Spline-based Trajectory Generation for Autonomous Truck-Trailer Vehicles in Low Speed Highway Scenarios

A. Al Jawahiri

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Spline-based Trajectory Generation for Autonomous Truck-Trailer Vehicles in Low Speed Highway Scenarios

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

A. Al Jawahiri

May 17, 2018

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

Trajectory generators are designed to create reference paths that are parametrized by time as well as by velocity or acceleration. In the literature, it has been noted that often path planners are the last step in the planning module of autonomous truck-trailer vehicles, which results in a collision-free track. A trajectory generator can improve this path by making it smoother, which improves passenger comfort and decreases stress on the engine. The available trajectory generators are in their majority based on large number of optimizations, which limits their real-time applicability. This research aims to create a computationally efficient parametric method for trajectory generation that leads to a continuous motion of the vehicle. Using cubic splines, a unique method of parametrization is applied such that a custom velocity profile is defined along the trajectory. Together with choosing appropriate boundary conditions for each path segment, a trajectory is generated that proves continuous reference tangent angle and yaw rate, while following a velocity profile that is continuous in acceleration. This leads to a continuous longitudinal and lateral motion of the truck-trailer vehicle. While continuity conditions are achieved, simulations results showed high reference yaw rate peaks, which could be difficult to track under non-ideal circumstances. An alternative generator was designed based on quintic splines, such that yaw rate peaks can be limited. However, this minimization leads to oscillations of the reference trajectory. A trade-off needs to be defined to balance between minimizing peak yaw rates and oscillations. Closed-loop simulations using a kinematic truck-trailer model show that tracking the generated trajectory lead to continuous steering wheel and velocity input signals. Furthermore, the trajectory was smoothly tracked without the use of an optimal controller. Eventually all research goals have been achieved. Further studies could analyse the performance of the proposed trajectory generator using a dynamic truck-trailer model such that effects of wheel slip and load transfer can be investigated. It is also of interest to analyse the trailer behaviour when tracking the trajectory.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my daily supervisor Arturo Tejada Ruiz for his continuous assistance during the writing of this thesis, and constant positive attitude during the progress. Furthermore, I would like to express my appreciation to Dragan Kostic for introducing me to the problem of trajectory generators and guiding me through the start of this project. Finally, my warmest gratitude to my family, who have supported me at all times.

Delft, University of Technology
May 17, 2018

A. Al Jawahiri

Chapter 1

Introduction

In [1] it is stated that there has been a steady growth in road freight transport in the European Union (EU) since the financial crisis in 2008. This development caused a shortage of truck drivers. The EU provides a set of rules regarding driving time and rest periods for truck drivers. According to [2] the daily driving period cannot exceed nine hours a day. This is not only an economic limitation, but also one of the sources of accidents since tired drivers make poor decisions. In [3] it is stated that human errors contribute to accidents in 90% of all truck accident cases. Current studies consider using solutions such as automated driving and platooning trucks. This resolves the shortage of truck drivers, realizes almost 24 hours a day driving trucks, while removing human errors such that more safety can be accomplished. Even though much research has been done in the field of automated vehicles, an ideal solution has not yet been found. The key ingredient of automation is the planning module composed, among others parts, by the trajectory generator, which is the subject of this thesis. This planning module and its components will be explained next.

1-1 Background

This research adapts to the planning structure defined in [4]. It starts with the route planner, which is commonly known as the navigation. Along this planned route, a search will be conducted in the near environment of the vehicle. All objects and road boundaries are detected in order to determine the collision-free area. Using this information, the manoeuvre planner decides the next course of action based on criteria such as risk, comfort, fuel efficiency, or other demands. For example, should the vehicle perform an overtaking manoeuvre, or simply stay on the current lane. Based on this choice, a collision-free path is constructed by the path planner. However, this path could be suboptimal in terms of comfort, path continuity or length, or other criteria. Furthermore, no velocity profile is defined along this path. An optimised path is created by the trajectory generator, which is parametrized by time and is able to fulfil the desired specifications. A more detailed explanation of each planning element is described in the next subsections.

1-1-1 Route planner

The planning module starts with the route planner. This system finds the best global route from one starting position to a given destination, commonly known as navigation. It can be based on finding the optimal route with respect to distance, time, fuel efficiency or other criteria. Current technologies such as described in [5] can even take real-time traffic information into account and adjust the route on the fly based on this knowledge.

1-1-2 Search space

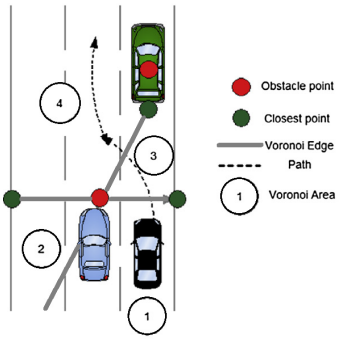
In order to plan a course for the autonomous vehicle such that collisions can be avoided and road boundaries are observed, a search of safe space must be performed. A few examples of search space methods are depicted in Figure 1-1. Voronoi diagrams (Figure 1-1a) calculate the distance between obstacles, and determine the collision-free space within these distances. Occupancy grids and cost maps (Figures 1-1b and 1-1c respectively) are probabilistic methods of space searching. The chances of an obstacle occupying the space is calculated, and a low-collision-risk-space is determined. Figure 1-1d shows the state lattice method, which creates a repetition of feasible paths from the vehicle. All paths that end up without collision are regarded as free space. Driving corridors (Figure 1-1e) represent a collision-free space, bounded by road and lane boundaries, where a vehicle is expected to move. A more detailed description of each search space methods is found in [4].

1-1-3 Manoeuvre planner

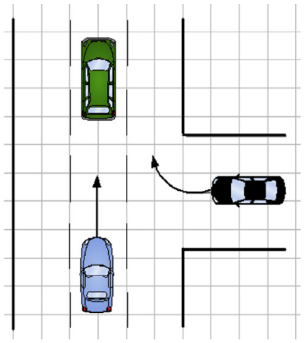
Manoeuvre planning can be seen as a decision planner. In [4] it is described that this system determines which high-level decision such as 'changing lane' or 'speeding up' the vehicle should take, based on risk, efficiency or vehicle limitations, while taking into account the collision-free space specified by the search space.

1-1-4 Path planner

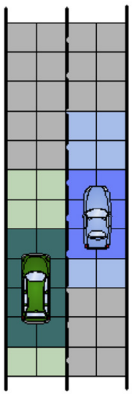
According to [4] is a path the geometric track that the vehicle should follow in order to reach its destination without colliding with obstacles. So path planning is determining a feasible path from an initial to a final configuration (position and heading direction) such that each state does not result in collision and adheres to the proposed motion constraints. The planner should conform to road and lane boundaries, as well as traffic rules. Often post-processing is applied in order to make the path smoother or reduce path length. Two common techniques for path planning are Rapidly exploring Random Tree (RRT) algorithms and Lattice Planners. In [6] it is described that the RRT algorithm builds a random space-filling tree, as is shown in Figure 1-2. Starting from a certain configuration, random collision-free and kinematically feasible line segments are constructed to fill the space until the desired configuration has been reached. Lattice Planners use the paths generated by the search space method State Lattice (see Figure 1-1d), which are already kinematically feasible and collision-free.



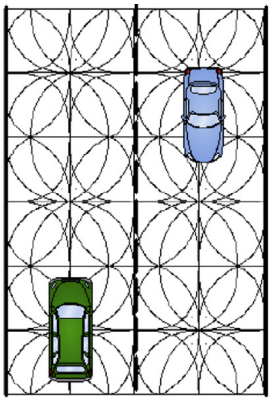
(a) Using Voronoi diagrams.



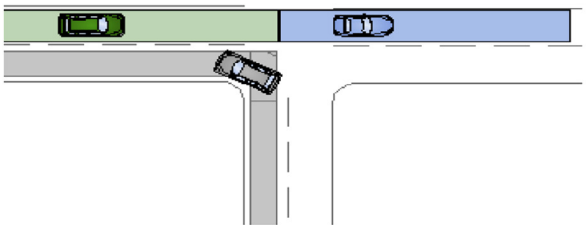
(b) Using occupancy grids.



(c) Using cost maps.



(d) Using state lattice.



(e) Using driving corridor.

Figure 1-1: Search space methods from [4].

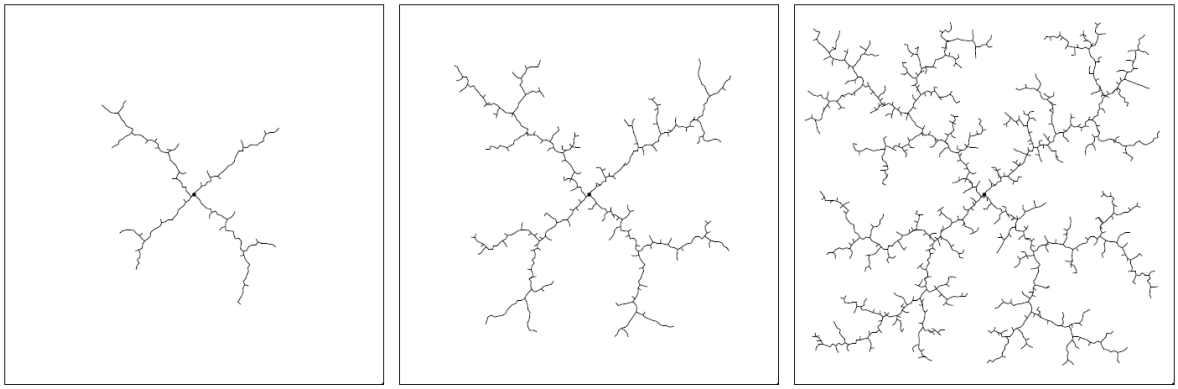


Figure 1-2: RRT search with increasing iterations [6].

1-1-5 Trajectory generator

According to [4] a trajectory is represented by a sequence of states visited by the vehicle, parametrized by time and, possibly, velocity or acceleration. Therefore, a trajectory generator does not only generate a geometric track, but could also define a reference velocity and an acceleration profile. For example, in a highway scenario, the trajectory planner is able to adjust the reference speed such that a safe distance can be kept from leading vehicles. Existing trajectory generators connect collision-free via-points given by the path planner such that acceleration or even jerk (time derivative of acceleration) are continuous and bounded. This leads to higher comfort levels and decreased stress on the vehicle engine. Some methods of trajectory generation include optimizing cost functions or utilizing graph searches, while other simply use higher order polynomials to connect the collision-free via-points determined by the path planner.

1-2 Thesis objective

It is noted that often path planners are the last step of the planning module. For example, in [7], an optimisation-based path planner creates a collision-free path from the starting to target configuration for truck-trailer vehicles. However, the computed geometric track is suboptimal in terms of continuity, comfort levels or other criteria. Many of these demands can be satisfied using trajectory generators.

Several trajectory generators have been designed by other researchers, and are discussed thoroughly in Chapter 3. Studies such as in [8–14] use exhaustive graph searches or large number of optimizations each sample, which demand high computational power of the on-board computers. In [14–17] trajectory generators are designed such that the path is continuous in position only, and discontinuous in higher order derivatives. Some of these solutions yield a trajectory with suddenly changing heading angle, which are difficult to track in practice by a vehicle. Moreover, abrupt changes in forward or angular velocity could cause discomfort of the vehicle passengers and stress on the engine.

To implement the trajectory generator in practice, practical criteria such as computational efficiency and certainty of optimality should be achieved in addition to the mathematical

requirements described above. The objective of this thesis is to select and design a method for trajectory generation that is suitable for truck-trailer vehicles and complies with the following practical specifications:

- The generator must be able to compute trajectories online. Travelling on the road is never the same twice. There might be an obstacle that must be avoided, or simply changing the lane to overtake another vehicle. Therefore each trajectory must be computed for the short-term future while on the road.
- The generated reference trajectory should eventually lead to a continuous movement of the vehicle in both longitudinal and lateral direction. This implies that not only the velocity profile should be continuous, but also the geometric shape of the trajectory.
- The generated trajectory should not be based on exhaustive searches or optimizations, and continuity should be proven. This makes sure that desired continuity specifications are achieved while being computationally efficient.

It is of particular interest to design the trajectory generator for truck-trailer vehicles. First of all, not much research has been done on trajectory generation for these type of vehicles. Even though a complete dynamic model represents a real-life truck-trailer vehicle most accurate, a kinematic model gives the opportunity to analyse simpler behaviours. This consists of continuity of the vehicle heading angle or the swaying behaviour of the trailer. It must be noted that a kinematic vehicle model only represents a real-life truck-trailer when driving with low speeds and small cornering manoeuvres.

With the objectives defined, the aim of this thesis can be defined as followed:

Design and implement a online parametric trajectory generator for truck-trailer vehicles that lead to a continuous longitudinal and lateral movement.

1-3 Thesis outline

The remainder of this thesis is organised as follows:

- **Chapter 2** presents the equations of motion for the kinematic truck-trailer vehicle model.
- **Chapter 3** presents state-of-the-art trajectory planning techniques and compares them based on the defined criteria.
- **Chapter 4** explains the procedure of generating a trajectory using cubic splines.
- **Chapter 5** introduces quintic splines as a method of trajectory generation such that the reference yaw rate is minimized.
- **Chapter 6** reports and evaluates results obtained in closed-loop simulation experiments of the kinematic truck-trailer model tracking the trajectory.
- **Chapter 7** revisits the research done in this thesis and suggests future work proposals.

Chapter 2

Vehicle modelling

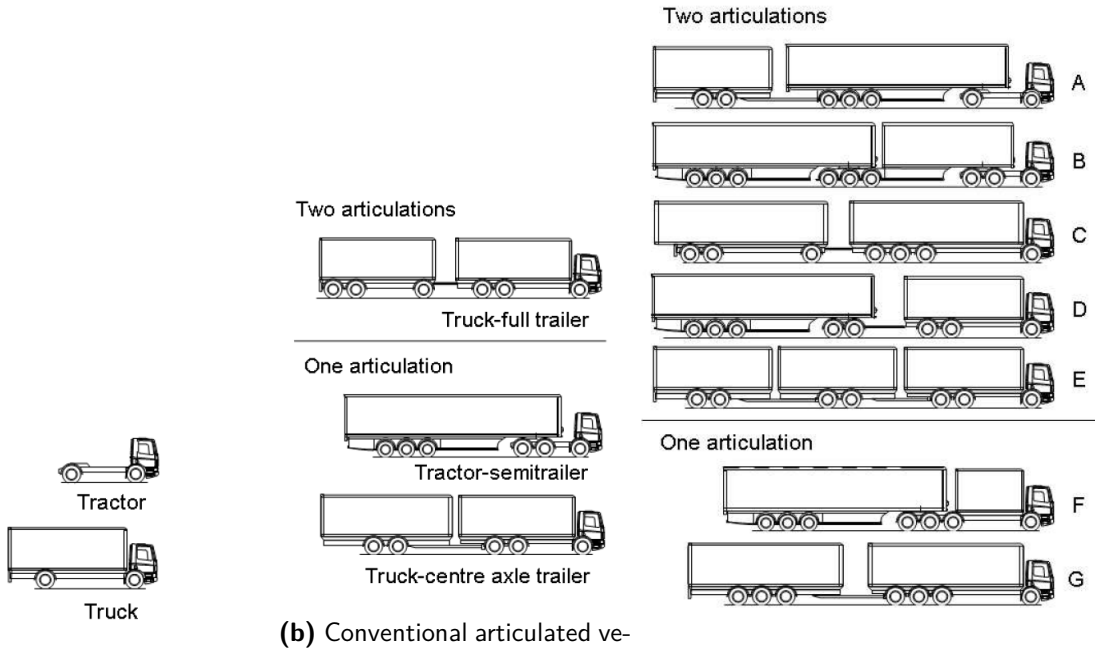
There are many different types of truck and trailer models, so a choice in vehicle type must be made first. Once a choice has been decided, a proper vehicle model will be derived. In this thesis a kinematic truck-trailer vehicle model is used to get an adequate representation of a real vehicle driving with lows speeds performing small cornering manoeuvres. A more extensive model would also contain vehicle dynamics, with for example load transfer and tyre slip. This chapter ends with a short discussion with the limitations of this model, and in what way it is used in this research.

2-1 Vehicle types

Figure 2-1 shows all truck and trailer combinations allowed in European Union countries according to [18]. The most important differences between the vehicles are the towing method and the number of articulations. Trailers can be towed by either a tractor or truck. The difference between these two is the fact that tractors do not carry a load by themselves, but pull trailers or semi-trailers with a load. However, trucks do carry a load and possibly also pull a semi-trailer or trailer. In some cases, when transporting larger freights, multiple trailers are articulated together (as is seen in Figure 2-1c). As can be noted from this figure, the maximum number of articulations in the EU is two. A tractor can pull a semi-trailer and a trailer or two semi-trailers (type A and B respectively in Figure 2-1c). A truck can pull a single trailer, a single semi-trailer, or two trailers (type C, D and E respectively in Figure 2-1c).

Since not much research is found on modelling of truck-trailer vehicles, a simple and most common type will be used as a basis. From there on, future studies can extend the model to more complex vehicle structures. In [18] it is stated that the most common found truck combinations in The Netherlands are the conventional articulated vehicles in Figure 2-1b. It is wise to start with one articulation, so the tractor-semitrailer and the truck-centre axle trailer are the only possible choices. Besides body roll dynamics, the only difference between modelling those two types is the hitch position (contact point between trailer and truck or

tractor). This point is located in front of the rear wheels of the tractor for the tractor-semitrailer, and behind the rear wheels of the truck for the truck-centre axle trailer. For simplicity, these two types of vehicles are called truck-trailers for the remainder of this thesis.



(a) Single vehicles.

(b) Conventional articulated vehicles.

(c) Ecocombi's

Figure 2-1: Truck combinations from [18].

2-2 Kinematic model

The next step consist of deriving the equations of motion for the simplified truck-trailer vehicle model. First, some assumptions should be defined that describe this kinematic model.

- The model contains no mass (and thus no load transfer).
- No rolling or pitching.
- No tyre slip is present.
- Resistive forces are negligible.
- The left and right wheels are assumed to be in the centre such that it represents a bicycle model.
- Each set of axles is assumed to be a single axis for both the truck and the (semi-)trailer.
- The trailer passively follows the truck.
- The vehicle moves on a flat surface.

Based on these assumptions, the model may seem to be a unrealistic representation of a real truck-trailer vehicle. However, the main reason behind using a kinematic model is that the non-holonomic vehicle constraint is integrated. This constraint describes the fact that trucks only have two control inputs (steering angle and engine torque), while its configuration space is three (lateral and longitudinal coordinates, and vehicle heading angle). This constraint implies that vehicles cannot move in a lateral motion instantaneously, but by using an appropriate combination of the available inputs. An example is a parallel parking operation, which is achievable by a combination of steering and applied torque.

Figure 2-2 shows the kinematic model of the simplified truck-trailer vehicle on the earth-fixed coordinate system (x,y). The vertically patterned vehicle represents the truck, and the horizontally patterned vehicle is the trailer. The two vehicles are joined at the hinge point, here located at the truck rear axle. Notice that wheels are assumed to be in the centre of the vehicle track that is, the vehicle kinematics are represented by a bicycle model. The model consists of the states $(x_v,y_v,\psi,\theta)^T$. x_v and y_v are coordinates of the centre of mass of the truck on the earth-fixed coordinate system. Since it is assumed that there is no load transfer, this centre of mass is always at fixed point on the truck. ψ is the relative angle between the truck heading direction and the fixed x-axis, and θ is the relative angle between the trailer heading direction and the x-axis. Obviously, inputs are required in order to control the state to desired values. The two inputs are $(\delta,v)^T$. In this case, δ is the angle between the vehicle direction and steering wheel direction, also called the steering angle. The second input is velocity v, which is in the direction of the truck front wheels.

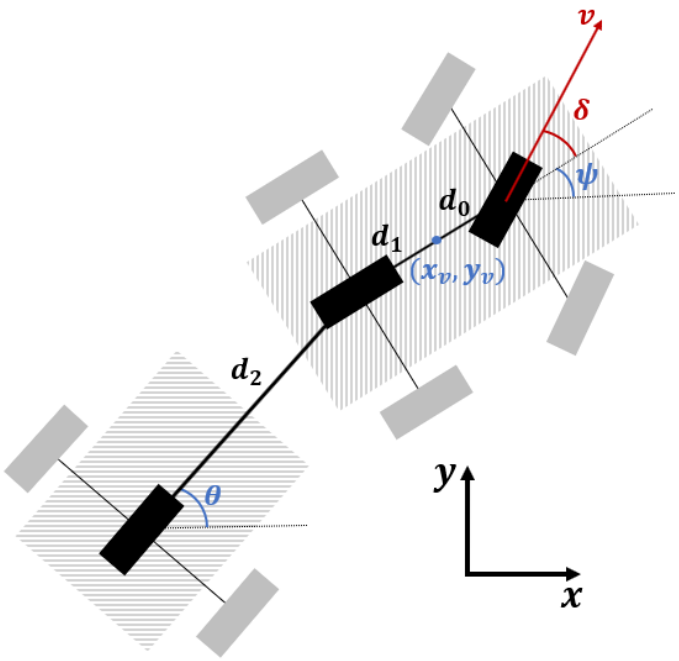


Figure 2-2: Kinematic truck-trailer bicycle model.

With the state and inputs defined, the equations of motion can be derived. Using regular vehicle bicycle kinematic model from [19] (without slip angle) and trailer kinematic model from [8], the following equations of motion are found for a kinematic truck-trailer model:

$$\begin{cases} \dot{x}_1 = v\cos(\psi) \\ \dot{y}_1 = v\sin(\psi) \\ \dot{\psi} = \frac{v}{d_0+d_1}\tan(\delta) \\ \dot{\theta} = \frac{v}{d_2}\sin(\psi-\theta), \end{cases} \tag{2-1}$$

where d_0, d_1, and d_2 are, respectively, the distance of the truck's centre of mass to its front and rear axles and the distance between the truck's rear axle and the trailer's axle. As can be noted from these equations of motion, angle θ is independent of the steering input δ, thus the trailer is passively following the truck.

2-3 Discussion

As mentioned before, this model only takes into account the basic kinematic behaviour of the truck-trailer combination. Obviously, more detailed models are available in case a more in depth analysis of the vehicle behaviours is desired. That, however, is outside the scope of this thesis, because the main focus is to generate the trajectory, and not to track the trajectory. This model is complete enough to see if a truck-trailer vehicle can kinematically follow the generated trajectory, based on the non-holonomic constraint for vehicles. If it is shown that this is possible, a more complex model could be designed with dynamics included. Using this extended model, the trajectory can be evaluated based on dynamic aspects such as load transfer or wheel slip. This evaluation is possible in future studies, also discussed in Chapter 7.

Furthermore, starting with an initially simple model gives the opportunity to gradually increase the complexity of the model. As is said before, this model shows if the generated trajectory adheres to the non-holonomic constraint. Extending the model by adding mass to the equations of motion can give simulation results regarding the momentum of the vehicle when following a trajectory. Further extending the model to a two-track model instead of a bicycle model shows the separate behaviour between left and right wheels. The point is, if one would have started simulation experiments with a very extensive vehicle model, it may be difficult to analyse where the problems are when trajectory tracking errors are obtained. With the kinematic model used in this research, tracking errors in simulations are most probably caused by the fact that the non-holonomic constraint is not adhered. Once no problems are found using this model, it can be extended by one system at a time. Analysing simulation results using this extended model shows possible faults in the added system (such a load transfer). This systematic approach to vehicle modelling may be time consuming, but it does give the possibility to analyse each separate vehicle system.

Eventually it is of interests how a truck behaves following a generated trajectory. Chapter 6 shows how this vehicle model tracks a generated trajectory in a close-loop simulation at low speeds with small cornering manoeuvres. This opens the opportunity to see how a vehicle behaves following the trajectory. For instance, analysis can be applied to the trailer trajectory to see how much it will sway, or how fast the truck heading angle changes over time.

Chapter 3

State-of-the-art trajectory generators

In the introduction is has been explained that the path planner creates a collision-free geometric path. By processing via-points on this collision-free path, trajectory generators define reference velocities, accelerations, or jerks along the path, while preserving the continuous geometric path properties.

A number of methods for trajectory generation used by other researchers are reviewed in this chapter, including a comparison of their advantages and disadvantages. As is mentioned in the introduction, the trajectory generator must meet the following requirements:

- The generator computes trajectories online.
- It should be parametrically continuous by defining continuous reference velocity and acceleration profiles along the path. This is known as C^2 (See Appendix A).
- It should be geometrically continuous in terms of position, heading direction, and yaw rate. This is known as G^2 (See Appendix A).
- It should not be based on exhaustive searches and cost function optimizations, and continuity should be proven.

Furthermore, computational efficiency is preferred. Based on these specifications, a choice must be made. This choice will be discussed in section 3-5. An overview of all discussed trajectory generation methods is provided in Table 3-1.

3-1 Clothoids

Using clothoids, the generated trajectory has a linearly changing curvature (reciprocal of radius). According to [20] the main advantage of clothoids is the fact that G^2 continuity is achieved. However, G^2 continuity does not ensure C^2 continuity (see Appendix A). In other

words, the reference acceleration computed by this trajectory generator is not necessarily continuous.

In both [21] and [20] it is claimed that with clothoids low computational complexity is realized, and thus real-time generation is feasible. Furthermore, in [21] clothoids are implemented into a real vehicle, in which kinematically feasible trajectories were realisable. However, it must be said that a rescheduling time of 100 ms was required, which is relatively high for real-time applications. This may be caused by the fact that at each sample a number of possible trajectories are computed and an exhaustive analysis of all these trajectories must be performed. In [21] it is also shown that a weight function can be used to improve navigation comfort at the expense of travelling time.

3-2 Polynomials

Polynomials are used in [9–12, 17] to define a feasable trajectory. The problem of computing trajectories through n+1 via-points can be solved by the fitting polynomial functions of degree n:

$$\begin{aligned} x_{\text{ref}}(t) &= a_0 + a_1 t + ... + a_n t^n \\ y_{\text{ref}}(t) &= b_0 + b_1 t + ... + b_n t^n. \end{aligned} \tag{3-1}$$

Here, $x_{\text{ref}}(t)$ and $y_{\text{ref}}(t)$ define the reference global coordinates as a function of time, and the scalar coefficients $a_0, \ldots, a_n$ and $b_0, \ldots, b_n$ must be found such that the trajectory interpolates all via-points. Several methods have been reported to generate trajectories using second to fifth order polynomials, as summarized next.

3-2-1 Second order polynomials

In [17] a custom algorithm (named TiJi) was designed to optimize a second order polynomial trajectory that reaches a goal configuration at a prescribed final time. The trajectory generator is able to recognize if a goal is unreachable. If this is the case, it returns a trajectory ending as close as possible to the goal.

It must be said that this paper hardly mentions comfort as a design specification. All that is applied in terms of comfort are acceleration bounds, but not necessarily continuity of acceleration. The research also mentions the fact that the computation time increases significantly when the number of required steps to the goal state increases. Furthermore, it is known that the second order time derivative of these polynomials are constant, which proves only C^1 continuity and not C^2 as needed. Constant accelerations result in discontinuous acceleration when two path segments join. The study does not mention geometric continuity.

3-2-2 Third order polynomials

Third order polynomials have been used in [9] to generate trajectories for special manoeuvres such as following a leading vehicle, performing lane changes and merging between other vehicles. All generated manoeuvres are kinematically feasible, and are implemented in a real-time vehicle.

However, it seems that the used acceleration profile can be improved, since it is assumed to be constant in this research. Constant accelerations lead to execution errors, since vehicle cannot change acceleration abruptly. This also causes discomfort for the vehicle passengers. Once again, the geometric continuity is unknown. Furthermore, an exhaustive search is required to find the optimal trajectory, since the reasoning is based on a cost-function approach.

3-2-3 Fourth order polynomials

In both [10] and [11] fourth order polynomials are used to generate trajectories. However, different design specifications are assumed in the two studies. In [10] a set of cost functions for each generated trajectory is considered to evaluate safety, comfort, efficiency, energy consumption and behaviour. In the end, most of those trajectories will be discarded, which could be seen as a waste of computational power. However, the authors claim that the post-optimization is of low computational complexity and is able to generate a solution within a few iterations. Furthermore, the study mentions implementing the system in a real autonomous vehicle in three scenarios: lane driving on S-shaped road, lane driving with static obstacles, and lane driving with moving obstacles. The authors claim that the generated velocity profile would always give a continuous acceleration, but they have given no results regarding the acceleration of the real vehicle performing the three manoeuvres.

In [11] constraints are set on maximum acceleration and forces in order to generate a feasible trajectory. Furthermore, boundary conditions are applied for the position and velocity at the begin and final state of the road segment. This means there is continuity in position and velocity, but not necessarily in acceleration. From their simulation results it can be noted that the longitudinal acceleration profile is indeed jerky.

3-2-4 Fifth order polynomials

Fifth order polynomials are used in [12]. The authors combined different lateral and longitudinal cost functions for different tasks such as following, merging and stopping. According to them they have succeeded in mimicking human-like driving behaviour on the highway, and have proven that the optimal solution, using quintic polynomials, has a continuous acceleration.

However, as is mentioned in [10], the curvature of every point on each trajectory needs to be computed and verified, which is computationally inefficient. Furthermore, even though the road curvature is continuous, the rate of change of this curvature changes very frequently from positive to negative values, which leads to jerky steering wheel movement and thus lateral discomfort. Also, it can be noted that in [12] a set of heuristics are used for manoeuvres such as following and merging. According to [22] it is unrealistic to find a complete set of heuristics that are applicable in all cases. Heuristics do not prove optimality and are therefore unreliable.

3-3 Spline curves

Splines are functions defined piecewise by polynomials. They use separate polynomial functions to define each segment between via-points of the trajectory. In [23] it is stated that when

n+1 via-points are given, instead of interpolating with one polynomial of degree n (such as in Section 3-2), it is possible to use n splines of degree p to construct the trajectory between each segment. Depending on the desired degree of continuity, a proper value of p can be chosen. The spline function is described by equation (3-2) and is shown in Figure 3-1.

$$\begin{aligned} s(t) =& \{q_k(t), t \in [t_k, t_{k+1}], k = 0, ..., n-1\}, \\ q_k(t) =& \begin{bmatrix} x_k(t) \\ y_k(t) \end{bmatrix} = \begin{bmatrix} a_{k,0} + a_{k,1}(t-t_k) + ... + a_{k,p}(t-t_k)^p \\ b_{k,0} + b_{k,1}(t-t_k) + ... + b_{k,p}(t-t_k)^p \end{bmatrix}. \end{aligned} \tag{3-2}$$

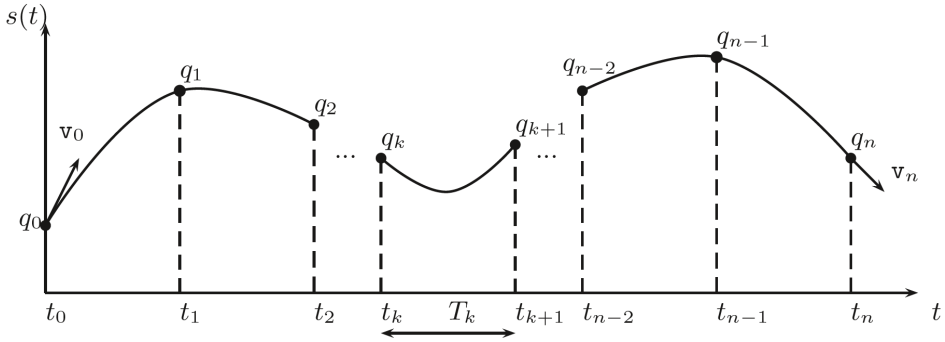


Figure 3-1: Spline trajectory through n+1 points [23].

Here, $s(t)$ is the spline, $q_k(t)$ the polynomial between two via-points and k the via-point number. From Equation (3-2) it can be noted that the spline s is constructed of n polynomials, with each a degree of p. The following sections will discuss trajectory generators for vehicles based on cubic ($p = 3$), quintic ($p = 5$), η^3 ($p = 7$), η^4 ($p = 9$) and Akima splines as reported in the literature.

3-3-1 Cubic splines

Cubic splines are continuous in path, velocity and acceleration if the boundary conditions are well defined. Since the second order time derivative of a third order polynomial is a linear function, each segment is continuous in acceleration too and thus C^2 continuity is achieved. Obviously this is also the case for higher order splines. However, if the boundary conditions of a segment are ill-chosen, the velocity or acceleration at a via-point exactly between two segments could be discontinuous. Also note that depending on the choice of parametrization, G^2 continuity is achievable (see also Appendix A).

In [24] and [25] cubic splines are used to generate trajectories. According to [25] kinematically feasible trajectories are realizable using cubic splines, and the computation is efficient enough for real-time implementation. Furthermore, the proposed method allows more aggressive driving style without compromising safety. Also, it is mentioned that splines curves are computationally efficient.

On the other hand, it has been noted in [26] that a trajectory consisting of cubic splines could lead to large oscillations as via-points move close to each other. This effect is illustrated in Figure 3-2. This is the case for all parametric trajectories such as polynomials and splines.

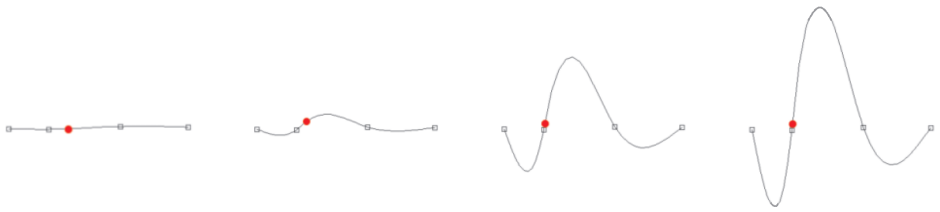


Figure 3-2: Curve fitting of cubic splines under noisy conditions, showing the sensitivity of the fitting to the noise.

3-3-2 Quintic splines

In [27] quintic splines are used to set boundary conditions for position, velocity, acceleration and time. Thus not only C^2 is achieved, but also time goals are generated for the vehicle. In both [27] and [28] it is also mentioned that the non-holonomic constraint of the vehicle is adhered. However, high maximum jerks are experienced in simulation results by [28], which is not ideal. The authors did implement a system to minimize peak magnitude of the jerk by means of optimization, and this resulted in a continuous jerk with reasonable peaks.

3-3-3 η^3 splines

η^3 splines (seventh order) are used in [29] to generate trajectories with a third order geometric continuity (see Appendix A regarding continuity). Since it is a seventh order spline, sixteen coefficients must be found for each segment (see Equation (3-2), coefficients $a_0, \ldots, a_7$ and $b_0, \ldots, b_7$). These coefficients are found by expressions which are a functions of η_i, with i=1, 2, ..., 6. The η-parameters can be freely chosen without violating the boundary conditions (position, velocity, acceleration) set on the via-points. The main advantage of η^3 splines is its ability to generate a variety of trajectories such as spirals and arcs among others.

Since many parameters have to be found for each segment, with extensive expressions to solve, it can be assumed that this method is computationally expensive and thus not realizable to implement in real-time vehicles.

3-3-4 η^4 splines

Ninth order splines, here called η^4 splines, are used in [8] to generate trajectories for truck-trailer vehicles in parking manoeuvres. According to the authors ninth order splines are G^4 continuous (See Appendix A). It was proven that the proposed method yielded a trailer trajectory that is feasible. According to the authors, a trailer trajectory is feasible if and only if it is G^4. Using η^4 splines, unconventional planning problems such as parking manoeuvres can also be addressed.

Because these are ninth order splines, twenty coefficients must be found ($a_0, \ldots, a_9$ and $b_0, \ldots, b_9$), which each are a function of η_i with i=1, 2, ..., 8. The twenty parameters are found by expressions shown in [8], which are obtained by means of tuning. Just like the η^3 splines, an optimization must be performed in order to search for ideal values of η_i. Furthermore, since the expressions required to obtain $a_0, \ldots, a_9$ and $b_0, \ldots, b_9$ are even more extensive than with η^3 splines, the computational effort will be even bigger.

3-3-5 Akima splines

Akima splines are used in [13]. They are quite similar to cubic splines, but are less affected by outlying via-points. In this method, each polynomial of the spline is determined by the coordinates of and the slope at the via-points. As shown in Appendix A this is equal to G^1 continuity. The generator requires low computational costs and has been tested in real vehicles on public roads. There is no doubt that the system is able to run online, because the method is tested in their vehicle over several thousand kilometres driving autonomously.

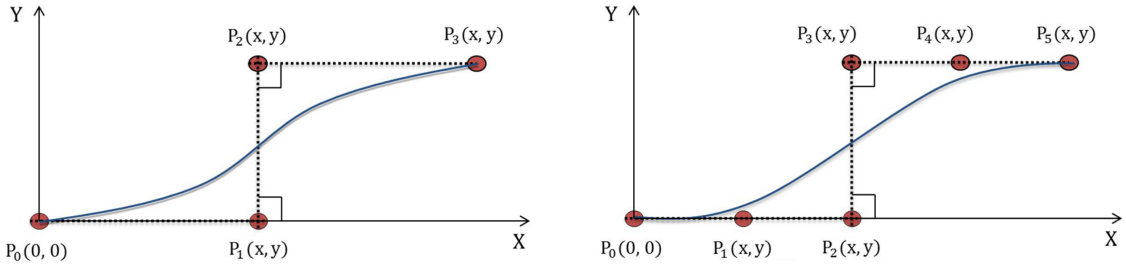
However, it seems that [13] has not given any simulation or practical results besides the fact that the system is able generate trajectories for multiple manoeuvres. It is unknown if this method yields continuous acceleration (C^2), or only continuous paths and its curvature (G^1). Furthermore, at each sample multiple possible trajectories are generated, and the optimal solution is chosen regarding current manoeuvre, safety and comfort. This choice is done by a set of heuristics, which are hard to define accurately.

3-4 Bezier curves

Bezier curves are parametric curves which create trajectories using P_n points, where n denotes the Bezier order. Equation (3-3) shows the general form for a Bezier curve.

$$B(t) = P_0(1-t)^n + 3P_1t(1-t)^{n-1} + ... + 3P_{n-1}t^{n-1}(1-t) + P_nt^n \quad , t \in [0,1] \tag{3-3}$$

Examples of third and fifth order Bezier curves are shown in Figure 3-3. Here P_0 is the initial position, whereas P_3 in Figure 3-3a and P_5 in Figure 3-3b are the final positions. The intermediate points P_1 to P_{n-1} (also called control points) are used to define the curve. For example, the control points in Figure 3-3 represent the road boundaries in a lane changing manoeuvre. Vehicle width and other specifications could also be taken into consideration when choosing the control points.



(a) Third order Bezier curve. **(b)** Fifth order Bezier curve.

Figure 3-3: Trajectory generation using Bezier curves [30].

Existing researches present third, fourth or fifth order Bezier curves to define the trajectories. Below is a short discussion regarding each of these methods, and their advantages and disadvantages regarding the specifications defined at the start of this chapter.

3-4-1 Third order Bezier curves

Third order Bezier curves are used in [14, 31–33]. The first three studies used explicitly third order Bezier curves, while in [14] third order Bezier curves are used in combination with fourth order.

In [32] it is stated that the main advantages of third order Bezier curves are their simplicity, curvature continuity, and computational efficiency. First of all, only four points P are required to generate the curve (see Figure 3-3a), and each of those points can be easily obtained for any manoeuvre. Second, curvature continuity is in position only, but not in velocity nor acceleration. Even though the research claims to aim for a comfortable driving experience, a discontinuity in velocity and acceleration would yield just the opposite. The third advantage (computational effort) is achieved because the generation is not based on exhaustive searches or optimizations, but by solving analytical expressions.

From the simulation results in [32] it seems that the steering wheel output was jerky and a lateral offset was obtained. This could be caused by the trajectory tracker (controller), but the fact that neither C^2 nor G^2 continuity is proven could also be responsible for this undesirable behaviour. In [30] it is stated that third order Bezier curves are inappropriate to generate a desired trajectory for a lane change manoeuvre. This method is unable to perform a continuous transition from driving in a straight line to starting a lane change, thus the trajectory is geometrically discontinuous at the point where the two manoeuvres join.

3-4-2 Fourth order Bezier curves

In [14] fourth order Bezier curves are utilized if the third order does not meet the curvature requirements. The requirements are based on maximum curvature realizable by the vehicle, road constraints and other specifications. However, C^2 continuity is still not guaranteed with fourth order Bezier curves. In order to compensate, a custom speed profile is generated next to the trajectory. This profile is based on comfort levels in lateral, longitudinal and vertical acceleration (each with custom weighing factors), and the curvature of the evaluated segment. According to the simulation results the speed is profile linear, which means the acceleration is constant along each segment and discontinuous when two segments join.

3-4-3 Fifth order Bezier curves

As was mentioned in subsection 3-4-1, in [30] it is claimed that third order Bezier curves are unable to create a continuous transition from driving straight to a lane change maneouvre. However, the authors mention that fifth order Bezier curves are capable of doing this, since the initial and final vehicle heading angle of the fifth order Bezier curve can be zero. This way a continuous transition is realizable. This G^1 continuity is demonstrated in simulation results, in which third order Bezier curves are compared to fifth order. As was expected, the fifth order gave much more desirable results, such as a smooth steering angle, lower steering rate, and smaller maximum curve radius.

While the curve from driving straight to an overtaking manoeuvre is continuous using fifth order, the velocity and acceleration are not necessarily continuous, thus not satisfying the C^2 continuity requirement.

3-5 Discussion

As mentioned in the introduction, the goal of this thesis is to develop a trajectory generator that satisfies the requirements mentioned at the start of this chapter. Such generator can be developed by modifying one of the methods reviewed in this chapter. While none of the afore-described methods satisfy all the requirements, a reasonable choice can be made which satisfies the design specifications up to a certain degree. Table 3-1 gives an overview of all trajectory generation methods and their specifications regarding the requirements.

Clothoids are G^2 continuous. The geometric path may seem smooth, but is not necessarily C^2 continuous. Furthermore, the sources using this method required exhaustive searches in order to obtain an optimal trajectory. The design specifications for this research clearly state that the generator should not rely on searches.

Polynomials do not satisfy the online trajectory generation requirement. This method requires multiple via-points along the route in order to generate one polynomial that interpolates all these points. Unfortunately, a change in via-points would invalidate the generated trajectory. Such changes are common during vehicle operations due to the dynamic nature of the environment, which would limit the online use of trajectory generation based on polynomials. However, C^2 continuity is realisable using polynomials. Furthermore, the method is not based on searches or optimizations and is computationally cheap.

Bezier curves are easily computed and its computational effort is low. This method is useful if the trajectory generator has to account for road boundaries. However, in this research it is assumed that the via-points given by the path planning module are collision free and close enough to each other such that it is hardly possible for the trajectory to exceed the road boundaries. Furthermore, the C^2 continuity condition is not proven using third, fourth, or fifth order Bezier curves.

Splines are based on polynomials generated separately for each segment between via-points. It is known that polynomials can be C^2 continuous (by using third order polynomials or higher), so the design specification regarding continuity is satisfied. However, the spline boundary conditions must be chosen carefully in order to maintain both C^2 and G^2 continuity between two path segments. Splines could thus be suitable with respect to the design specifications. A choice must be made in the order of splines, which vary from third to ninth order in the investigated studies. Since third order splines satisfy all the design specifications already, it is unnecessary to use higher order splines. It only increases the complexity of the solution, while increasing computation time. For instance, if one requires a continuous jerk (third order derivative of path), fifth order splines should be used. It must be mentioned that in [8] it is stated that a trailer path is feasible if and only if the path is a G^4-path, which is realizable using η^4-splines. However, this includes trajectories such as parking manoeuvres, which are not the focus in this research. In this thesis the attention lies on highway scenarios with relatively simple manoeuvres such as lane changes. Furthermore, computationally heavy optimizations are required for η^4-splines.

Since Akima splines are a variation of cubic splines, these should also be considered. However, it seems the research using Akima splines requires a set of heuristics to choose the optimal trajectory. Since heuristics do not prove optimality and thus are unreliable, this method will not be further considered here. Furthermore, only G^1 continuity is achieved using Akima splines. Thus, out of all methods of trajectory generation found in the literature, cubic splines

satisfy the majority of the design specifications. The next chapter explains how this method was adapted for trajectory generation in this research.

Table 3-1: Overview of trajectory generation methods found in the literature, grouped by their core approach (clothoids, polynomials, splines, Bezier curves. See column 1). The second column shows whether the method is suitable for online trajectory generation. The third column describes the degree of parametric and geometric continuity of the trajectories generated by each method. The fourth column, 'Optimality', specifies whether an optimal solution is proven, or optimization is necessary. The last column shows remarks that could influence the selection of a method.

Method	**Online**	**Continuity**	**Optimality**	**Comments/ Assumptions**
Clothoids [20, 21]	Yes	G^2	Optimization required	-
Polynomials				
Second order [17]	No	C^1	Unknown	High computation time for long planning
Third order [9]	No	C^2	Exhaustive search	Constant acceleration
Fourth order [10, 11]	No	Unknown	Optimization required	Discontinuous acceleration profile
Fifth order [12]	No	C^3 and G^1	Heuristics and Optimization	Computationally inefficient
Splines				
Cubic [24, 25]	Yes	C^2 and G^2	Proven	Sensitive to outliers
Quintic [27, 28]	Yes	C^3	Proven	-
η^3 [29]	Yes	C^2 and G^3	Proven	Not applied to vehicles
η^4 [8]	Yes	C^2 and G^4	Optimization required	Computationally expensive
Akima [13]	Yes	G^1	Heuristics	Implemented in real vehicle
Bezier Curves				
Third order [14, 31–33]	Unknown	Discontinuous	Proven	Discontinuous at via-points
Fourth order [14]	Yes	C^1	Optimization required	-
Fifth order [30]	Unknown	G^1	Proven	Continuous steering rate

Chapter 4

Cubic splines

This chapter describes the method used to develop a trajectory generator based on cubic splines. First, the input information to the generator is explained, including a clear overview of the assumptions made in deriving the generator. Next, the procedure followed to derive the unknown spline coefficients is explained in detail. Here it is made clear that certain boundary conditions must be defined in order to reach the desired continuity specifications. This is followed by a description of the time-parameterization applied to the splines so a given velocity profile is defined along the trajectory. It is also shown that numerical integration is required to solve this parametrization. The last section discusses the benefits of trajectory generation using cubic splines, based on some preliminary results and describes possible limitations of the approach.

4-1 Input information

Recall from Chapter 1 that a trajectory is generated over the path generated by the path planner, which creates a collision free geometric track. Since this path is collision free, choosing several points on it should define a sequence of via-points that are still collision-free when interpolated. Figure 4-1 shows an example of a sequence of via-points on the earth-fixed coordinate system (x, y) that represent a Single Lane Change (SLC) manoeuvre. Starting from coordinates $(0, 1)$, the manoeuvre starts with straight driving in the x-direction, performs the lane change after a couple of via-points, and eventually ends up driving straight again. The following information is known about these via-points.

- The coordinates on the earth-fixed coordinates system (x, y)
- The desired velocity at each of the via-points

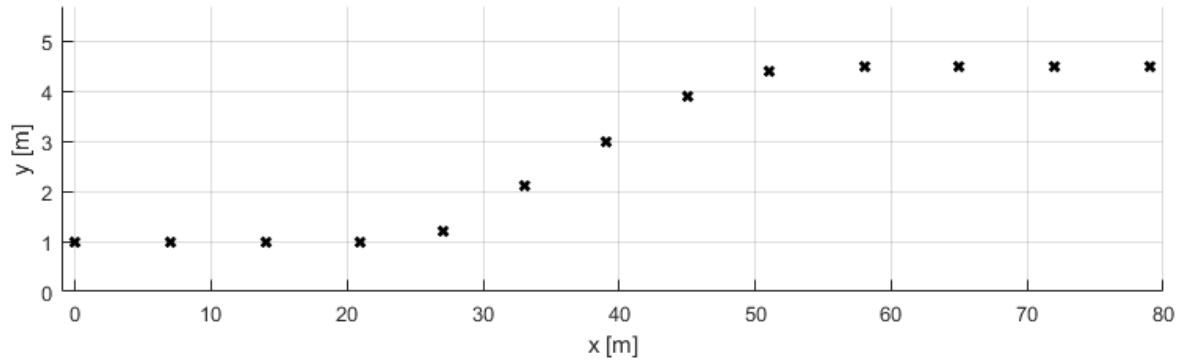


Figure 4-1: Sequence of via-points representing a single lane change manoeuvre.

In this research it is assumed that the manoeuvre planner (see Chapter 1) is able to set a desired velocity on each via-point. As an example, if the maximum allowable speed increases, the manoeuvre planner sets the desired velocity on each via-point such that this new maximum speed is gradually reached. And in the case of a SLC manoeuvre, one might want to increase the velocity slightly before taking over another vehicle. Figure 4-2 shows the desired velocity at each via-point for a possible SLC manoeuvre.

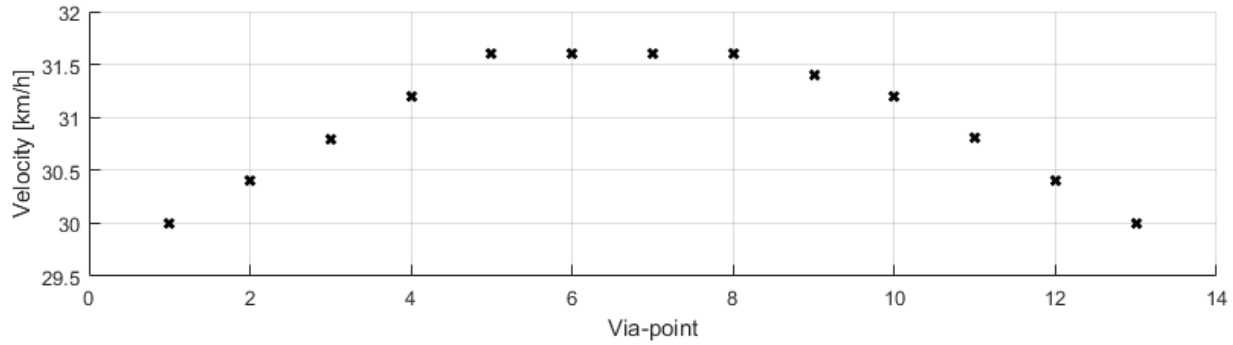


Figure 4-2: Desired velocity at each via-point.

Since one of the research goals is to design an online trajectory generator, not all future via-points can be used as inputs to the generator. In the case of Figure 4-1, if the vehicle is at coordinate $(0, 1)$, only the information (position and desired velocity) of this via-point is known, and the next via-point at $(7, 1)$. These points define what is called a segment. Once the vehicle arrives at this second via-point, information of the third point becomes available for use such that a trajectory can be generated for the second segment. As mentioned before, in this chapter the iterative trajectory generation is done by cubic splines. The procedure to compute the spline coefficients is described next.

4-2 Spline coefficients

Equation (3-2) in Chapter 3 shows the general form of spline equations. When customized for cubic splines, these equations become as follows:

$$x_{\text{ref},k}(t) = \quad a_{k,0} + a_{k,1}u + a_{k,2}u^2 + a_{k,3}u^3\big|_{u=u(t)} \tag{4-1}$$

$$y_{\text{ref},k}(t) = \quad b_{k,0} + b_{k,1}u + b_{k,2}u^2 + b_{k,3}u^3\big|_{u=u(t)}\,, \tag{4-2}$$

with subscript 'ref' implying reference, and k the current segment. Notice from Equations (4-1) and (4-2) that the splines are parametrized by a function of time, u. This parametrization will be explained further in Section 4-3. This section will explain how spline coefficients $a_0 \dots a_3$ and $b_0 \dots b_3$ are found for each segment. These coefficients are found by defining boundary conditions such that the trajectory is continuous in position, tangent direction and yaw rate (rate of change of path tangent direction). These continuity specifications will eventually contribute to a continuous steering rate and velocity input to the vehicle, and thus leading to a continuous reference lateral and longitudinal movement as is stated in the objective. This continuity becomes clear in Chapter 6, where the control inputs are computed.

For cubic splines, four boundary conditions should be set up in order to reach the desired continuity specifications. It is assumed that every segment starts with parameter u equal to zero, and ends with a value of one. This assumption simplifies defining boundary conditions, since segments now start at $u = 0$. Furthermore, this parametrization sets each segment to be independent on its length; it always ends at $u = 1$.

4-2-1 Initial position

The first known information is the initial position. It is known from which x and y coordinates a segment starts. Since it is assumed that $u = 0$ at the beginning of each segment, substituting u for 0 in Equations (4-1) and (4-2) gives segment's k initial position:

$$x_k(0) = \quad a_{0,k} \tag{4-3}$$

$$y_k(0) = \quad b_{0,k}. \tag{4-4}$$

Using the fact that the segment starts at coordinates $(x_{k,\text{init}}, y_{k,\text{init}})$, a solution has been found for the following two coefficients:

$$a_{0,k} = \quad x_{k,\text{init}} \tag{4-5}$$

$$b_{0,k} = \quad y_{k,\text{init}}. \tag{4-6}$$

4-2-2 Initial tangent angle

The second set of boundary conditions uses the initial tangent direction. In Section 4-1 it was said that only the position and velocity is given as input information. However, the desired tangent direction for the start of a segment is also known. If the vehicle is at the start of a

segment, its current tangent direction is known. Since it is desired to have continuity of the tangent direction, upcoming segment k should start with this same tangent direction.

The tangent to the trajectory can be separated into the x-direction and y-direction. These values are found by taking the partial derivatives of $x_{\text{ref}}(u)$ and $y_{\text{ref}}(u)$ over parameter u, which gives the following equations:

$$\frac{\partial x_{\text{ref}}(u)}{\partial u} = a_1 + 2a_2u + 3a_3u^2\big|_{u=u(t)} \tag{4-7}$$

$$\frac{\partial y_{\text{ref}}(u)}{\partial u} = b_1 + 2b_2u + 3b_3u^2\big|_{u=u(t)}\,. \tag{4-8}$$

For simplicity, these partial derivatives are denoted as $x'(u)$ and $y'(u)$ for the remainder of this thesis. Since not every segment is the same length, the unit tangent must be calculated. This way two segments can be joined with a continuous tangent angle, regardless of the segment length. The unit tangent to the trajectory, here $\hat{p}(u)$, is calculated by:

$$\hat{p}(u) = \frac{p'(u)}{|p'(u)|}, \tag{4-9}$$

with $p'(u) = [x'(u), y'(u)]^T$. Substituting the partial derivatives of $x_{\text{ref}}(u)$ and $y_{\text{ref}}(u)$ in Equation (4-9) gives Equation (4-10).

$$\hat{p}(u) = \frac{\begin{bmatrix} a_1 + 2a_2u + 3a_3u^2 \\ b_1 + 2b_2u + 3b_3u^2 \end{bmatrix}}{\sqrt{(a_1 + 2a_2u + 3a_3u^2)^2 + (b_1 + 2b_2u + 3b_3u^2)^2}} \tag{4-10}$$

Evaluating the unit tangent at the start of segment k, with $u = 0$, gives the expression shown in Equation (4-11).

$$\hat{p}(0) = \frac{\begin{bmatrix} a_{1,k} \\ b_{1,k} \end{bmatrix}}{\sqrt{a_{1,k}^2 + b_{1,k}^2}} \tag{4-11}$$

Since the previous segment's final unit tangent, $\hat{p}_{k-1}(1)$, is known, a solution can be found for this segments coefficients $a_{1,k}$ and $b_{1,k}$.

$$\frac{\begin{bmatrix} a_{1,k} \\ b_{1,k} \end{bmatrix}}{\sqrt{a_{1,k}^2 + b_{1,k}^2}} = \hat{p}_{k-1}(1) \tag{4-12}$$

It must be noted that Equation (4-12) does not have a unique solution due to the normalization factor $\sqrt{a_{1,k}^2 + b_{1,k}^2}$. Setting the additional constraint $\sqrt{a_{1,k}^2 + b_{1,k}^2} = 1$ is a good solution, because it sets the tangent magnitude to 1. This way the initial unit tangent for this segment is equal to last segment's unit tangent $\hat{p}_{k-1}(1)$.

4-2-3 Initial yaw rate

The third set of conditions causes two segments to join with continuous yaw rate. The yaw rate is the rate of change of the tangent angle, or in other words, the time derivative of tangent angle ψ (see Figure 4-3).

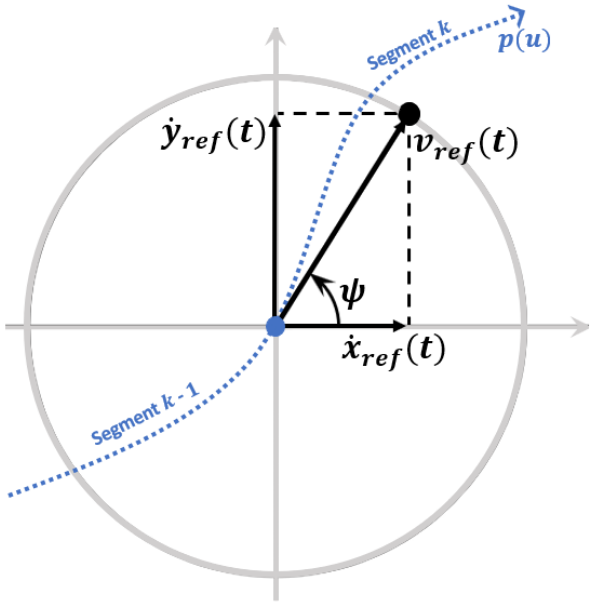


Figure 4-3: Tangent angle ψ at interconnection between segments k-1 and k visualized using the velocity vector $v_{\text{ref}}(t)$.

With the help of Figure 4-3, angle ψ can be calculated using the velocities in x and y direction.

$$\psi(t) = \arctan\left(\frac{\dot{y}_{\text{ref}}(t)}{\dot{x}_{\text{ref}}(t)}\right) \tag{4-13}$$

The time derivatives of x_{ref} and y_{ref} are found by Equations (4-14) and (4-15) respectively.

$$\dot{x}_{\text{ref}}(t) = x'(u) \cdot \dot{u}(t)|_{u=u(t)} \tag{4-14}$$
$$\dot{y}_{\text{ref}}(t) = y'(u) \cdot \dot{u}(t)|_{u=u(t)} \tag{4-15}$$

Here, $\dot{u}(t)$ denotes the time derivative of parameter u. Substituting Equations (4-14) and (4-15) in Equation (4-13) gives the following expression for angle $\psi(t)$:

$$\psi(t) = \arctan\left(\frac{y'(u) \cdot \dot{u}(t)}{x'(u) \cdot \dot{u}(t)}\right) = \arctan\left(\frac{y'(u)}{x'(u)}\right). \tag{4-16}$$

Obviously, Equation (4-16) only holds for $\dot{u}(t) \neq 0 \; \forall \; t \in [t_0, t_{\text{F}}]$. Section 4-3, Equations (4-39) and (4-40) show this is always true when the velocity is not equal to zero. Taking the time derivative of angle ψ gives the yaw rate, shown in Equation (4-17).

$$\dot{\psi}(t) = \frac{x'(t)\dot{y}'(t) - y'(t)\dot{x}'(t)}{x'(t)^2 + y'(t)^2} \tag{4-17}$$

In order to determine the full expression of $\dot{\psi}(t)$, define $\dot{x}'(t)$ and $\dot{y}'(t)$ (time derivatives of Equations (4-7) and (4-8) respectively) as:

$$\dot{x}'(t) = (2a_2 + 6a_3u(t)) \cdot \dot{u}(t) \tag{4-18}$$

$$\dot{y}'(t) = (2b_2 + 6b_3u(t)) \cdot \dot{u}(t). \tag{4-19}$$

Substituting these in Equation (4-17) gives:

$$\dot{\psi}(t) = \left. \frac{(a_1 + 2a_2u + 3a_3u^2)(2b_2 + 6b_3u)\dot{u} - (b_1 + 2b_2u + 3b_3u^2)(2a_2 + 6a_3u)\dot{u}}{(a_1 + 2a_2u + 3a_3u^2)^2 + (b_1 + 2b_2u + 3b_3u^2)^2} \right|_{u=u(t)}. \tag{4-20}$$

Evaluating the initial yaw rate for segment k, with $u = 0$ gives Equation (4-21).

$$\dot{\psi}_k(0) = \frac{2v_k(0)(a_{1,k}b_{2,k} - b_{1,k}a_{2,k})}{\sqrt{a_{1,k}^2 + b_{1,k}^2}(a_{1,k}^2 + b_{1,k}^2)} \tag{4-21}$$

Here, $v_k(0)$ is the initial velocity for segment k. Using the fact that $\dot{\psi}_{k-1}(1) = \dot{\psi}_k(0)$, Equation (4-22) can be derived.

$$\frac{2v_k(0)(a_{1,k}b_{2,k} - b_{1,k}a_{2,k})}{\sqrt{a_{1,k}^2 + b_{1,k}^2}(a_{1,k}^2 + b_{1,k}^2)} = \dot{\psi}_{k-1}(1) \tag{4-22}$$

This equation can be further simplified using $\sqrt{a_{1,k}^2 + b_{1,k}^2} = 1$, giving the following equation:

$$2v_k(0)(a_{1,k}b_{2,k} - b_{1,k}a_{2,k}) = \dot{\psi}_{k-1}(1) \tag{4-23}$$

Equation (4-23) has two unknowns ($a_{2,k}$ and $b_{2,k}$), so an unique solution cannot yet be obtained. However, if a similar assumption is made as in Section 4-2-2, by considering that $\sqrt{a_{2,k}^2 + b_{2,k}^2} = 1$, a unique solution can be found for these two coefficients such that two segments join each other with continuous yaw rate.

4-2-4 Final position

The last condition is the segment's final x and y coordinates. The final coordinates of segment k can be obtained by evaluating Equations (4-1) and (4-2) at $u = 1$. This gives Equations (4-24) and (4-25).

$$x_k(1) = a_{0,k} + a_{1,k} + a_{2,k} + a_{3,k} \tag{4-24}$$

$$y_k(1) = b_{0,k} + b_{1,k} + b_{2,k} + b_{3,k} \tag{4-25}$$

Using the information that this segment's spline should end at the desired final coordinates, ($x_{k,\text{final}}$, $y_{k,\text{final}}$), the following two equations can be formulated:

$$a_{0,k} + a_{1,k} + a_{2,k} + a_{3,k} = x_{k,\text{final}} \tag{4-26}$$

$$b_{0,k} + b_{1,k} + b_{2,k} + b_{3,k} = y_{k,\text{final}}. \tag{4-27}$$

Since spline coefficients $a_{0,k} \dots a_{2,k}$ and $b_{0,k} \dots b_{2,k}$ are known from using previous boundary conditions, $a_{3,k}$ and $b_{3,k}$ are the only unknowns in Equations (4-26) and (4-27), and can easily be obtained by rearranging these two equations.

$$a_{3,k} = x_{k,\text{final}} - a_{0,k} - a_{1,k} - a_{2,k} \tag{4-28}$$

$$b_{3,k} = y_{k,\text{final}} - b_{0,k} - b_{1,k} - b_{2,k} \tag{4-29}$$

4-3 Parametrization

As is seen in Equations (4-1) and (4-2), splines are a function of parameter u. This parameter should be a function of time, such that a trajectory, and not a geometric path is defined. In [23] parameter u has been set to be equal to time, that is: $u(t) = t - t_k, t \in [t_k, t_{k+1}]$. This way the trajectory is directly a function of time. The problem is that it is not possible to follow a non-constant velocity profile using this method of parametrization. In this research it is of interest to follow a velocity profile with at least a continuous acceleration. In [23] it is also attempted to set the parameter as a fixed linear, quadratic or higher order function of time. These relations are used to satisfy constraints on the maximum velocity, acceleration or jerk (time derivative of acceleration). In this case, no guarantee is given on the continuity of the velocity profile. This research tries an unique approach of parametrization by deriving $u(t)$ from a custom defined velocity profile. This is done by first defining the reference velocity as in Equation (4-30).

$$v_{\text{ref}}(t) = \sqrt{\dot{x}_{\text{ref}}(t)^2 + \dot{y}_{\text{ref}}(t)^2} \tag{4-30}$$

Using Equations (4-14) and (4-15) for the time derivatives of x_{ref} and y_{ref} gives Equation (4-31).

$$v_{\text{ref}}(t) = \dot{u}(t)\sqrt{x'(u)^2 + y'(u)^2}\Big|_{u=u(t)} \tag{4-31}$$

Rearranging the order of Equation (4-31) leads to the first-order non-linear differential function equation on $u(t)$ shown in Equation (4-32).

$$\dot{u}(t) = \frac{v_{\text{ref}}(t)}{\sqrt{x'(u)^2 + y'(u)^2}}\Big|_{u=u(t)} = f(u,t) \tag{4-32}$$

Equation (4-32) is solved numerically in the simulations. The method is described in detail in Section 4-5.

Equation (4-32) defines how parameter u is a function of time such that a custom defined velocity profile $v_{\text{ref}}(t)$ is defined along the trajectory. This way one can design a velocity profile per segment based on desired properties. In this research a C^2 continuous velocity profile is desired, thus being continuous in both velocity and acceleration. The next section describes the process to defining such a reference velocity profile.

4-4 Velocity profile

There are many methods to designing a reference velocity profile. One could use polynomials of a certain degree to fit a curve from the current velocity to the desired velocity of the next via-point. The problem lies within the parametrization, since the reference velocity is a function of time, and this segment's final time t_f is unknown yet. However, it is possible if one would use a symmetrical velocity profile such as in Figure 4-4.

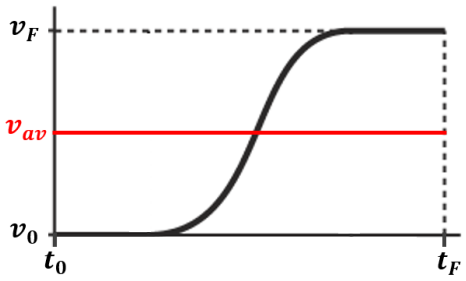


Figure 4-4: Velocity profile v_{ref} defined using a Sigmoid function.

The black curve starts the segment at v_0, and smoothly gaining to the desired velocity v_F at the end of the segment. A Sigmoid function approximates such a curve. Since it has a symmetrical property, taking the average velocity between v_0 and v_F (red curve in Figure 4-4) gives the opportunity to find this segment's final time t_F. One would arrive at the end of a segment at the same time using either the Sigmoid function velocity profile or the average velocity profile. Notice in Figure 4-4 that the integral under the black curve is equal to the integral under the red curve for the same period $[t_0, t_F]$. This integral is equal to segment length L_k. Since the spline coefficients have been found (Section 4-2), L_k is fixed and can be determined using the following equation:

$$L_k = \int_0^1 \sqrt{(x'_{\text{ref},k}(u)^2 + y'_{\text{ref},k}(u)^2}\, \mathrm{d}u. \tag{4-33}$$

Given segment length L_k and average velocity v_{av}, one can find t_F by using:

$$t_F = \frac{L_k}{v_{\text{av}}} + t_0. \tag{4-34}$$

With this segment's final time known, a velocity profile can be defined in the domain $[t_0, t_F]$. This research uses an adapted Pearl-Reed logistic function from [34], which is described by the following equation:

$$v_{\text{ref}}(t) = \frac{v_0 \cdot e^{cr} + v_F \cdot e^{rt}}{e^{cr} + e^{rt}}, \tag{4-35}$$

with v_0 and v_F this segment's initial and final velocities respectively, such as in Figure 4-4. Coefficient c operates as a step location shifter. In this research it is assumed to always be at the middle of the segment, at $\frac{t_0+t_F}{2}$. Coefficient r changes the slope of the velocity. For instance, if the velocity difference is large, this slope should be increased to make sure the final velocity is reached within the time domain. However, the slope cannot be too large, because the reference acceleration would exceed the maximum acceleration of the vehicle. The maximum acceleration using the velocity profile shown in Equation (4-35) can be calculated using the following equation:

$$a_{\max} = \frac{r(v_F - v_0)}{4}. \tag{4-36}$$

In this research the difference in velocity for each segment is approximately the same, so coefficient r is at a fixed value. Obviously, if this change in velocity is not the same, the coefficient must be tuned appropriately using Equation (4-36) and taking maximum accelerations of truck-trailer vehicles into account.

Besides the symmetrical property of the Sigmoid function, it also has great continuity features. See Figure 4-5 for a segment's acceleration profile corresponding with the reference velocity as in Equation (4-35). As can be noted from this figure, it starts and ends with an acceleration of zero. This leads to a continuous interconnection between segments in terms of acceleration. In other words, C^2 continuity has been achieved using the Sigmoid function as a reference velocity profile.

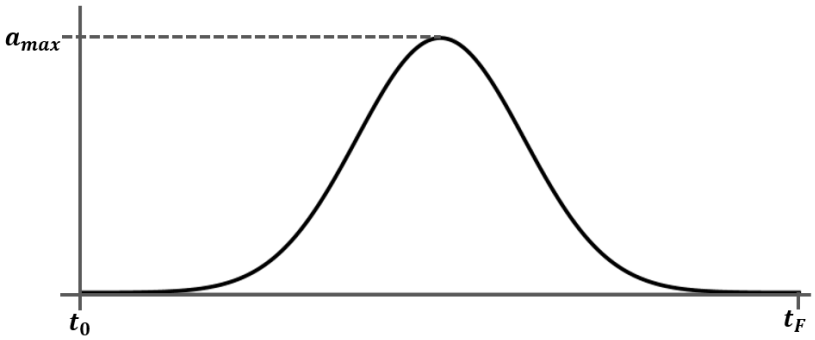


Figure 4-5: Acceleration profile corresponding with the Sigmoid function velocity profile.

However, using a Sigmoid function as a reference velocity profile is not ideal. While continuity of acceleration is achieved, starting and ending every segment with an acceleration of zero is not the most efficient way of speeding up or slowing down. Furthermore, as is seen in Figure 4-5, large acceleration peaks are obtained each segment using this reference velocity profile. According to [35], depending on the weight-to-power ratio, the maximum acceleration for truck-trailer vehicles varies between approximately 0.3 and 0.7 [m/s^2] when driving around 30 [km/h]. In order to stay below these bounds, the velocity difference cannot be too large for each segment. In this research it is assumed that the desired velocity on each via-point is given by the manoeuvre planner, so this planning module should take the maximum accelerations into account when deciding the velocity for the next via-point. In this thesis' simulations the velocity differences are manually set to be small enough to stay below the maximum accelerations given by [35]. Future studies could look into methods of deriving a velocity profile without it necessarily being symmetric. This could perhaps be done by estimating the time it takes to traverse this segment's path.

4-5 Numerical integration

With the reference velocity profile defined, the differential equation shown in Equation (4-32) can be solved. Before doing so, the differential equation must be tested on the existence and uniqueness of the solution. This is done by the Picard-Lindelöf theorem in [36].

The following theorem proves the existence of a solution. Suppose that $f(u,t)$ is a continuous function in the region

$$R = (u,t) : |u - u_0| \leq a, |t - t_0| \leq b, \quad (a,b > 0). \tag{4-37}$$

Since f is continuous in a closed and bounded domain, it is necessarily bounded in R, i.e. there exists $K > 0$ such that $f(u,t) \leq K \ \forall (u,t) \in R$. Therefore the differential equation has

at least one solution $u = u(t)$ defined in the interval $|t - t_0| < \beta$ where

$$\beta = \min\left\{\frac{a}{K}, b\right\}. \tag{4-38}$$

Showing the full expression of $f(u,t)$ gives the following equation:

$$\dot{u}(t) = \left.\frac{v_{\text{ref}}(t)}{(a_1 + 2a_2u + 3a_3u^2)^2 + (b_1 + 2b_2u + 3b_3u^2)^2}\right|_{u=u(t)} = f(u,t) \tag{4-39}$$

A discontinuity could occur if the reference velocity is discontinuous. However, this velocity profile is defined such that it is C^2. Another discontinuity could occur if the denominator in Equation (4-39) would become zero. The only possibility of this happening is at the start of the segment with $u = 0$ and $t = 0$. Evaluating the differential function at this initial point gives Equation (4-40).

$$\dot{u}(0) = \frac{v_{\text{ref}}(0)}{\sqrt{a_1^2 + b_1^2}} \tag{4-40}$$

In Section 4-2 the condition $\sqrt{a_1^2 + b_1^2} = 1$ was set in order to find an unique solution for spline coefficients a_1 and b_1. This causes the denominator in Equation (4-40) to never become equal to zero. Therefore, $f(u,t)$ is continuous over the domain $u =[0,1]$ and $t =[t_0, t_F]$, and at least one solution exists.

The second theorem proves the uniqueness of a solution. Suppose that f and $\partial f/\partial u$ are continuous functions in R (defined in Equation (4-37)). Hence, both f and $\partial f/\partial u$ are bounded in R, i.e.,

$$\text{(a) } |f(u,t)| \leq K \quad \text{and} \quad \text{(b)} \left|\frac{\partial f}{\partial u}\right| \leq L \quad \forall (u,t) \in R. \tag{4-41}$$

Then differential equation $f(u,t)$ has at most one solution $u = u(t)$ defined in the interval $|t - t_0| < \beta$ where

$$\beta = \min\left\{\frac{a}{K}, b\right\}. \tag{4-42}$$

Condition (a) is shown to be true at the existence theorem. Performing the partial derivative described in condition (b) gives the following equation:

$$\frac{\partial f}{\partial u}(u,t) = \left.-\frac{2((2a_2+6a_3u)(a_1+2a_2+3a_3u^2)+(2b_2+6b_3u)(b_1+2b_2u+3b_3u^2))v_{\text{ref}}(t)}{2((a_1+2a_2u+3a_3u^2)^2+(b_1+2b_2u+3b_3u^2)^2)^{3/2}}\right|_{u=u(t)}. \tag{4-43}$$

Once again, the reference velocity is continuous over the domain. The only other discontinuity could occur at the start of the segment. Evaluating Equation (4-43) at $u = 0$ and $t = 0$ gives the following equation:

$$\frac{\partial f}{\partial u}(0,0) = -\frac{4(a_1a_2 + b_1b_2)v_{\text{ref}}(0)}{2(a_1^2 + b_1^2)^{3/2}}. \tag{4-44}$$

The condition $\sqrt{a_1^2 + b_1^2} = 1$ causes Equation (4-44) to be bounded, thus realizing Equation (4-43) to be continuous over the domain $u =[0,1]$ and $t =[t_0, t_F]$. Both conditions in Equation (4-41) are true, so uniqueness of a solution is proven.

Since equation (4-39) is non-linear, it is difficult to find a closed-form analytical solution for it. Thus, a numerical method should be done to perform the integration $\dot{u}(t)$. This is done by the Classical Runge-Kutta (RK4) method as is described in [37,38]. Define the differential equation that is to be solved by

$$\dot{u}(t) = f(u,t), \qquad u(t_0) = 0 \tag{4-45}$$

The initial condition states that $u = 0$ at the start of each segment. The RK4 method defines an approximation of the next value of parameter u and time t as in Equations (4-46) and (4-47).

$$u_{n+1} = u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{4-46}$$

$$t_{n+1} = t_n + h, \tag{4-47}$$

for $n = 0, 1, 2, 3...$ Here, h denotes the step-size, and $k_1 \dots k_4$ are four slopes defined by

$$\begin{aligned} k_1 &= f(u_n, t_n) \\ k_2 &= f(u_n + h\frac{k_1}{2}, t_n + \frac{h}{2}) \\ k_3 &= f(u_n + h\frac{k_2}{2}, t_n + \frac{h}{2}) \\ k_4 &= f(u_n + hk_3, t_n + h). \end{aligned}$$

Performing this numerical integration until final time t_F has been reached gives an accurate solution of $u(t)$. Notice, t_F has already been computed as is described in Section 4-4. Using the RK4 method, the local truncation error is on the order $\mathcal{O}(h^5)$, while the total accumulated error is $\mathcal{O}(h^4)$. By reducing step-size h the accuracy of the integration can be increased, but at the cost of computational effort.

4-6 Discussion

This chapter showed the procedure of generating a trajectory by using cubic splines. Spline coefficients are chosen such that continuity in position, tangent direction and yaw rate is achieved between segments. This way the reference lateral movement of the vehicle is continuous. Furthermore, an unique method of parametrization is applied such that a custom constructed reference velocity profile is defined along the trajectory. This profile is chosen such that continuity in longitudinal velocity and acceleration is achieved. This complete procedure of trajectory generation is computationally efficient, since only one trajectory is computed instead of generating multiple trajectories for each segment and discarding most of them. Furthermore, it is online because only the vehicle's current configuration and the next via-point's desired position and velocity are used.

In order to assess this method's performance, a set of simulations was performed. Figure 4-6 shows the generated trajectory for a lane change manoeuvre using the cubic splines method.

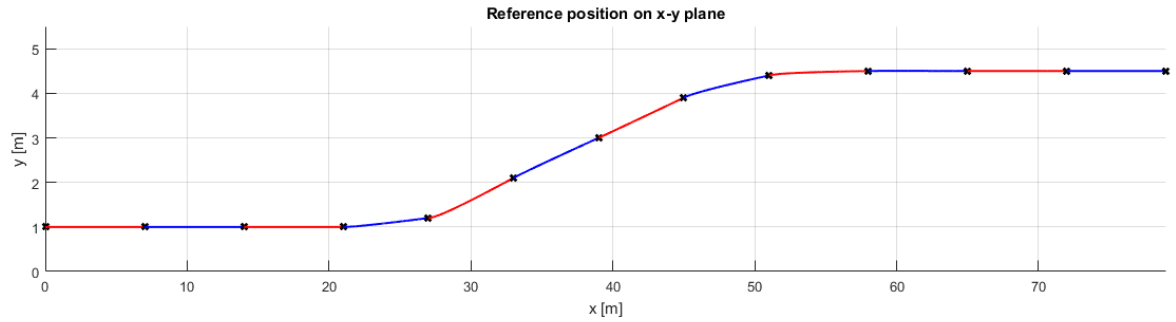


Figure 4-6: Reference trajectory using the cubic splines method.

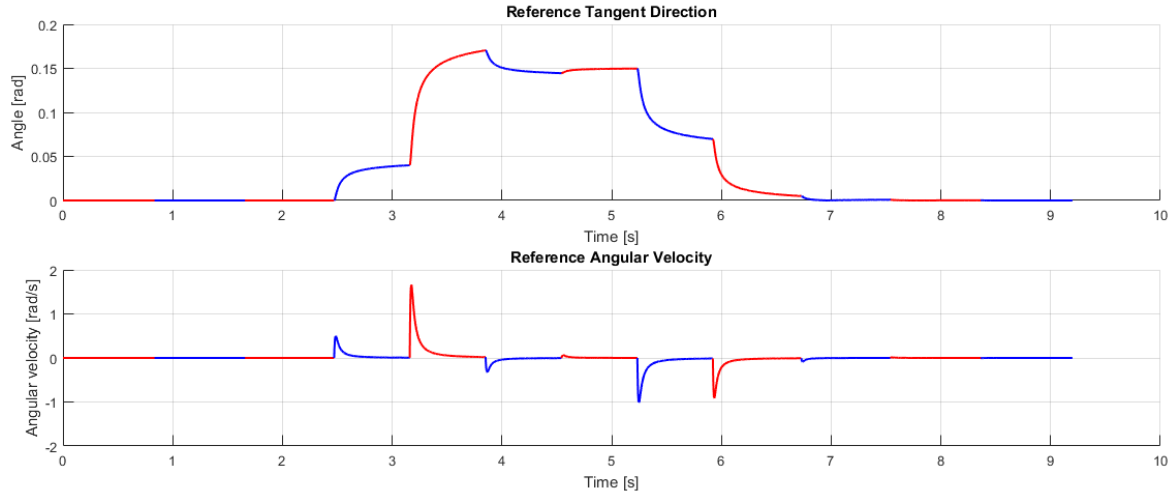


Figure 4-7: Reference tangent angle (above) and yaw rate (below) using cubic splines method.

Figure 4-7 shows the reference tangent angle and yaw rate for this same trajectory. Recall that both of these functions should be continuous according to the procedure described in Section 4-2. Figure 4-7, however, seems to suggest that the reference yaw rate (or angular velocity) is discontinuous. This is not the case, as Figure 4-8 clearly shows that at an interconnection between two segments the yaw rate is indeed continuous. However, it seems that the reference yaw rate becomes very large at the start of each segment. At one segment it even reaches an angular velocity of approximately 1.5 [rad/s], which is equal to an lateral acceleration of approximately 14 [m/s^2] when driving 30 [km/h]. This is unacceptably large, and should be reduced. Although the trajectory generator described in this chapter fulfils all the design requirements described in Chapter 1, some of the trajectories it generates cannot be followed in practice by a standard vehicle. Such trajectories require a large sudden movement in the lateral direction, and thus requiring high yaw rate peaks. The next chapter shows a possible solution to reducing these peaks, while still utilizing a parametric method.

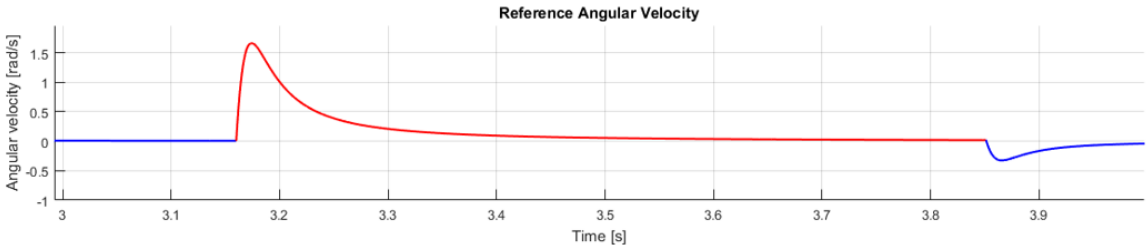


Figure 4-8: Close look at the reference yaw rate.

Chapter 5

Quintic splines

The problem of having high yaw rate peaks in the reference trajectory cannot be solved directly by using only cubic splines, since all spline coefficients are fixed in order to keep continuity in position, tangent angle, and yaw rate. However, if higher order splines are chosen, some coefficients become available for tuning. Choosing quintic splines yields a trajectory $p(t) = [x_{\text{ref}}(t), y_{\text{ref}}(t)]^T$ defined as followed:

$$x_{\text{ref}}(t) = a_0 + a_1 u + a_2 u^2 + a_3 u^3 + a_5 u^5 \Big|_{u=u(t)} \tag{5-1}$$

$$y_{\text{ref}}(t) = b_0 + b_1 u + b_2 u^2 + b_3 u^3 + b_5 u^5 \Big|_{u=u(t)} . \tag{5-2}$$

Notice that it would also be possible to choose fourth order splines to get additional coefficients for tuning. However, that would result in an even number for the highest order of u, which will completely change the behaviour of the trajectory. This is caused by the symmetric property of polynomials with an even order, as is described in [39]. To visualize why this would be a problem, consider interpolating via-points using parabolas; trying to keep the tangent angle continuous between segments will lead to an oscillating trajectory. This problem is avoided by using an odd number for the highest order, such as quintic splines.

Furthermore, it could be noticed that in Equations (5-1) and (5-2) coefficients a_4 and b_4 have been left out, and only a_5 and b_5 have been added. The reason behind this is the fact that only one extra condition is demanded: minimization of peak yaw rate, thus only one coefficient needs to be added per direction (x and y).

5-1 Spline coefficients

As before, the spline coefficients must be found first. Coefficients $a_0 \dots a_2$ and $b_0 \dots b_2$ are obtained using the same procedure as in Chapter 4; boundary conditions for position, tangent direction, and yaw rate. The next subsections will describe how the rest of the coefficients are obtained.

5-1-1 Coefficients a_3 and b_3

A slight change of method is required to obtain coefficients a_3 and b_3. However, the same boundary condition is utilized. This is done by using this segment's desired final position. For segment k, the final position at $u = 1$ for quintic splines is defined by Equations (5-3) and (5-4).

$$x_k(1) = \quad a_{0,k} + a_{1,k} + a_{2,k} + a_{3,k} + a_{5,k} \tag{5-3}$$
$$y_k(1) = \quad b_{0,k} + b_{1,k} + b_{2,k} + b_{3,k} + b_{5,k} \tag{5-4}$$

Using the information that this segment's spline should end at the desired final coordinates ($x_{k,\text{final}}$, $y_{k,\text{final}}$), the following two equations can be formulated.

$$a_{0,k} + a_{1,k} + a_{2,k} + a_{3,k} + a_{5,k} = \quad x_{k,\text{final}} \tag{5-5}$$
$$b_{0,k} + b_{1,k} + b_{2,k} + b_{3,k} + b_{5,k} = \quad y_{k,\text{final}} \tag{5-6}$$

Whereas with cubic splines only coefficients $a_{3,k}$ and $b_{3,k}$ were unknown, now also coefficients $a_{5,k}$ and $b_{5,k}$ are unknown. So for now, no explicit solution can be found for a_3 and b_3. What can be done is expressing $a_{3,k}$ and $b_{3,k}$ as functions of $a_{5,k}$ and $b_{5,k}$ respectively.

$$a_{3,k} = \quad x_{k,\text{final}} - a_{0,k} - a_{1,k} - a_{2,k} - a_{5,k} \tag{5-7}$$
$$b_{3,k} = \quad y_{k,\text{final}} - b_{0,k} - b_{1,k} - b_{2,k} - b_{5,k} \tag{5-8}$$

So if solutions are found for $a_{5,k}$ and $b_{5,k}$, Equations (5-7) and (5-8) can be used to find $a_{3,k}$ and $b_{3,k}$.

5-1-2 Coefficients a_5 and b_5

The problem of the cubic splines method was that it caused high yaw rate peaks. The next subsections describe how the tunable coefficients a_5 and b_5 can be found such that these yaw rate peaks can be reduced.

The objective function

The full expression of the yaw rate function $\dot{\psi}(t)$ using quintic splines is shown in Equation (5-9).

$$\dot{\psi}(t) = \left.\frac{(a_1+2a_2u+3a_3u^2+5a_5u^4)(2b_2+6b_3u+20b_5u^3)\dot{u}-(b_1+2b_2u+3b_3u^2+5b_5u^4)(2a_2+6a_3u+20a_5u^3)\dot{u}}{(a_1+2a_2u+3a_3u^2+5a_5u^4)^2+(b_1+2b_2u+3b_3u^2+5b_5u^4)^2}\right|_{u=u(t)} \tag{5-9}$$

Substituting a_3 and b_3 in this function by Equations (5-7) and (5-8), the only free parameters become a_5 and b_5. The two free coefficients a_5 and $b_5$can be found by solving the minimization problem stated in Equation (5-10).

$$\min_{a_5,b_5} \quad \left|\dot{\psi}(t)\right|_\infty \tag{5-10}$$

Solving this equation would result in minimal yaw rate peaks for each segment. Instead of minimizing, one could also choose to define an upper bound such that the yaw rate would

never exceed a certain limit. However, it is difficult to define proper bounds. There are many factors that influence it, such as the vehicle weight or velocity. Furthermore, why would one define upper bounds when it is possible to make the yaw rate minimal. The goal of this thesis is to generate a reference trajectory that is continuous to realize comfort. A vehicle passenger would feel more comfortable with minimal lateral accelerations, thus minimal yaw rates. If the goal was to obtain optimal cornering manoeuvres for racing scenarios, then minimal yaw rates would not be desired, and upper bounds would be preferred.

The problem with the optimization defined in Equation (5-10) is the fact that the domain $[t_0, t_F]$ is not fully known. In Section 4-4 it is explained that t_F can be found using the average velocity for the segment. But in order to do so, all coefficients must be known, and in this case a_5 and b_5 are still unknown. Therefore, this segment's final time t_F is unknown.

A possible solution to this problem could be done by separating the yaw rate function into two functions, as is shown in Equation (5-11).

$$\dot{\psi}(t) = f(u) \cdot v_{\text{ref}}(t) \tag{5-11}$$

Here, function $f(u)$ contains all parts of $\dot{\psi}(t)$ that can be expressed as a function of parameter u, and $v_{\text{ref}}(t)$ is the only term that is a function of time. Using this separation, according to [41] a submultiplicative property can be applied to the norm, which is shown in Equation (5-12).

$$\left|\dot{\psi}(t)\right|_\infty \leq |f(u)|_\infty \cdot |v_{\text{ref}}(t)|_\infty \tag{5-12}$$

This gives the opportunity to define a new minimization function, such as in Equation (5-13).

$$\min_{a_5, b_5} \quad |f(u)|_\infty \tag{5-13}$$

Whereas with Equation (5-10) the segment's final time t_F was unknown, the final value of parameter u in Equation (5-13) is simply equal to 1. Therefore the minimization shown in Equation (5-13) can be performed.

It may seem that minimizing $|f(u)|_\infty$ is not the same as minimizing $\left|\dot{\psi}(t)\right|_\infty$, which is true. The only term that has been left out of the minimization function is $|v_{\text{ref}}(t)|_\infty$. However, the value of this term is easily obtained, since it is equal to v_0 in case of a decelerating manoeuvre, and v_F in case of an accelerating manoeuvre (see Figure 4-4). Therefore, the difference between minimizing Equations (5-13) and (5-10) is this maximum velocity of a segment, which is known beforehand. However, a slightly suboptimal solution may be found by minimizing Equation (5-13), because it is a search of the upper bound of the yaw rate's magnitude, caused by the submultiplicative property described in Equation (5-12).

The optimization process

As an example, the function $f(u)$ at segment k for a fixed value of a_5 and a varying value of b_5 (indicated by a darker curve with increasing value of b_5) is shown in Figure 5-1. It must be noted that coefficients a_5 and b_5 have been varied on the interval [-1, 1]. Evaluating values outside this interval does not yield lower peak values for the reference yaw rate, while only increasing the computation time.

Searching for the lowest peak is a simple task. The first step consists of evaluating the absolute maximum value of $f(u)$ for different values of b_5 on the interval [-1, 1]. The next step is to find the value of coefficient b_5 for which the peak of $f(u)$ is the minimal. The solution is plotted as a red curve in Figure 5-1. Repeating the above process for other values of a_5 within the interval [-1, 1] gives the optimal solution of the two free variables for which the peak of $f(u)$ is the lowest.

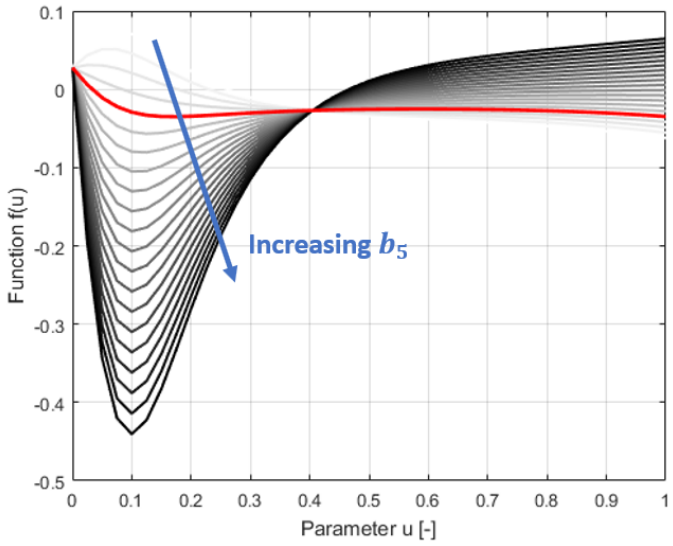


Figure 5-1: Function $f(u)$, for increasing values of b_5 and a fixed value of a_5. Curve with lowest peak indicated by the red line.

The results of this optimization is shown in Figures 5-2 and 5-3, which depict the reference trajectory and its corresponding tangent angle and yaw rate profile respectively. Whereas with cubic splines the reference yaw rate reached values up to 1.5 [rad/s], the maximum yaw rate in this case reaches 0.5 [rad/s] at most. This translates to a lateral acceleration of approximately 4 [m/s^2] when driving 30 [km/h]. The reduction of the yaw rate peaks comes at a cost, however. As is shown in Figure 5-2, the reference position profile oscillates. The next section describes the cause of this oscillation, and a possible method of reducing it.

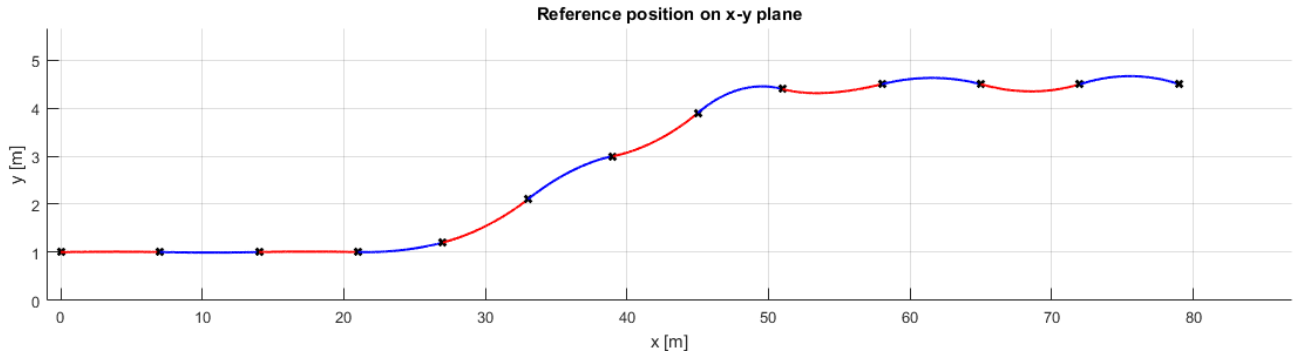


Figure 5-2: Reference trajectory using quintic splines method with yaw rate peak minimization.

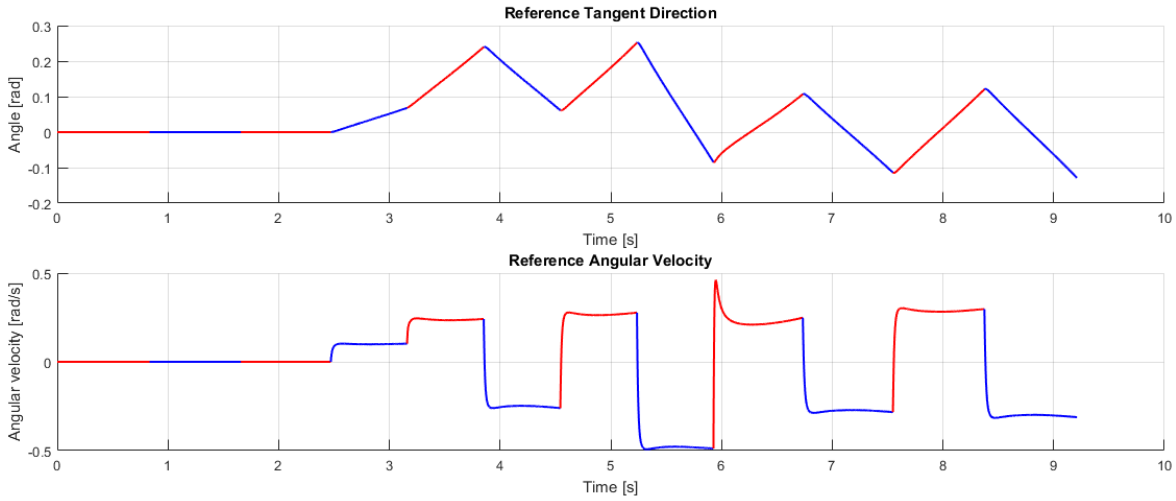


Figure 5-3: Reference tangent angle (above) and yaw rate (below) using quintic splines method with yaw rate peak minimization.

5-2 Reducing oscillations

It seems that minimizing the reference yaw rate peaks causes oscillations in the generated trajectory. This effect can be explained using Figure 5-1. Here, the curve with the minimal yaw rate peak for one segment is displayed in red. Notice that the final value of function f at $u = 1$ is not equal to zero, and thus the final reference yaw rate per segment is not zero. Since continuity of yaw rate is preserved, the next segment will start at this non-zero reference yaw rate. This occurrence keeps repeating and results in a non-stabilizing solution of the reference yaw rate for each segment, and thus causing an oscillating reference trajectory.

A possible solution to this problem is to define a trade-off between minimal reference yaw rate peaks and reducing the oscillations. Searching for the set of values for a_5 and b_5 for which the oscillations are minimized is done by solving Equation (5-14).

$$\lim_{u \to 1} f(u) = 0 \tag{5-14}$$

In other words, a set of values for a_5 and b_5 is searched such that this segment's final yaw rate is equal to zero. Figure 5-4 shows function $f(u)$ for a fixed value of a_5, and b_5 varied on the interval [-1, 1], indicated by the increasingly darker curve. The solution for which value of b_5 the final value is closest to zero is indicated in Figure 5-4 as a blue curve.

Repeating this process with coefficient a_5 varied on the same interval, an optimal solution is found for the two free coefficients.

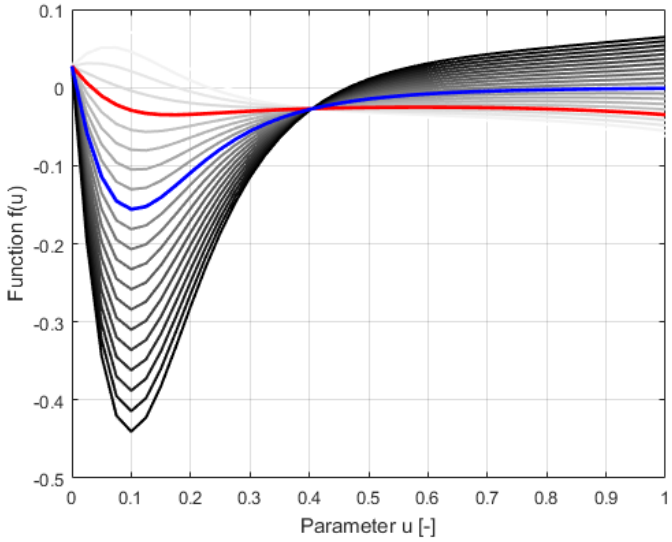


Figure 5-4: Function $f(u)$, with varying values of b_5. Curve with final value equal to zero indicated by the blue line.

The challenge is to define a proper trade-off between minimizing reference yaw rate peaks and reducing oscillations. A possible solution could be applied by introducing a linear trade-off between the two factors. For example, consider $[a_5, b_5]^T_{\text{peak}}$ the optimal set of solutions for which the peak of $f(u)$ is minimal, and $[a_5, b_5]^T_{\text{final}}$ the set of solutions for which the final value of $f(u)$ is minimal, then the trade-off solution could be found by:

$$\begin{bmatrix} a_5 \\ b_5 \end{bmatrix}_{\text{trade-off}} = (1-\alpha) \begin{bmatrix} a_5 \\ b_5 \end{bmatrix}_{\text{peak}} + \alpha \begin{bmatrix} a_5 \\ b_5 \end{bmatrix}_{\text{final}} \tag{5-15}$$

with α the trade-off value between zero and one. Choosing $\alpha = 0$ results in fully minimized reference yaw rate peaks, with a high chance of having an oscillating reference trajectory. Choosing $\alpha = 1$ results in minimal oscillations of the reference trajectory, but relatively high reference yaw rate peaks. The choice of parameter α depends on the scenario. For instance, having a fully loaded truck demands low lateral accelerations, or else the truck could roll over, so a small value of α must be chosen.

As an example, Figures 5-5 and 5-6 show the reference trajectory profile and the corresponding tangent angle and yaw rate profile respectively using α equal to 0.4. It could be noted that the oscillations have been significantly reduced with respect to Figure 5-2, while the reference yaw rate peaks are reasonably low. In this case, the reference yaw rate reaches approximately 0.8 [rad/s] at most. Compared to the 1.5 [rad/s] yaw rate reached with cubic splines, this method has reduced the peaks drastically. Furthermore, compared to the method that purely reduces the yaw rate peaks (shown in Figures 5-2 and 5-3), the maximum yaw rate has only slightly increased from 0.5 to 0.8 [rad/s].

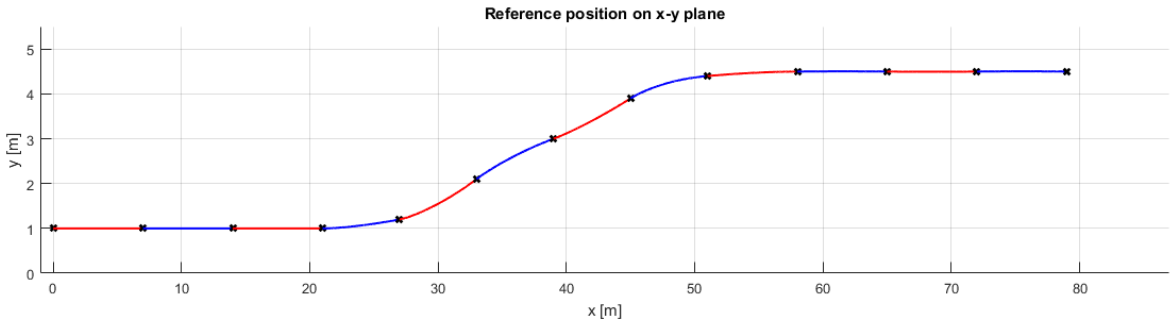


Figure 5-5: Reference trajectory using quintic splines method with linear trade-off between minimizing yaw rate peaks and oscillation.

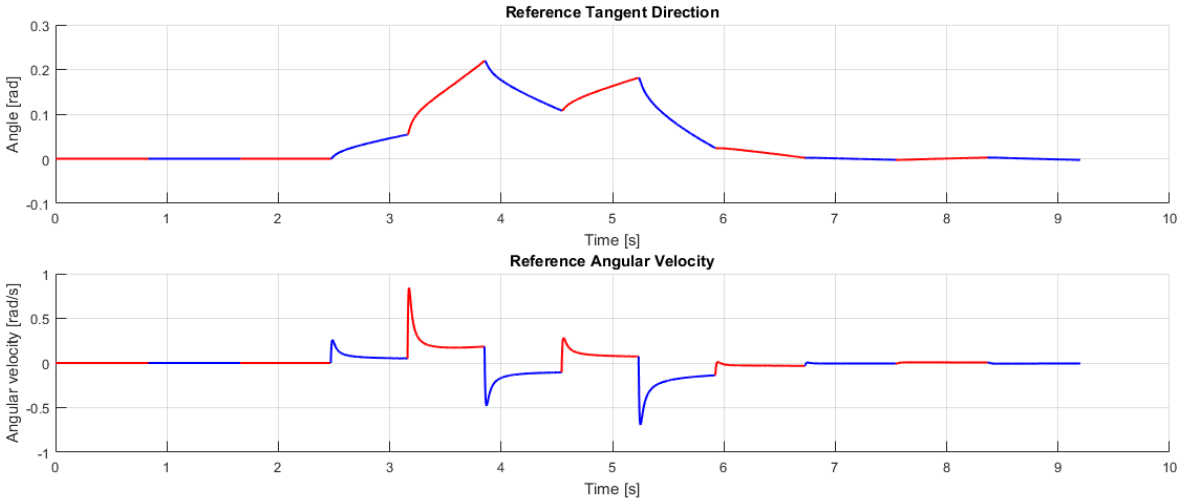


Figure 5-6: Reference tangent angle (above) and yaw rate (below) using quintic splines method with linear trade-off between minimizing yaw rate peaks and oscillation.

Chapter 6

Trajectory tracking

This chapter presents a initial exploration of how well the trajectories generated using the methods described in Chapters 4 and 5 can be tracked by a vehicle. Although those chapters showed preliminary results that illustrate the continuity of the reference trajectories in position, tangent direction and yaw rate (thus allowing them to be followed by a vehicle with continuous lateral motion), the generators were not tested in closed loop with the vehicle dynamics. Such test are presented here. To this end, a trajectory tracker was implemented which generates control inputs for the vehicle. The latter is modelled using the kinematic model explained in Chapter 2. The trajectory tracker works in closed loop so to minimize the tracking error. Several simulation parameters are varied to compared different manoeuvres, so to analyse results with different lane-change times or inter via-points distances.

6-1 Global to local coordinate frame

The first step is to transform the reference coordinates x_{ref} and y_{ref} from the earth-fixed coordinate system (x, y) to the vehicle local coordinate system (lateral and longitudinal position from the vehicle's point of view). Figure 6-1 shows an illustration with the vehicle's current state $s_v \triangleq [x_v, y_v, \psi_v]^T$, and the desired reference state $s_{\text{ref}} \triangleq [x_{\text{ref}}, y_{\text{ref}}, \psi_{\text{ref}}]^T$. These are the coordinates on the earth-fixed coordinate system (x, y) and the tangent angle measured from the x-axis. The error in the earth-fixed coordinate system is denoted as $[e_x, e_y, e_\psi]^T$ and can by:

$$\begin{bmatrix} e_x \\ e_y \\ e_\psi \end{bmatrix} = \begin{bmatrix} x_{\text{ref}} - x_v \\ y_{\text{ref}} - y_v \\ \psi_{\text{ref}} - \psi_v \end{bmatrix}. \tag{6-1}$$

With the global error defined as in Equation (6-1), the local error can be determined. This local error is also shown in Figure 6-2. Here it can be noticed that in this coordinate frame the longitudinal axis is in the direction of the vehicle heading angle, and the lateral axis

perpendicular to this direction. Therefore, the local errors can be calculated using Equation (6-2).

$$\begin{bmatrix} e_{\text{lat}} \\ e_{\text{long}} \end{bmatrix} = R(\psi_v) \begin{bmatrix} e_x \\ e_y \end{bmatrix} \tag{6-2}$$

Here, $\mathrm{R}(\psi_v)$ denotes the rotation matrix that performs the rotation from global to local coordinates, and is described by Equation (6-3).

$$R(\psi_v) = \begin{bmatrix} -\sin(\psi_v) & \cos(\psi_v) \\ \cos(\psi_v) & \sin(\psi_v) \end{bmatrix} \tag{6-3}$$

Obviously, the error in tangent direction (e_ψ) does not change when mapping from the global to local coordinate system.

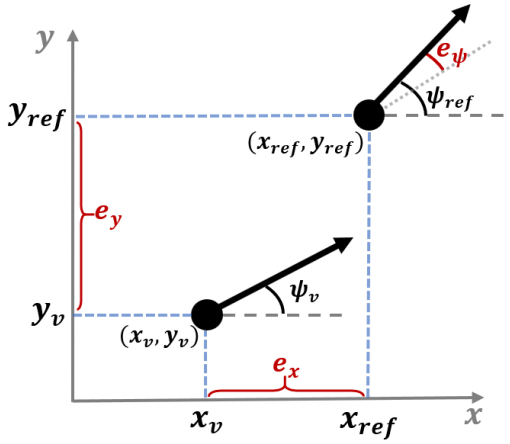


Figure 6-1: Vehicle's current and reference state on the earth-fixed coordinate system, with the global error denoted as (e_x, e_y, e_ψ).

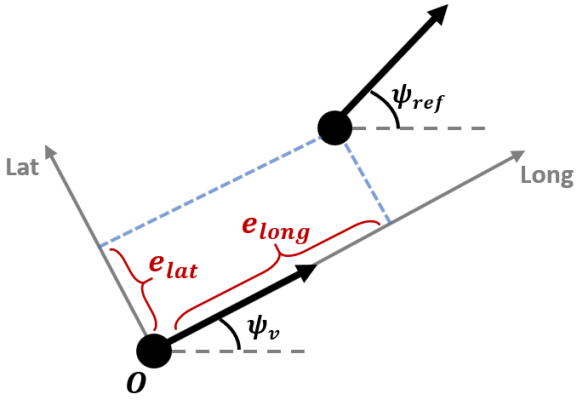


Figure 6-2: Vehicle's current and reference state on its local coordinate system (lateral and longitudinal), with the local error denoted as $(e_{\text{lat}}, e_{\text{long}})$.

6-2 Control input

With the local error defined as in Equation (6-2), a control input can be generated. Since the trajectory generator is parametrized by time, the controller should be able to generate a tracking input that is also parametrized by time. This is important, because otherwise the designed reference velocity profile will not be followed.

It must be noted that the main focus of this research is on the generation of the trajectory, and not on its tracking. Therefore an existing controller will be used for simulation purposes. Even though it is possible to design a custom trajectory tracker, there are many studies that have done this before. Many controllers can be found that satisfy desired specifications. As an example, in [42] a tracking controller is designed for unicycle mobile robots. This controller guarantees global asymptotic stability of the tracking errors while being able to track a velocity profile. Furthermore, the tuning parameters require only some initial tuning and the control law is computationally cheap. In [42] it is discussed that the proposed controller offers several advantages compared to those in the literature, such as better suppression of tracking errors due to more freedom in controller tuning. The controller designed in [42] will therefore be used for simulation purposes in this research.

6-2-1 Control laws

The control laws proposed in [42] are defined as followed:

$$v^*(t) = v_{\text{ref}}(t)\cos(e_\psi(t)) + r_x \tanh(k_x e_{\text{long}}(t)) \tag{6-4}$$

$$\omega^*(t) = \dot{\psi}_{\text{ref}}(t) + k_y v_{\text{ref}}(t)\text{sinc}(e_\psi(t))e_{\text{lat}}(t)\sqrt{\frac{c}{1+c(e_{\text{lat}}^2+e_{\text{long}}^2)}} + r_\psi \tanh(k_\psi e_\psi(t)), \tag{6-5}$$

with $v^*(t)$ the control law for the longitudinal component of the tyre velocity, and $\omega^*(t)$ the control law for the rotational velocity. Furthermore, $v_{\text{ref}}(t)$ is the reference forward velocity, as is defined in Section 4-4, and $\dot{\psi}_{\text{ref}}(t)$ the reference angular velocity (also known as yaw rate). Since the trajectory generator uses a parametric method, the function shown in Equation (5-9) can be used to determine the reference yaw rate at each time step. Furthermore, c, k_x, k_y, k_ψ, r_x, $r_\psi \in R_+$ are design parameters than can be tuned in order to reach desired tracking performance. These parameters have been set as constants during simulations in this research, and lead to bounded and uniformly continuous control functions according to [42]. In [42] a method is described to define the design parameters as time-variant to further increase the performance of the resulting controller. This way bounds can be set on the maximum reference longitudinal and rotational velocity. However, since only small errors are expected in the simulation results with relatively simple manoeuvres, setting the parameters as constant should give adequate control performance results.

6-2-2 High level control

With the control laws defined as in Equations (6-4) and (6-5), a reference input can be generated. As is described in Section 2-2, the two inputs to the kinematic truck-trailer model

are steering input δ_{in} and velocity input v_{in}. Define two trigonometric relationships shown in Equations (6-6) and (6-7) as is described in [43].

$$v^*(t) = v(t)\cos(\delta) \tag{6-6}$$

$$d_0\omega^*(t) = v(t)\sin(\delta), \tag{6-7}$$

with $v^*(t)$ the longitudinal component of the tyre velocity $v(t)$, and $d_0\omega^*(t)$ the lateral component. These two trigonometric relationships can be used to determine the high level control of the two vehicle inputs. Combining these two equations gives the control input signal:

$$\delta_{\text{in}}(t) = \arctan(\frac{d_0\omega^*(t)}{v^*(t)}). \tag{6-8}$$

Here, d_0 is the distance from the Centre of Gravity (CoG) of the truck to the front axle, as can be seen in Figure 2-2. Notice that this input signal is based on $\omega^*(t)$ and $v^*(t)$. So in order to achieve a continuous steering control input (which is one of the goals in this research), these two functions should also be continuous. From Equations (6-6) and (6-7) it can be noticed that the two equations are a function of $v_{\text{ref}}(t)$ and $\dot{\psi}_{\text{ref}}(t)$. Now it becomes clear why the reference velocity profile has been designed such that it is C^2 (see Section 4-4) and the spline coefficients have been chosen such that the trajectory has a continuous reference yaw rate (see Sections 4-2 and 5-1).

Since the tyre velocity $v(t)$ is described by the longitudinal component $v^*(t)$ and the lateral component $d_0\omega^*(t)$, the control input for the velocity can be derived by:

$$v_{\text{in}}(t) = \sqrt{v^*(t)^2 + (d_0\omega^*(t))^2}. \tag{6-9}$$

Thus two vehicle inputs that are derived such that tracking error is minimized and the signals are bounded while being computationally efficient. This gives the opportunity to perform closed-loop simulations to analyse how the kinematic truck-trailer vehicle tracks a generated trajectory.

6-3 Vehicle simulations

The trajectory generators are analysed briefly in Chapters 4 and 5 using Single Lane Change (SLC) manoeuvres. This same manoeuvre will be used to investigate how well the trajectory is tracked using the kinematic vehicle model described in Chapter 2 and the controller proposed in [42]. This section contains trajectory tracking simulations of the lane change with different manoeuvres. It is of interest to analyse results with different lane-change times or inter via-points distance, because it is yet unknown what these should ideally be. In the literature the SLC manoeuvre is only described as a sinusoid steering input, but not what the geometric shape of the trajectory or inter via-point distance should be. Besides, it is interesting how the spline-based methods for trajectory generation described in Chapters 4 and 5 perform under different circumstances. The generation is fast, so reducing the distance between via-points should not cause a problem because of computation time. But this may perhaps cause high reference yaw rates.

See Figure 6-3 for the diagram of the SLC manoeuvre along with its parameters.

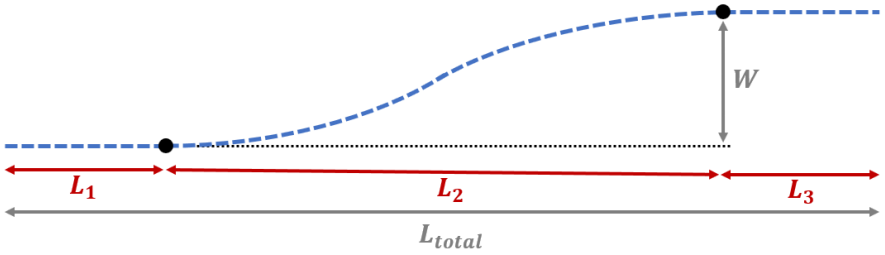


Figure 6-3: The single lane change manoeuvre along with its parameters.

Here the path associated with the SLC is shown as a dotted blue curve. Parameter W is the lateral translation of the lane change, and L_{total} the total length in which the manoeuvre must be executed. Simulations start by driving forward in the x-direction for a length of L_1. L_2 is the distance in which the lateral movement is performed. The manoeuvre ends with driving forwards again for a distance equal to L_3. The parameters W and L_{total} are equal to 3.5 [m] and 80 [m] respectively in the simulations. These are kept constant, while parameters L_1 to L_3 are varied in order to test the performance of the trajectory generator (and tracker) under different circumstances. For instance, increasing L_2 while reducing L_1 and L_3 gives more room for the lane change, but less time to stabilize the vehicle motion at the end of the manoeuvre.

Each simulation tests how well the trajectory can be tracked using spline-based trajectory generation. One of the performance criteria is the Root Mean Square Error (RMSE) between the vehicle state (position, heading angle and velocity) and the reference state. The RMSE computes the average error between two states, and is done by Equation (6-10).

$$\text{RMSE}_s = \frac{\sqrt{\sum_{k=0}^{n}(s_v(t_0 + k\Delta t) - s_{\text{ref}}(t_0 + k\Delta t))^2}}{n} \quad \text{, with } \Delta t = \frac{t_F - t_0}{n} \tag{6-10}$$

Here s_v is the vehicle state and s_r the reference state. Furthermore, it is also of interest to see the largest error measured for each vehicle state such that sudden peaks in error can be observed. Additionally, comparing the generated vehicle inputs ($\delta_{\text{in}}(t)$ and $v_{\text{in}}(t)$) of the simulation runs gives insight on the continuity of these signals. It is also of interest to analyse how cubic splines perform. In Chapter 4 it was found that the reference trajectory using cubic splines had high yaw rate peaks. Analysis can be performed whether these yaw rate peaks cause problems when tracking the trajectory. Specifically for truck-trailer vehicles it is also interesting to analyse the trailer trajectory for each of the mentioned simulations. While this section contains the results of all simulation, an extensive discussion is provided in Section 6-4. A summary of all simulation parameters and results is shown in Table 6-1.

6-3-1 Baseline simulation

The first simulation run is the SLC manoeuvre using quintic splines as is shown in Figure 5-5. Here, $L_1 = 21$ [m], $L_2 = 37$ [m] and $L_3 = 22$ [m]. The linear distance between via-points is approximately 7 [m]. Obviously, the segment length is not equal to this distance, but is based on the reference trajectory. This reference trajectory has a maximum yaw rate of approximately 0.7 [rad/s] using a trade-off value of $\alpha = 0.4$ (see Equation (5-15)). The reference velocity varies between 30 and 31.5 [km/h] along the manoeuvre, with each segment

containing an acceleration below 0.3 [m/s^2] such that the reference velocity profile can be followed by a truck-trailer vehicle according to [35].

The controller parameters have been tuned manually until satisfactory results are achieved. Of course, better results can be obtained if more time is spent on the tuning of the controllers, or perhaps a method is found and applied to finding the most optimal parameters.

Figure 6-4 shows the vehicle input signals used to track the trajectory. Appendix B-1, Figure B-1 shows the position tracking performance and B-2 shows the velocity tracking performance. The tracking performance in terms of RMSE and maximum error is shown in Table 6-1 together with all the information of the simulation, such as controller parameters and road information.

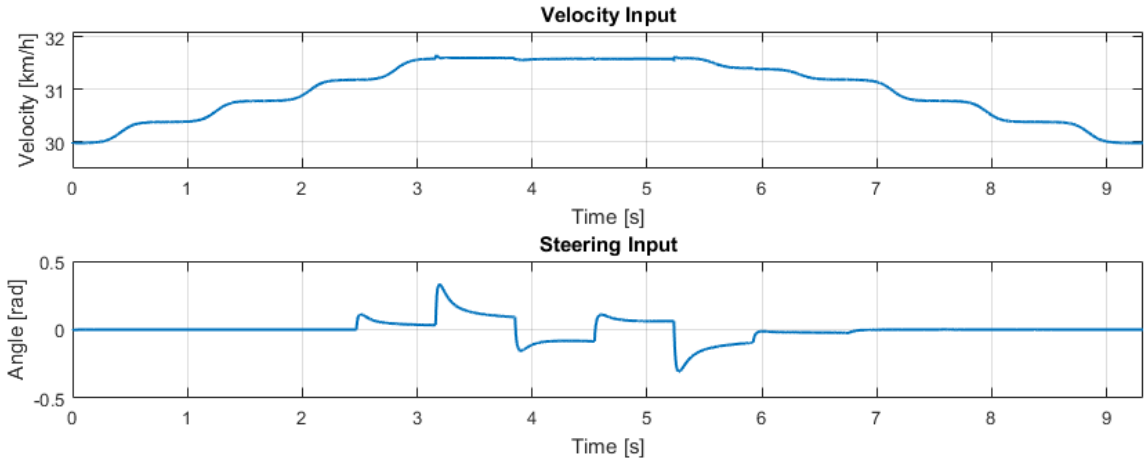


Figure 6-4: Vehicle input signals in the baseline simulation.

6-3-2 Varying via-point distance simulations

In the baseline simulation the via-points are at a distance of approximately 7 [m] from each other. This has been chosen such that it takes roughly 1 second to traverse each segment. The reason behind this is that a new trajectory must be generated every 1 second, so it is not computationally heavy. Furthermore, the via-points should still be close enough to each other such that a manoeuvre is well defined to apply short-term trajectory generation. However, no literature has been found on how long each segment should be for trajectory generation for vehicles, so this 1 second rule of thumb cannot be backed up by other studies. Therefore it is of interest to analyse the flexibility of the trajectory generator and tracker when via-points are closer to or farther apart from each other.

As a result, two additional simulations using quintic splines have been run: with a inter via-point distance of approximately 5 [m] and 10 [m]. It seems that the controller parameters did not require retuning in order to achieve better results, so these are the same as in the baseline simulation. Furthermore, parameters L_1 to L_3 are also the same as before. For the simulation with a segment length of 10 [m], trade-off value $\alpha = 0.6$ is chosen such that oscillations are relatively small. This results in a reference trajectory with a maximum yaw rate of 1 [rad/s]. The simulation with segment length of 5 [m] used $\alpha = 0.4$, which leads to a maximum yaw rate of approximately 0.3 [rad/s].

Running the close-loop simulations leads to input signals shown in Figures 6-5 and 6-6. The tracking performance is shown in Figures B-3 to B-6.

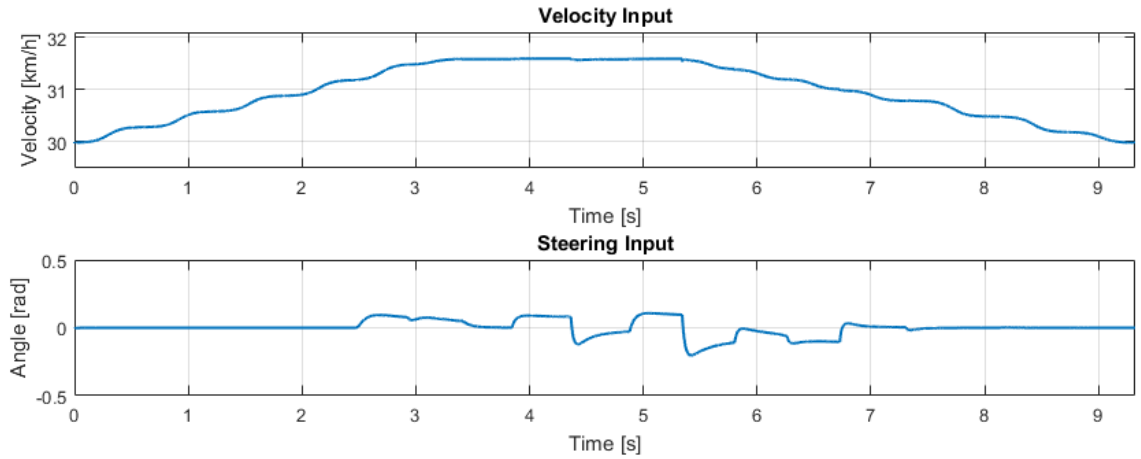


Figure 6-5: Vehicle input signals in the simulation with closer via-points.

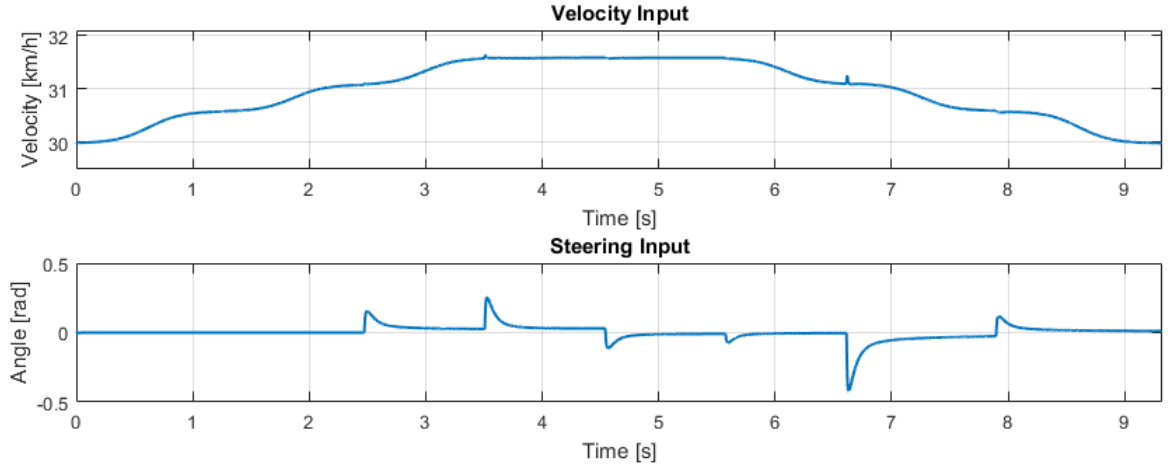


Figure 6-6: Vehicle input signals in the simulation with farther apart via-points.

6-3-3 Longer lane change manoeuvre simulation

The baseline simulation described in Section 6-3-1 performs the lateral movement of the lane change in 37 [m]. Some significant peaks in yaw rate are obtained in this manoeuvre. It is of interest to analyse whether smaller peaks are obtained with a longer lane change manoeuvre. In this case, what would happen if parameter L_2 changes to approximately 50 [m]? Obviously, the vehicle has more time to perform the lateral movement, but less space to stabilize at the end of the manoeuvre (at L_3).

The simulation with a larger value for L_2 is conducted with once again the same controller parameters as before. Quintic splines are used with trade-off value $\alpha = 0.5$, leading to a

stable reference trajectory with a maximum yaw rate equal to 0.5 [rad/s].

The results of the input signals is shown in Figure 6-7, and the tracking performance is shown in Appendix B-4.

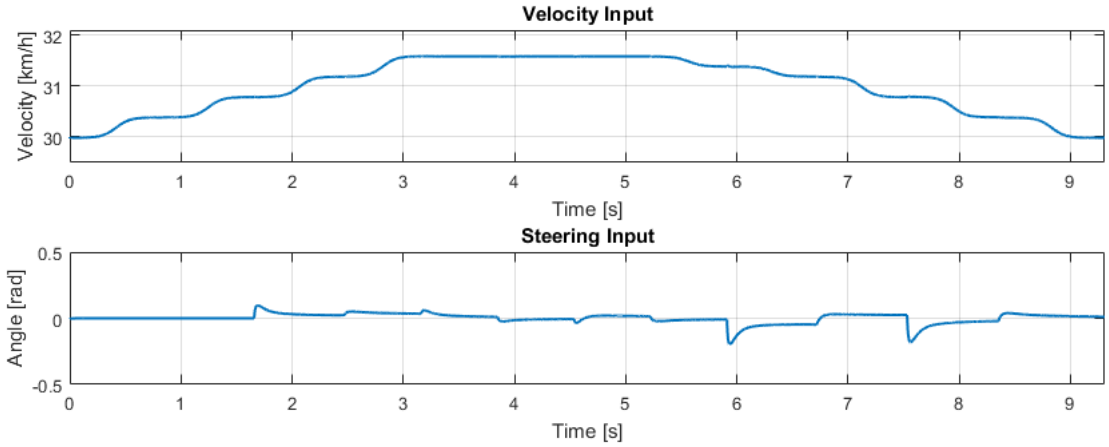


Figure 6-7: Vehicle input signals in the simulation with a longer lane change manoeuvre.

6-3-4 Cubic splines simulation

Whereas previous simulations used quintic splines, it is also interesting to analyse a closed-loop simulation using cubic splines. The reference trajectory has a maximum yaw rate of 1.7 [rad/s], and simulations will show how this affects the tracking performance. Using the same road parameters L_1 to L_3 and controller parameters as the baseline simulation described in Subsection 6-3-1, Figure 6-8 is obtained as the control input signals.

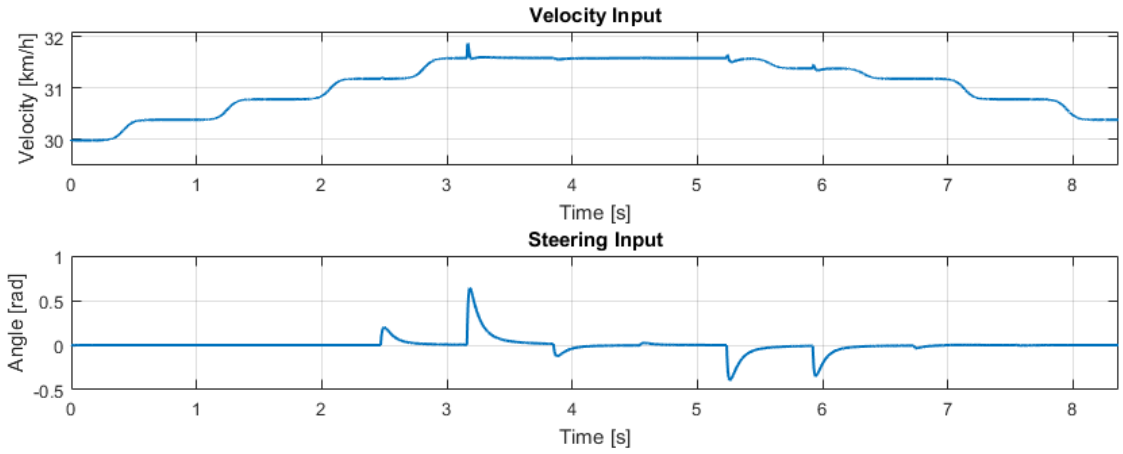


Figure 6-8: Vehicle input signals in the simulation using cubic splines trajectory generation.

6-3-5 Trailer Trajectory

The controller is set up such that the truck is tracking the trajectory, and the trailer is passively following the truck. Since no reference trajectory is defined for the trailer, the average or maximum error cannot be calculated. However, the trailer trajectory for each simulation can be compared to each other. Figure 6-9 shows the reference trajectory for the truck as a dotted blue curve, and the trailer trajectory resulting from the simulations described before.

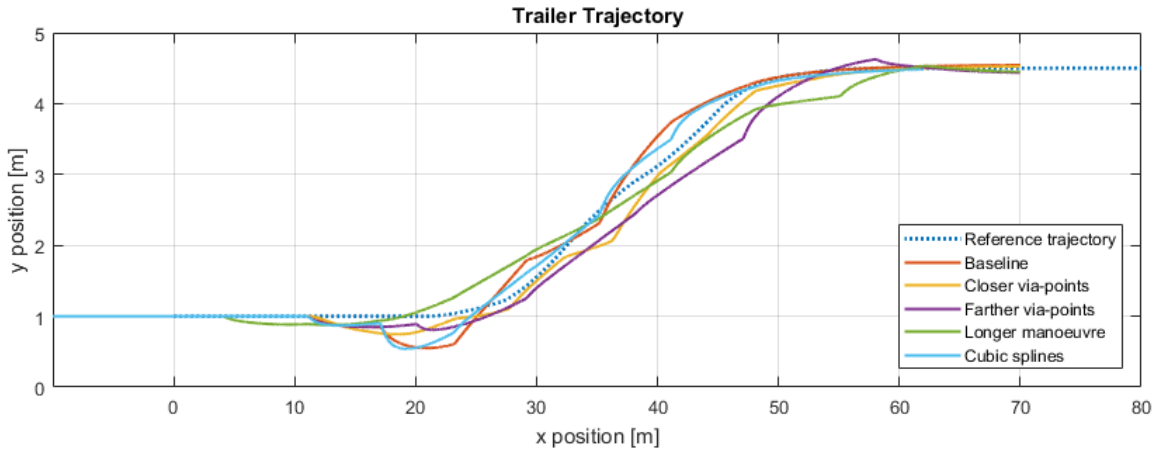


Figure 6-9: Trailer trajectory resulting from all simulations

Table 6-1: Results of simulations tracking spline based trajectories. The five columns show the variations in the performed simulations. The first row block shows the (manually) tuned controller parameters. The second block shows the road information (see Figure 4-1 for clarification). The third block shows important information about the generated trajectory. The last two row blocks show the results regarding tracking performance.

	Baseline	**Close via-points**	**Far via-points**	**Longer manoeuvre**	**Cubic splines**
Controller parameters					
r_x	40	40	40	40	40
r_ψ	8	8	8	8	8
k_x	45	45	45	45	45
k_y	1	1	1	1	1
k_ψ	10	10	10	10	10
c	0	0	0	0	0
Road parameters					
L_1 [m]	21	21	21	14	21
L_2 [m]	37	37	36	51	37
L_3 [m]	22	22	23	15	22
Segment length [m]	7	5	10	7	7
Trajectory data					
Trade-off value α	0.4	0.4	0.6	0.5	-
Maximum yaw rate [rad/s]	0.7	0.3	1	0.5	1.7
RMSE					
x [m]	0.0015	0.0012	0.0011	0.0008	0.0023
y [m]	0.032	0.022	0.018	0.012	0.016
v [m/s]	0.0041	0.0040	0.044	0.0042	0.058
ψ [rad]	0.0025	0.0020	0.0017	0.0013	0.025
Maximum error					
Lateral [m]	0.050	0.037	0.036	0.022	0.0344
Longitudinal [m]	2.6×10^{-4}	1.0×10^{-4}	4.2×10^{-4}	9.6×10^{-5}	9.7×10^{-4}
v [m/s]	0.018	0.018	0.042	0.0075	0.075
ψ [rad]	0.011	0.0074	0.013	0.0066	0.018

6-4 Discussion

It is important to remark that the simulation results presented so far need to be interpreted carefully, since the tracking errors in Table 6-1 can be the result of a number of causes. First of all, the optimal control parameters are most probably not found. There are six parameters that must be tuned, so it is inevitable to obtain errors due to suboptimal tuning; [42] does not provide information regarding the ideal tuning process for their controller. It is only mentioned that these parameters give enough freedom for better suppression of tracking errors. Besides tuning, simulation errors might be obtained because of trade-off parameter α. In the simulations, a value is chosen such that the trajectory is stabilized at the end of the manoeuvre, when the trajectory is supposed to be in the horizontal direction. Some manoeuvres need a higher trade-off value in order to stabilize, thus leading to higher yaw rate peaks in the reference trajectory. This reference yaw rate profile might be hard to track by the truck-trailer model, causing possible tracking errors. Finally, errors could be obtained because the via-point locations are chosen manually. The sequence of these points represent a lane change manoeuvre quite well. But a slight alteration in position of one of the via-points could change the reference trajectory drastically. This could cause the yaw rate peaks to become smaller, or the trajectory could stabilize faster at the end of the manoeuvre without increasing trade-off value α.

However, analysing simulation results can still give insight on the degree of continuity of the eventual trajectory. The most important simulation is the first one, because it shows how the trajectory is tracked under baseline conditions. The intention of this research was to generate a trajectory that will lead to a continuous longitudinal and lateral movement of the truck-trailer vehicle when tracked. The reference velocity profile was defined such that both the velocity and the acceleration are continuous. Furthermore, the spline coefficients are defined using boundary conditions in order to obtain continuous reference tangent direction and yaw rate between segments. Since the high-level control inputs are a function of the trajectory's reference velocity and yaw rate profile, continuous input signals are expected. Figure 6-4 shows that both inputs are indeed continuous. The velocity input signal is still shaped as smooth steps, and the steering input has peaks of 0.3 [rad/s] (approximately 17°), but is still continuous. The trajectory is tracked well according to Table 6-1, with an overall small RMSE and maximum error. This means that even without an ideal controller, the trajectory is easily tracked by the kinematic truck-trailer model.

Furthermore, it was also of interest to compare simulation results to each other to analyse the effects of slightly different manoeuvres. First of all, defining via-points to be closer to each other (segment linear length of approximately 5 [m]) resulted in peaks of the steering input signal equal to a maximum of 0.2 [rad]. This means that the lateral comfort is higher. Moreover, the RMSE is smaller overall, so a trajectory with shorter segments can be tracked with less error. If the via-points are defined to be farther apart from each other, in this case 10 [m], the maximum errors become larger. Furthermore, this change causes the trajectory generator to have problems stabilizing at the end of the manoeuvre, as can be seen in Figure B-5. Even with a relatively high trade-off value, full stabilization cannot be achieved, while still obtaining high reference yaw rate peaks of 1 [rad/s]. Eventually, the simulation with the longer lane change manoeuvre resulted in a steering input with smaller peaks, as is seen in Figure 6-7. This is expected, because more time is given to perform the lateral movement. However, this is at the cost of less stabilization time at the end of the manoeuvre.

Additionally, it is interesting to analyse results using cubic splines compared to the baseline simulation using quintic splines. It can be immediately noticed that using a trajectory generated by cubic splines causes peaks in velocity control input (Figure 6-8). Also, the steering input angle is quite high, reaching values up to 0.6 [rad]. This is most probably caused by the high yaw rate peaks obtained in the reference trajectory. Furthermore, both the RMSE and maximum error is higher using cubic splines. All these results show that the trajectory generated by quintic splines result in better performance compared to cubic splines.

It must be noted that tiny peaks in the velocity input are obtained, even with quintic splines. These can be seen most clearly in Figure 6-6 at approximately $t = 6.6$ [s]. While at first it was thought that this is caused by improper tuning of the controller, it turns out to be otherwise. Looking at Equation (6-9), the velocity input is, among others, a function of $\delta_{\text{in}}(t)$, which itself is a function of $\omega^*(t)$. Equation (6-5) shows that $\omega^*(t)$ is dependent on the reference yaw rate $\dot{\psi}_{\text{ref}}(t)$. Since spline-based trajectory generation cause some peaks in the reference yaw rate at the start of each segment, eventually peaks are obtained in the vehicle input signal. A kinematic vehicle model is used in this research, so it has no problem tracking an input signal with sudden peaks in velocity. However, it would be an issue if a dynamic model was used, with aspects such as load transfer and wheel slip. So in the end, using the controller proposed by [42], high reference yaw rate peaks do not only cause high steering input signals, but also peaks in the velocity input.

All simulations seem to have sudden changing trailer trajectories at the point where two segments join. This sudden change appears to be larger when the reference yaw rate is large. For instance, in Figure 6-9, the light-blue curve showing the trailer trajectory using cubic splines suddenly changes at approximately $x = 17$ [m]. This is the exact moment the reference yaw rate reaches its maximum value of 1.7 [rad/s]. The yellow curve showing the trailer trajectory using smaller inter via-point distance has less sudden changing trajectories, probably caused by the fact that the reference yaw rate peaks are smaller. The trailer seems to be swaying at a maximum distance of 0.5 [m] measured from the reference trajectory. This effect is reduced using quintic splines with closer via-points.

To conclude this discussion:

- The spline-based approaches result in continuous control inputs, thus leading to a continuous motion of the truck.
- Defining smaller inter via-point distance reduces error peaks, at the cost of less computation time.
- The simulation with the longer lane change manoeuvre has both lower average and peak errors, but has troubles stabilizing the vehicle at the end of the manoeuvre.
- Reference yaw rate peaks cause peaks in both the velocity and steering control inputs. This also leads to sudden changing trailer trajectories.

Chapter 7

Conclusions and future work proposals

This chapter summarizes the work presented in this thesis, provides the main conclusions and ends with a list of future research directions.

7-1 Thesis summary

At first an overview was given for planning in general. Soon it became clear that path planners only satisfy the condition of creating a collision free track. The added benefits of trajectory generators would be the fact that path continuity of higher orders can be achieved. Since generators are parametrized by time as well as velocity or acceleration, a custom speed profile can be defined along the trajectory such that continuity in longitudinal direction can be achieved. Furthermore, properly designing the trajectory generator could realize continuity in lateral direction. These two advantages increase passenger comfort levels while enhancing engine performance. Moreover, using a parametric method for trajectory generation would not only make it computationally efficient, but gives the opportunity to prove continuity conditions. A search in literature was conducted, and it was found that cubic splines fit the most accurate to these specifications.

Splines are piecewise polynomials that interpolate via-points. Using cubic splines, it was found that boundary conditions had to be set up in order to reach the desired lateral continuity. Defining conditions on each segment's initial and final position, its initial tangent direction and yaw rate will lead to a smooth steering input to the vehicle. A unique method of parametrization was applied to the splines such that a predefined velocity profile was defined along the trajectory. It seems that this profile needed to be symmetrically shaped in order to be able to estimate the final time for each segment. Computationally efficient numerical integration (RK4) was performed in order to obtain the spline parameter as a function of time such that a trajectory is defined. The generated trajectory was continuous in longitudinal (forward velocity) and lateral (yaw rate) direction, but high yaw rate peaks were observed. Some of these generated trajectories cannot be followed in practice by a vehicle.

The yaw rate peaks have been reduced using quintic splines such that coefficients become available for tuning, without disturbing the achieved continuity. It was noticed that the yaw rate function cannot be minimized as it was a parametrised by time. Using the submultiplicative property of norms, a new objective as a function of parameter u was defined. Searching for the solution is fast and gives drastically lower yaw rate peaks, but at the cost of oscillations in the reference trajectory. Eventually, a linear trade-off is defined such that a choice can be made between minimizing yaw rate peaks and reducing oscillations.

The spline-based trajectory generators were analysed on its performance by simulating a kinematic truck-trailer vehicle tracking the trajectory in closed-loop. Even though a kinematic model does not fully represent a real-life vehicle, it does give the opportunity to investigate the non-holonomic constraint of vehicles and the trailer behaviour. Using the control input proposed by [42], continuous lateral and longitudinal movement is obtained in simulations, with small tracking errors. However, it was noticed that yaw rate peaks have an effect on the steering and velocity control inputs and cause sudden changing trailer trajectories. Furthermore, results are highly dependent on via-point location. It all depends on where the path planner decides to set via-points.

7-2 Conclusions

To conclude this thesis:

- The parametrization applied to the splines gives the opportunity to define a custom velocity profile along the trajectory. The downside is that this profile must be symmetrically shaped for each segment.

- The generated trajectory is continuous in position, tangent direction, and yaw rate. It seems that continuity is not enough to obtain a fully comfortable motion of the vehicle, since the reference yaw rate could become high at the start of the segment. Optimization is utilized with quintic splines to reduce the reference yaw rate.

- Closed-loop simulation were performed using a kinematic truck-trailer model. Such model represents a real-life vehicle on low speeds with small cornering manoeuvres. Therefore a SLC manoeuvre with a reference velocity of approximately 30 [km/h] is a proper manoeuvre.

- Although the controller proposed in [42] is not ideal for a truck-trailer, closed-loop simulations resulted in small tracking errors using spline-based trajectory generation.

- The goal was to design a trajectory generator that would lead to a continuous motion of the truck-trailer model in both the longitudinal and lateral direction. This goal has been achieved, but some peaks are observed in steering and velocity control inputs, and sudden changing trailer trajectories. It seems that one of the causes are yaw rate peaks in the reference trajectory.

7-3 Future work proposals

While the method for trajectory generation proposed in this thesis gives solutions to multiple problems, some issues must be looked at in future works. These proposals do not only improve the method, but are advancements to eventually implementing the generator to real-life truck-trailer vehicles.

- *Analyse spline behaviour.* What order of splines should be used? Would perhaps optimizing a 7th order spline (coefficients a_7 and b_7) give different results. Also, which coefficient should be used for the final position boundary condition (here a_3 and b_3), and which one for the yaw rate peak minimization (here a_5 and b_5)? Investigate why not the other way around.
- *Improve velocity profile.* Here a Sigmoid function is used to define a smooth step velocity profile. However, this method is not the most efficient way of speeding up. In simulations it takes approximately 2 seconds to increase the vehicle velocity from 30 to 31 [km/h]. The Sigmoid function has been used because of its symmetrical property. Future studies could look into methods of defining an efficient velocity profile by avoiding it to be symmetrical.
- *Input information.* It must be analysed how realistic the input information is. How fast can the path planner decide the next via-point, because better simulation results are found with shorter segments. Furthermore, can the desired velocity on the next via-point be determined while keeping maximum vehicle accelerations in mind?
- *Extensive Trajectory Tracker and Dynamic Vehicle Model Design.* Assess the proposed trajectory generator using a trajectory tracker specifically designed for truck-trailer models. Moreover, analyse vehicle response using a dynamic model with load transfer, wheel slip and other systems included.
- *Trailer trajectory analysis.* The trailer movement can be analysed more in depth, such as its angular velocity profile or its heading angle. There is no control on the trailer swaying. Perhaps a separate trajectory could be generated for the trailer. It is also possible to add an actively controlled hinge between truck and trailer.

Appendix A

Parametric and Geometric continuity

There are two different notions regarding smoothness to k^{th} order. Firstly, parametric continuity is written as C^k, and defines smoothness of both the curve and its parametrization. In the case of trajectory generation, C^1 continuity represents continuous path and its time derivative (velocity), C^2 adds continuous acceleration, and this goes on for higher values of k.

The second type of continuity is geometric continuity, and is denotes as G^k. According to [44] it is the relaxed form of parametric continuity independent of parametrizations, but still sufficient for geometric smoothness of the curve. For illustration, consider two curve segments. G^1 implies that the two curves share a common tangent direction at the joint point. In addition, the two curves meet with G^2 continuity if and only if they also share a common curvature at the joint point. In the case of η^3 and η^4 splines (discussed in [29] and [8] respectively), geometric continuity of third and fourth order respectively is achieved. For G^3, the trajectory has a continuous tangent vector, curvature and its derivative along the arc length, and G^4 adds second order curvature derivative with respect to arc length.

In [23] it is stated that two segments meet with G^n continuity if and only if there exists a parametrization such that these two segments meet with C^n continuity at the joint. In this research Section 4-3 discusses the method of parametrization such that a velocity profile (here at least C^2 continuous) is followed. Therefore two segments can meet with G^2 continuity.

Appendix B

Trajectory tracking simulation figures

B-1 Baseline simulation

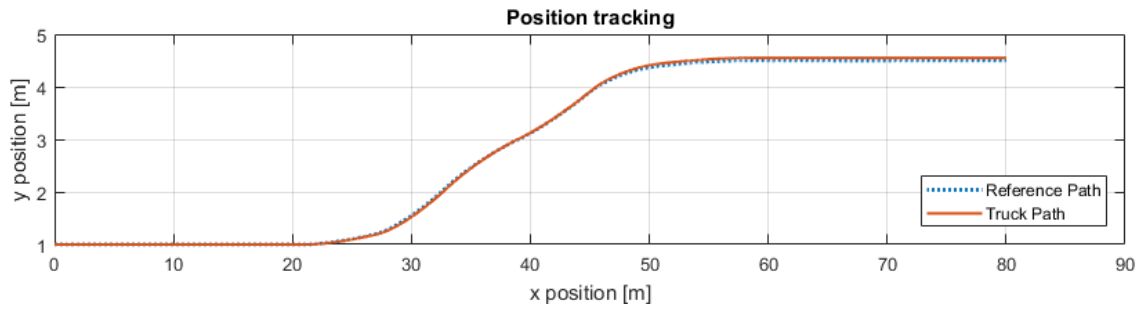


Figure B-1: Position tracking performance in the baseline simulation.

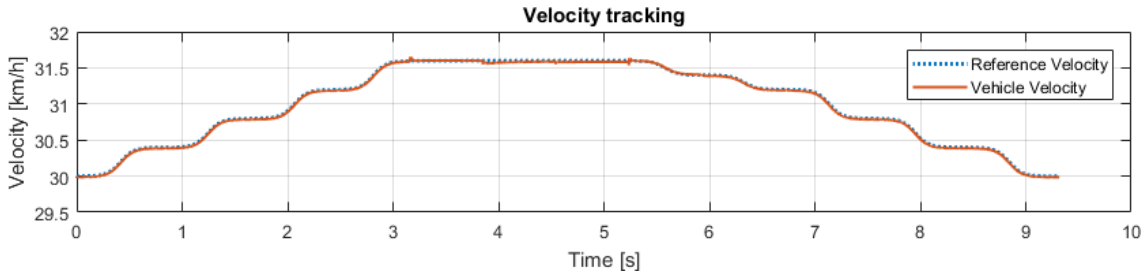


Figure B-2: Velocity tracking performance in the baseline simulation.

B-2 Closer via-points simulation

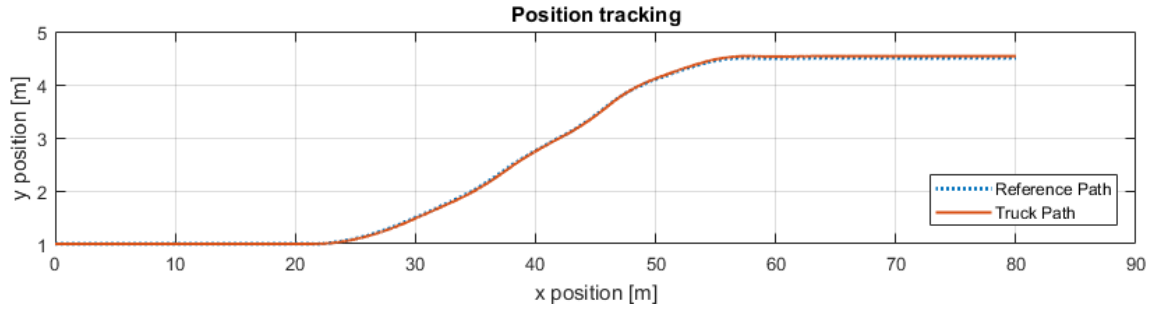


Figure B-3: Position tracking performance in the simulation with closer via-points.

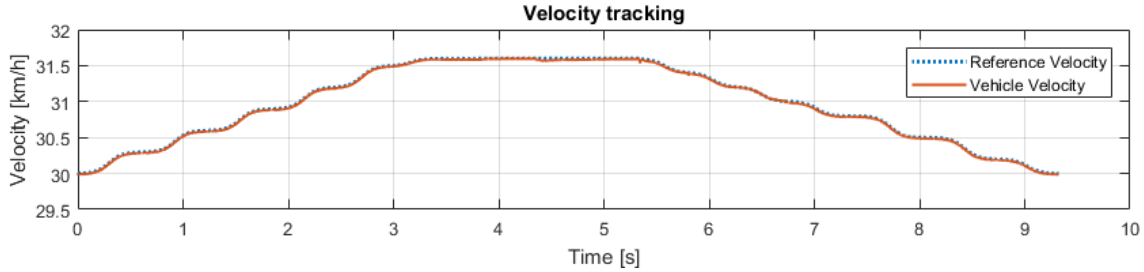


Figure B-4: Velocity tracking performance in the simulation with closer via-points.

B-3 Further via-points simulation

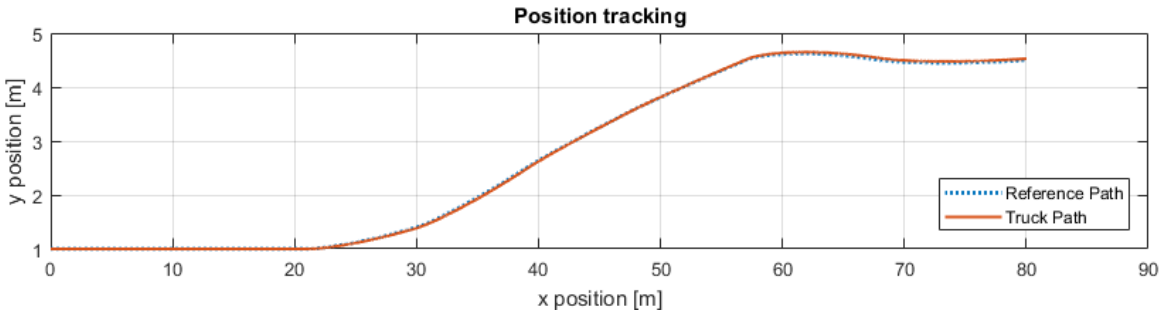


Figure B-5: Position tracking performance in the simulation with farther apart via-points.

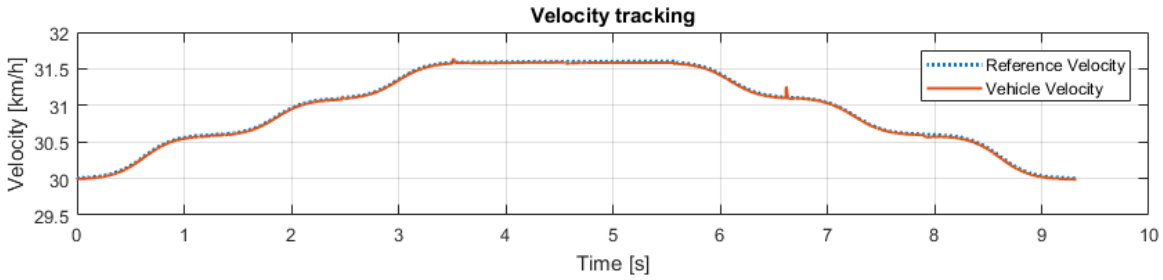


Figure B-6: Velocity tracking performance in the simulation with farther apart via-points.

B-4 Longer lane change manoeuvre simulation

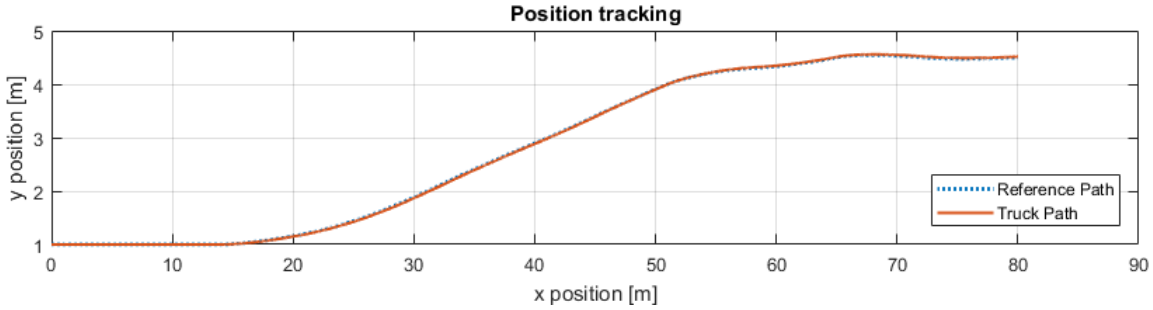


Figure B-7: Position tracking performance in the simulation with a longer lane change manoeuvre.

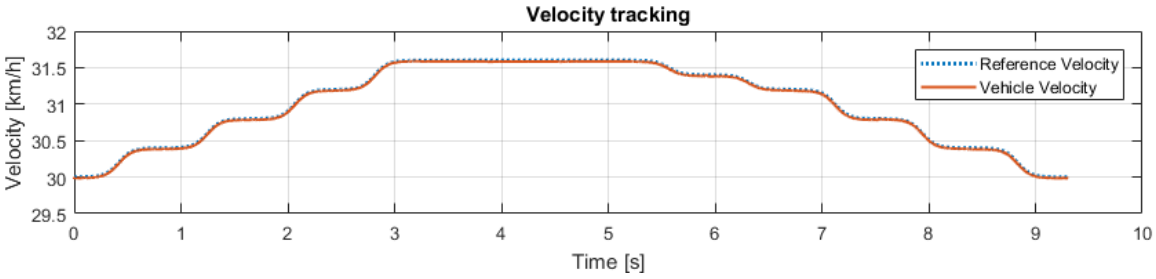


Figure B-8: Velocity tracking performance in the simulation with a longer lane change manoeuvre.

B-5 Cubic splines simulation

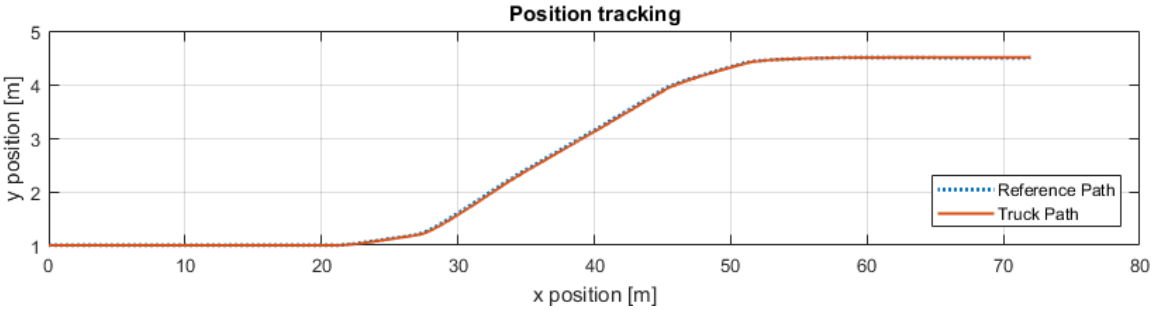


Figure B-9: Position tracking performance in the simulation using cubic splines trajectory generation.

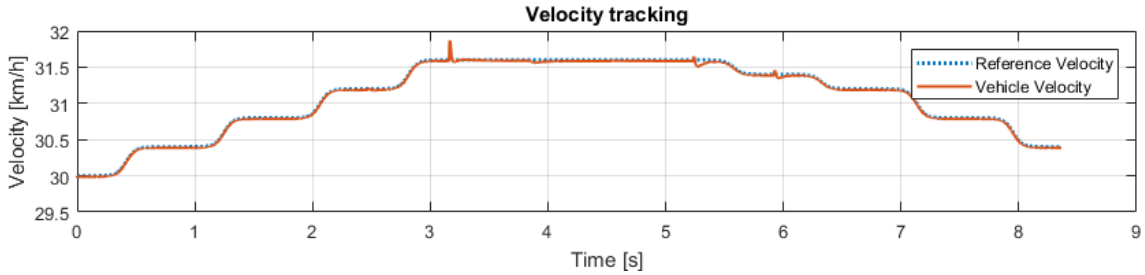


Figure B-10: Velocity tracking performance in the simulation using cubic splines trajectory generation.

Bibliography

[1] "An overview of the eu road transport market in 2015," tech. rep., European Comission, May 2017.

[2] Regulation (EC) No 561/2006, Mar. 2006.

[3] C. Almqvist and K. Heinig, "European accident research and safety report 2013," tech. rep., Volvo Trucks, Jan. 2013.

[4] C. Katrakazas, M. Quddus, W. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, Nov. 2015.

[5] "About - google maps."

[6] S. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," tech. rep., Iowa State University, 1998.

[7] P. Zips, M. Böck, and A. Kugi, "An optimisation-based path planner for truck-trailer systems with driving direction changes.," in *ICRA*, pp. 630–636, IEEE, 2015.

[8] F. Ghilardelli, G. Lini, and A. Piazzi, "Path generation using η^4-splines for a truck and trailer vehicle," *IEEE Trans. Automation Science and Engineering*, vol. 11, no. 1, pp. 187–203, 2014.

[9] M. McNaughton, C. Urmson, J. Dolan, and J. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Proceedings of the International Conference on Robotics and Automation*, pp. 4889–4895, May 2011.

[10] W. Xu, J. Wei, J. Dolan, H. Zhao, and H. Zha, "Real-time motion planner with trajectory optimization for autonomous vehicles," in *ICRA*, IEEE, 2012.

[11] Y. Chong, O. Sawodny, H. Chen, J. Zimmermann, and A. Lutz, "Motion planning for an autonomous vehicle driving on motorways by using flatness properties," in *CCA*, Sept. 2010.

[12] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenèt frame," in *ICRA*, pp. 987–993, IEEE, 2010.

[13] M. Wang, T. Ganjineh, and R. Rojas, "Action annotated trajectory generation for autonomous maneuvers on structured road networks," in *ICARA*, pp. 67–72, IEEE, 2011.

[14] D. González, J. Pérez, R. Lattarulo, V. Milanés, and F. Nashashibi, "Continuous curvature planning with obstacle avoidance capabilities in urban scenarios," in *ITSC*, pp. 1430–1435, IEEE, 2014.

[15] H. Sun, W. Deng, S. Zhang, S. Wang, and Y. Zhang, "Trajectory planning for vehicle autonomous driving with uncertainties," in *ICCSS*, Oct. 2014.

[16] A. Best, S. Narang, D. Barber, and D. Manocha, "Autonovi: Autonomous vehicle planning with dynamic maneuvers and traffic constraints.," *CoRR*, vol. abs/1703.08561, 2017.

[17] V. Delsart, T. Fraichard, and L. Martinez, "Real-time trajectory generation for car-like vehicles navigating dynamic environments," in *ICRA*, pp. 3401–3406, IEEE, 2009.

[18] M. Luijten, "Lateral dynamic behaviour of articulated commercial vehicles," Master's thesis, Eindhoven University of Technology, Aug. 2010.

[19] R. Rajamani, *Vehicle Dynamics and Control*. Springer, 2012.

[20] M. Brezak and I. Petrović, "Path smoothing using clothoids for differential drive mobile robots," *IFAC Proceedings Volumes*, vol. 44, pp. 1133–1138, Jan. 2011.

[21] A. Broggi, P. Medici, P. Zani, A. Coati, and M. Panciroli, "Autonomous vehicles control in the vislab intercontinental autonomous challenge," *Annual Reviews in Control*, vol. 36, pp. 161–171, Apr. 2012.

[22] T. Gu and J. Dolan, "On-road motion planning for autonomous vehicles," in *Proceedings of the 5th international conference on Intelligent Robotics and Applications*, pp. 588–597, May 2012.

[23] L. Biogiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Springer-Verlag Berlin Heidelberg, 2008.

[24] A. Kelly and B. Nagy, "Reactive nonholonomic trajectory generation via parametric optimal control," *I. J. Robotics Res.*, vol. 22, no. 7-8, pp. 583–602, 2003.

[25] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Transactions on Robotics*, vol. 29, pp. 913–929, Aug. 2013.

[26] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, pp. 485–501, Apr. 2010.

[27] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *IROS*, pp. 1879–1884, IEEE, 2009.

[28] D. Madas, M. Nosratinia, M. Keshavarz, P. Sundstrom, R. Philippsen, A. Eidehall, and K. Dahlen, "On path planning methods for automotive collision avoidance," in *IEEE Intelligent Vehicle Symposium*, June 2013.

[29] A. Piazzi, C. G. L. Bianco, and M. Romano, "η^3-splines for the smooth path generation of wheeled mobile robots," *IEEE Trans. Robot.*, vol. 23, pp. 1089–1095, Oct. 2007.

[30] I. Bae, J. Moon, H. Park, J. Kim, and S. Kim, "Path generation and tracking based on a bézier curve for a steering rate controller of autonomous vehicles," in *ITSC*, pp. 436–441, IEEE, 2013.

[31] L. Ma, J. Yang, and M. Zhang, "A two-level path planning method for on-road autonomous driving," in *ITSC*, pp. 436–441, ISDEA, Jan. 2012.

[32] J. Pérez, J. Godoy, J. Villagra, and E. Onieva, "Trajectory generator for autonomous vehicles in urban environments," in *ICRA*, pp. 409–414, IEEE, 2013.

[33] P. Zhao, J. Chen, T. Mei, and H. Liang, "Dynamic motion planning for autonomous vehicle in unknown environments," in *Intelligent Vehicles Symposium*, pp. 284–289, IEEE, June 2011.

[34] R. Pearl and L. Reed, "On the rate of growth of the population of the united states since 1790 and its mathematical representation," *Proceedings of the national academy of sciences*, vol. 6, pp. 275–288, June 1920.

[35] G. Yang, H. Xu, Z. Wang, and Z. Tian, "Truck acceleration behavior study and acceleration lane length recommendations for metered on-ramps," *International Journal of Transportation Science and Technology*, pp. 93–102, 2016.

[36] E. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*. McGraw-Hill, 1955.

[37] C. Runge, *Über die numerische Auflösung von Differentialgleichungen*. Springer, 1895.

[38] M. Kutta, *Beitrag zur näherungweisen Integration totaler Differentialgleichungen*. 1905.

[39] S. Gordon, "On symmetries of polynomials," *PRIMUS*, vol. 9, pp. 13–20, 1999.

[40] F. Dubeau and J. Savoie, "Periodic even degree spline interpolation on a uniform partition," *Journal of Approximation Theory*, vol. 44, pp. 43–54, 1985.

[41] C. Meyer, *Matrix analysis and applied linear algebra*. SIAM, 2001.

[42] D. Kostic, S. Adinandra, J. Caarls, N. V. D. Wouw, and H. Nijmeijer, "Collision-free tracking control of unicycle mobile robots," *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009.

[43] D. Kostic, R. de Lange, H. van den Brand, K. Lekkerkerker, F. Gaisser, P. Jonker, R. Happee, and J. W. van der Wiel, "Driverless, control-intensive," *Mikroniek*, pp. 5–8, 2017.

[44] B. Barsky and T. DeRose, "Geometric continuity of parametric curves: three equivalent characterizations.," *IEEE Computer Graphics and Applications*, vol. 9, no. 6, pp. 60–69, 1989.

Glossary

List of Acronyms

CoG	Centre of Gravity
EU	European Union
RK4	Classical Runge-Kutta
RMSE	Root Mean Square Error
SLC	Single Lane Change

List of Symbols

α	Trade-off value
δ	Steering angle
η^3	Seventh order splines
η^4	Ninth order splines
$\mathcal{O}$	Order of accuracy
ψ	Heading angle of the truck with respect to the x-axis
θ	Heading angle of the trailer with respect to the x-axis
$\hat{p}$	Unit tangent to the trajectory
c	Step location shifter for the Sigmoid function
d_0	Distance from the truck's centre of mass to its front axle
d_1	Distance from the truck's centre of mass to its rear axle
d_2	Distance from the truck's rear axle to the trailer's axle
e	Tracking error
h	Step size
n	Polynomial degree
p	Polynomial degree for splines
p'	Tangent to the trajectory
q	Polynomial function
r	Slope of the step for the Sigmoid function
u	Spline parameter
v	Velocity in the direction of the wheels
x	Horizontal direction on the earth-fixed coordinate system
y	Vertical direction on the earth-fixed coordinate system
L	Length parameter for the lane change manoeuvre
R	Rotation matrix
s	State
$_{av}$	Average
$_{init}$	Initial
$_{k}$	Segment
$_{lat}$	Lateral
$_{long}$	Longitudinal
$_{ref}$	Reference
a_i	Polynomial coefficients in x-direction
b_i	Polynomial coefficients in y-direction
C^n	Parametric continuity of n-th order
G^n	Geometric continuity of n-th order