d^2} \\ \beta_e' &= \beta' - s_r' \beta_r'. \end{aligned} \tag{C.5}$$

Substitution of the cascaded controller

$$\beta' = \frac{\pm L_1 \kappa_d'}{1 + L_1^2 \kappa_d^2} - \psi' + u \tag{C.6}$$

yields

$$
\begin{aligned}
y_e' &= \pm \sin\varphi_e \\
\varphi_e' &= \frac{L_1\kappa_d \pm \tan\theta_e}{L_1(1 \mp L_1\kappa_d \tan\theta_e)} - s_r'\kappa_r \\
\theta_e' &= u \\
\beta_e' &= \frac{\pm L_1\kappa_d'}{1+L_1^2\kappa_d^2} - \psi' - s_r'\beta_r' + u,
\end{aligned}
\tag{C.7}
$$

with

$$
\begin{aligned}
\kappa_d &= \kappa_r s_r' \mp c_y\sigma(y_e) - c_\varphi \sin\varphi_e \\
\kappa_d' &= \kappa_r' s_r' + \kappa_r s_r'' - c_y \frac{\mathrm{d}\sigma(y_e)}{\mathrm{d}y_e}\sin\varphi_{1e} - c_\varphi\cos(\varphi_e)\varphi_e' \\
s_r' &= \frac{\cos\varphi_e}{1-y_e\kappa_r} \\
s_r'' &= \frac{-\sin(\varphi_e)\varphi_e'(1-y_e\kappa_r) + \cos(\varphi_e)(y_e'\kappa_r + y_e\kappa_r')}{(1-y_e\kappa_r)^2} \\
\psi' &= \frac{L_1\kappa_d \pm \tan\theta_e}{L_1(1 \mp L_1\kappa_d\tan\theta_e)}\left[\frac{L_1}{L_{0b}}\cos\psi - 1\right] \mp \frac{1}{L_{0b}}\sin\psi \\
\psi &= \pm\tan^{-1}(L_1\kappa_d) + \theta_e - \beta_r - \beta_e
\end{aligned}
\tag{C.8}
$$

In this system, the error β_e is required to be stabilised to yield a stable system, meaning that the tractor rear will be stabilised as well as the trailer. Controller u remains as the to be designed input to this system, for which it is possible to locally stabilise this system around the origin. Even though the approach seemed simple and intuitive, namely to proceed with the controller of the previous section and steer the tractor such that the desired trailer behaviour is obtained, the resulting expressions have become very complicated. This is a result of the controller operating on the spatial derivative of the steering angle β', resulting in a dynamic controller and requiring the derivatives of most of the system states.

Lyapunov-based

Instead, we reconsider the dynamics of (4.13), where we directly seek to apply a Lyapunov-based synthesis, similar to the approach in section 4.2. Taking the Lyapunov function candidate

$$
V = S(y_e) + \frac{1}{c_1}(1-\cos\varphi_e) + \frac{1}{c_2}(1-\cos\psi_e) \tag{C.9}
$$

yields the spatial derivative

$$
\begin{aligned}
V' = \sigma(y_e)\left[\pm\sin\varphi_e\right] &+ \frac{1}{c_1}\sin\varphi_e\left[\kappa - s_r'\kappa_r\right] + \frac{1}{c_2}\sin\psi_e\Bigg[\kappa\left[\frac{L_1}{L_{0b}}\cos(\psi_e+\psi_r) - 1\right] \\
&- s_r'\kappa_r\left[\frac{L_1}{L_{0b}}\cos\psi_r - 1\right] \mp \frac{1}{L_{0b}}\sin(\psi_e+\psi_r) \pm \frac{1}{L_{0b}}s_r'\sin\psi_r\Bigg],
\end{aligned}
\tag{C.10}
$$

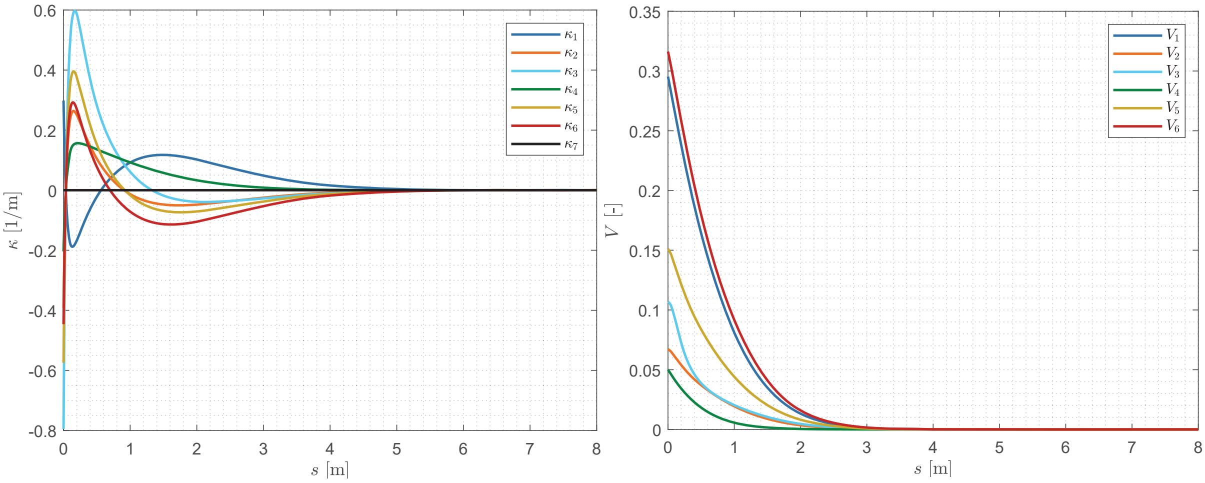
where the virtual input κ occurs multiple times. Here, it is strictly possible to find a control law making this Lyapunov function derivative negative (semi)definite, however it implies using the denominator

$$
\kappa \sim \frac{1}{\sin\varphi_e + \frac{1}{L_{0b}}\sin\psi_e(L_1\cos\psi_e - 1)}. \tag{C.11}
$$

Equating this denominator to zero describes a set of poses, including the origin, for which the controller becomes unbounded, meaning that this controller is not applicable. Taking a linearisation of the error system around the origin yields a Lyapunov function containing cross-terms, which is supposedly also required for the nonlinear system.

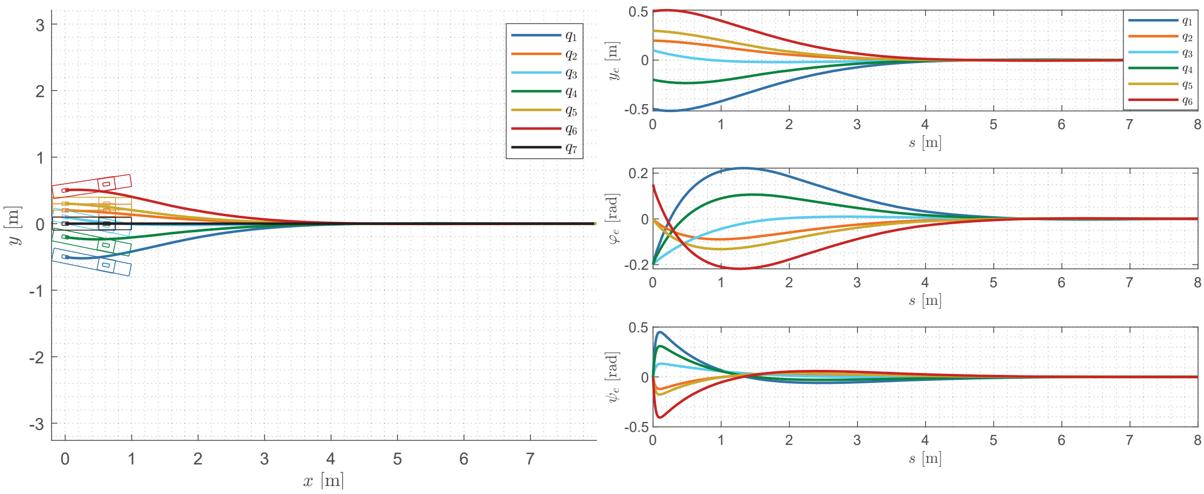
Tracking simulations

The additional plots from the simulations of section 4.5 are shown in this section. Here, multiple initial conditions, over several reference paths, in both driving directions are considered and good error attenuation and reference tracking is observed for each of them. The outline of the initial pose of the truck is shown to illustrate the initial errors of the simulations.



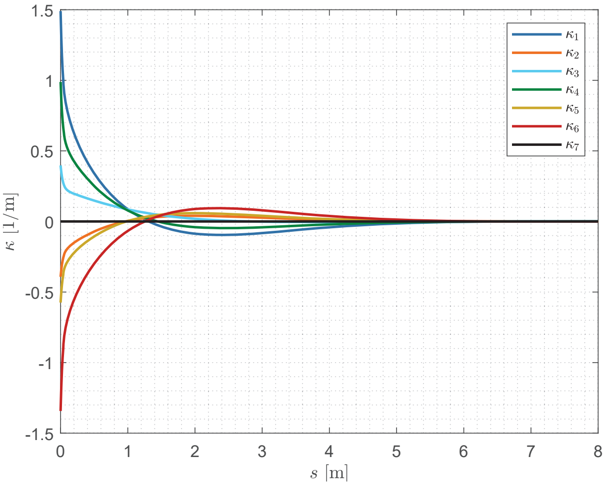
(a) Curvature $\kappa(s)$. (b) Lyapunov functions.

Figure C.1: Additional plots backward simulation with straight reference.

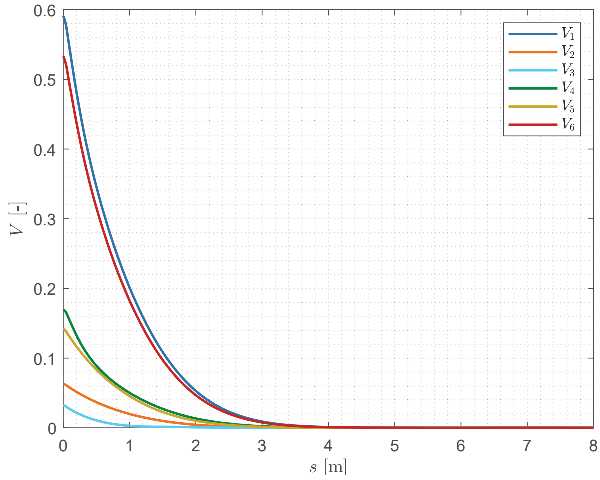


(a) Path overview. (b) Error traces.

Figure C.2: Controller validation for forward driving on straight reference path.

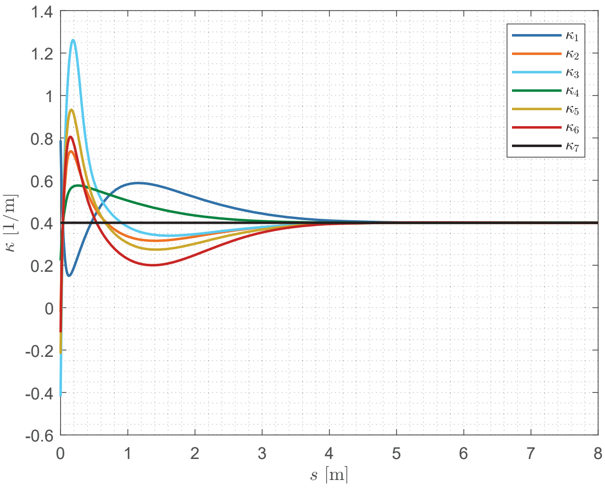


(a) Curvature $\kappa(s)$.

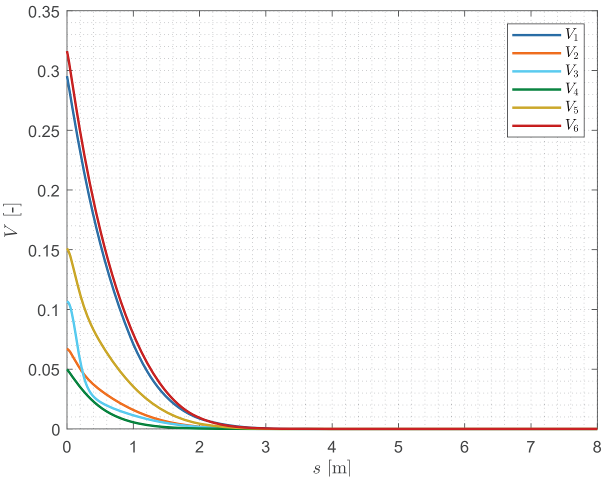


(b) Lyapunov functions.

Figure C.3: Additional plots forward simulation with straight reference.

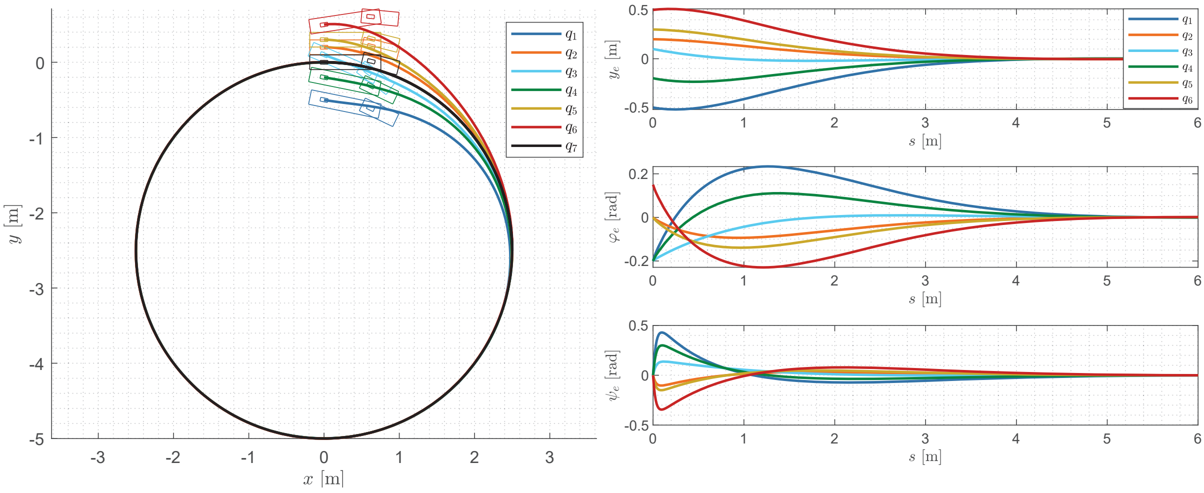


(a) Curvature $\kappa(s)$.



(b) Lyapunov functions.

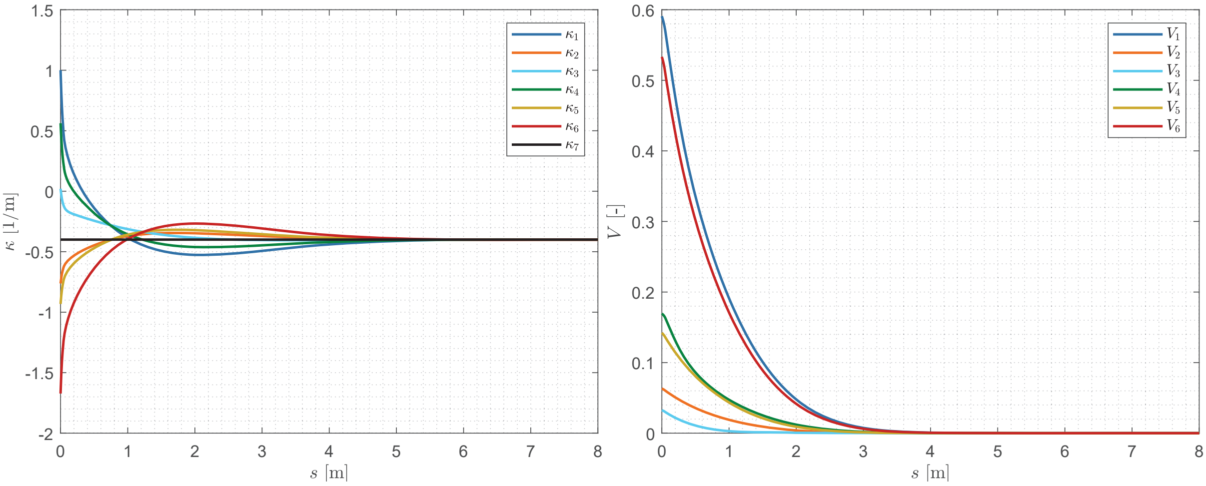
Figure C.4: Additional plots backward simulation with circular reference.



(a) Path overview.

(b) Error traces.

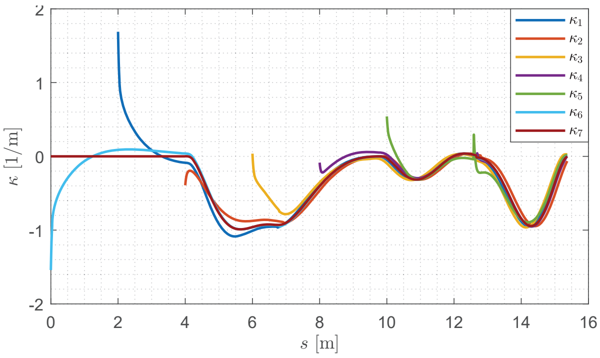
Figure C.5: Controller validation for forward driving on circular reference path.



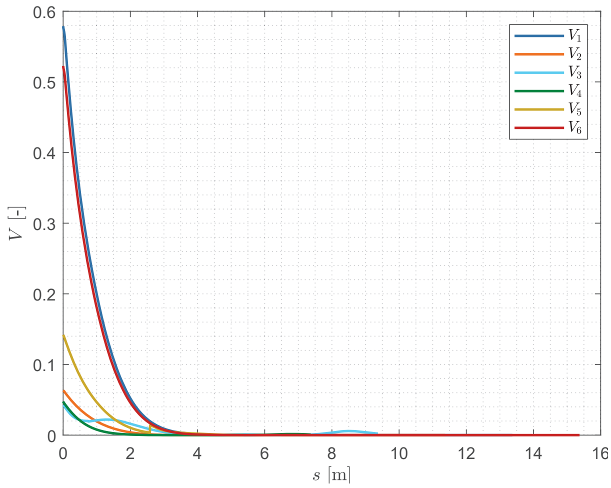
(a) Curvature $\kappa(s)$.

(b) Lyapunov functions.

Figure C.6: Additional plots forward simulation with straight reference.



(a) Curvature $\kappa(s)$.



(b) Lyapunov functions.

Figure C.7: Additional plots TruckLab scenario.

TU/e Technische Universiteit **Eindhoven** University of Technology

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

24-06-2021

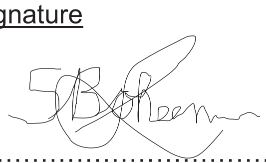
Name

Stan van Boheemen

ID-number

0958907

Signature



Submit the signed declaration to the student administration of your department.

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU