Convex Optimization-based Design of a Speed Planner for Autonomous Heavy Duty Vehicles

SIMON FRÖLANDER

RAWAN HASAN
Convex Optimization-based Design of a Speed Planner for Autonomous Heavy Duty Vehicles

SIMON FRÖLlander

RAWAN HASAN

Degree Projects in Optimization and Systems Theory (30 ECTS credits)
Degree Programme in Aerospace Engineering (120 credits)
KTH Royal Institute of Technology year 2019
Supervisors at Scania: Rui Oliveira, Mahbub K. Hussain
Supervisor at KTH: Per Enqvist
Examiner at KTH: Per Enqvist
Abstract

The objective of this master thesis is to design a speed planner for a heavy duty vehicle (HDV) that focuses on achieving a time efficient and comfortable driving, given different constraints such as road geometry, vehicle dynamics, acceleration, deceleration, speed limits etc. through an optimization based approach, namely by formulating it as a convex problem in a receding horizon fashion. The thesis begins with considering static environments and henceforth is extended to govern dynamic environments such as taking into account other vehicles and highway merging. The problem is firstly formulated as an optimal control problem and after discretization is reformulated as a Second-order Cone Program (SOCP) problem. By ensuring the convexity of the problem, the results obtained are guaranteed to be globally optimal. The speed planner is implemented in CVX, a modelling system for convex optimization in MATLAB. Results show that the speed planner algorithm yields the performances required, where the speed profiles obtained are smoother with higher weights on the smoothness objective while still retaining travel time efficiency. The algorithm is also extended to avoid dynamic obstacles and adapting the speed of the HDV to the speed of a vehicle driving in front of it.
Konvex optimeringsbaserad design av en hastighetsplanerare för tunga fordon

Sammanfattning

Målet med detta masterarbete är att designa en hastighetsplanerare för ett tungt fordon (HDV) där fokus läggs på att uppnå en tidseffektiv och bekväm färd för olika begränsningar såsom väggeometri, fordonsdynamik, acceleration, retardation, hastighetsbegränsningar etc. genom ett optimeringsbaserat tillvägagångssätt, nämligen genom att formulera det som ett konvext problem i realtid. Arbetet börjar med att hantera statiska miljöer och därefter utvidgas till att hantera dynamiska miljöer såsom att ta hänsyn till andra trafikanter och motorvägspåfart. Problemformuleringen formuleras först som ett optimalt styrproblem och efter diskretisering omformuleras som ett andra ordningens konproblem (SOCP). Genom att säkerställa problemets konvexitet, garanteras de erhållna resultaten att vara globalt optimala. Hastighetsplaneraren implementeras i CVX, ett modelleringssystem för konvex optimering i MATLAB. Resultaten visar att hastighetsplaneringsalgoritmen ger de nödvändiga prestationerna, där de erhållna hastighetsprofilerna är mer bekväma med högre komfortvikter samtidigt som körtidseffektiviteten behålls. Algoritmen utökas ytterligare till att undvika dynamiska hinder och anpassa lastibilens hastighet till hastigheten på ett fordon som kör framför.
Acknowledgements

We would, first and for the most, like to thank our supervisors at Scania, Rui Oliveira and Mahbub K. Hussain, for their continuous effort, guidance and insights that was needed to continue and finish our thesis work. Secondly, we want to show gratitude and appreciation to Burak Demirel and Pedro F. Lima for their help and contributions to our mathematical and implementation parts of the thesis. We would also like to thank our supervisor at the Royal Institute of Technology, Per Enqvist, for always being available for questions and academical insights. Further gratitude and respect goes to our group manager Assad Alam for being a great source of inspiration and encouragement in our career-related questions. Lastly, we want to give a big recognition and praise for our group at Scania for giving us the opportunity to feel like a part of the team by integrating us in the group meetings and guest lectures throughout our time at Scania.

The research presented in this thesis has been supported by the iQPilot 2016-02547 project funded by the innovation and research programme FFI/VINNOVA.
# Contents

1 Introduction ......................................................... 1

1.1 Background .................................................................. 1

1.1.1 Speed Planning in Autonomous Vehicles .................. 2

1.1.2 Related Work ...................................................... 3

1.2 Problem Formulation ............................................... 5

1.2.1 Limitations ......................................................... 5

1.3 Theory ........................................................................ 5

1.3.1 Time Optimal Control Problem .............................. 5

1.3.2 Convexity ............................................................. 7

1.3.3 Second Order Cone Programming ............................ 7

2 Mathematical Formulation ........................................... 9

2.1 Path Representation ................................................. 9

2.2 Optimal Control Formulation ....................................... 10

2.2.1 Objective Function .............................................. 11

2.2.2 Constraints ......................................................... 12

2.2.3 Complete problem formulation .............................. 18

2.3 Convexity ............................................................... 18

2.4 Discretization of Optimal Control Problem ................... 19

2.4.1 Objective Function .............................................. 19

2.4.2 Constraints ......................................................... 20

2.5 Second Order Cone Program Formulation ..................... 21

2.5.1 Time Objective .................................................... 21

2.5.2 Smoothness Objective  ......................................... 22

3 Implementation and Results ......................................... 25

3.1 Parameters and Test Platform ..................................... 25

3.2 Time and Smoothness Objective ................................. 26

3.3 Semi-hard Comfort Limit ......................................... 28

3.4 Time Window .......................................................... 30

3.5 Replanning .............................................................. 31

3.5.1 Vehicle Following ............................................... 31

4 Discussion ............................................................... 35

4.1 Objective Function and Constraint Performances .......... 35

4.2 Replanning and Vehicle Following .............................. 36

4.3 Conclusions ........................................................... 36
Nomenclature

\( A \) Front area of the vehicle
\( K \) Curvature
\( R \) Rotation matrix
\( \alpha \) Longitudinal acceleration - optimization variable
\( \beta \) Longitudinal velocity squared - optimization variable
\( a \) Acceleration in the global frame
\( v \) Velocity in the global frame
\( \mu \) Friction coefficient
\( \omega_1 \) Weight on the minimum time objective
\( \omega_2 \) Weight on the minimum jerk objective
\( \phi \) Inclination angle of the road
\( \rho \) Air density
\( \theta \) Heading
\( a_{c, \text{lat}} \) Comfort limit on the acceleration, lateral
\( a_{c, \text{long}} \) Comfort limit on the acceleration, longitudinal
\( c_d \) Drag coefficient
\( g \) Gravitational acceleration
\( m \) Mass of the vehicle
\( s \) Arclength
\( v_{\text{limit}} \) Combined speed limit
\( w \) Workspace variable, \((x, y)\)
\( x, y, z \) Global position
\( \lambda_1 \) Weight for the semi-hard comfort limit, longitudinal
\( \lambda_2 \) Weight for the semi-hard comfort limit, lateral
\( \sigma_{\text{lat}} \) Slack variable for the semi-hard comfort limit, lateral
$\sigma_{\text{long}}$ Slack variable for the semi-hard comfort limit, longitudinal

c, d, e Slack variables required for SOCP formulation

$u$ Traction force in longitudinal and lateral directions - optimization variable
Chapter 1

Introduction

This chapter will firstly introduce a background section describing the development of autonomous vehicles throughout the recent years and why they have been an important subject in the field of transportation. An overview of different methods and models used in the current speed planning systems is also presented. Lastly, the problem formulation with corresponding limitations and the theory applied in the thesis, will be presented.

1.1 Background

Currently, one of the most discussed topics is the endeavor of introducing fully developed autonomous vehicles (AVs). The interest of introducing fully AVs arose in the recent 100 years, more precisely, as early as 1939 in the United States World’s Fair. One vision of the fair was to discuss new ideas of how to introduce automated vehicles that one day could replace the conventional ways of transportation and how to establish automated highways and platforms in order to facilitate this transition. Some of the many motivations of this idea are primarily to increase the overall driving safety, decrease the greenhouse gas emissions by enforcing the autonomous vehicles to drive more conventional and efficient, to minimize the fuel consumption and costs of congestion and to omit the costs of hiring drivers in the transportation industries, Stanley et al. (2016).

Advancements in the technology have given rise to the creation of the automated vehicles which are either fully self-driven or assists drivers into making more safe decisions. Autonomous vehicles can be categorized into six levels which are further separated into two groups, where group one includes the levels 0-2 and the second group includes the remaining levels, 3-5. Level 0-2 involves vehicles controlled by the driver, but includes driver assistance features and level 3-5 are controlled autonomously, with minimal human intervention. Level 0 is completely controlled by a human driver and level 5 is fully autonomous. Technical and theoretical developments of achieving level five of AVs is still ongoing worldwide, however the technologies that secure driving safety such as adaptive speed controllers, braking systems and other advanced driver-assistance system (ADAS) are commonly used in current modern vehicles, Society of Automotive Engineers (2019).
1.1.1 Speed Planning in Autonomous Vehicles

Speed planning is a part of the greater purpose and is one of the main areas in the field of autonomous vehicles. The fundamental purpose of a speed planner is basically to replace the speed regulating processes that a human driver is exposed to while driving. In order to permit this replacement, several aspects related to the autonomous vehicle and its surrounding areas must be addressed. The chart in fig. 1.1 shows an overall scheme of how different submodules within the autonomous vehicle system is related to each other.

![Figure 1.1: Organizational-system chart for autonomous vehicles.](image)

The perception and localization module collects information regarding the surroundings of the vehicle by exploiting sensor and map data. The mission planning module makes a long term plan of the whole mission, in other words, plans a route from the start position to the desired destination. In the next step, the motion task and the world representation is sent to a path planner that plans a much more precise path for a given horizon ahead of the vehicle. This planner will make sure to avoid obstacles that the mission planning module did not have any information about. Afterwards the path generated by the path planner is sent to a speed planner that will create a speed profile, that contains a desired speed for each waypoint on the path. The goal of the speed planner can vary greatly dependent on the criteria that is needed to be achieved, e.g. it could be to minimize the travel time, to minimize fuel consumption, to create a comfortable ride for the passenger, or a combination of those. Finally the path together with the speed profile are sent to a feedback controller that tries to track the given plan.

A path together with a speed profile is defined as a trajectory. The field of trajectory planning is well exploited and there exists a big variety of methods to solve the problem. Roughly, one could categorize speed planning methods into two categories, coupled and decoupled speed planning.
The coupled speed planning methods aim to solve the general trajectory planning problem directly, including both path and speed whereas the decoupled speed planning methods, initially finds a path and adapts a speed profile afterwards, as described above.

The coupled methods that consider both path and speed have the advantage that they can adapt to a dynamic environment more efficient since they work in both space and time domain. However, many of these algorithms become incredibly large and requires a lot of computational power. By dividing the problem in two smaller problems as done with the decoupled methods, the complexity of the overall problem can be drastically reduced. Since it is of main importance to be able to run the planner in real time, decoupled approaches are often chosen.

### 1.1.2 Related Work

Previous research in the area of speed planning for autonomous vehicles, given different paths, shows great knowledge in how to tackle static environments as well as dynamic environments. Here, static environments are considered as the surroundings that do not change over time, whereas dynamic environments are surrounding that are exposed to constant change where it is highly important to act rapidly in order to avoid critical incidents such as collisions, side-slip or rollover.

Articles that are related to this subject will be presented in this section, to show an overview of the methods and theories used to obtain the current speed planners. Articles that falls within the coupled and decoupled speed planning are respectively presented in the first and second section.

One article that is not directly related to the speed planning subject, but has had contributions to the modelling of the curve and speed limitations is presented in Chu et al. (2018). Considering that side-slip and rollover on curvy roads are the main reasons of accidents, Chu et al. (2018) present a study of a model that combines the vehicle dynamics with the road states to obtain the acceleration and deceleration limitations to maintain a safe travel. The advantage of this model is that it predicts which velocities the vehicle should stay within in order to avoid accidents on curvy roads.

**Coupled speed planning**

To minimize the complexity and maximize the computational efficiency, speed profiles are often based on simplified strategies by assuming constant acceleration or speed. These assumptions mainly result in low driving comfort. Heinrich et al. (2018) presents three methods to smooth out the speed profile of a trajectory determined by a graph-based A* algorithm while still assuming constant acceleration and achieving spatiotemporal consistency i.e. to obtain consistent solutions w.r.t. space and time. The first method generates a smooth speed profile by a polynomial based heuristic method that provides a continuous acceleration while the second approach is based on numerical optimization where minimization of the jerk, the third-order derivative of the position, and the steering rate are desired. The third approach generalizes the second approach and expresses the speed, acceleration and jerk as polynomials which in turn are the optimization variables.
Decoupled speed planning

Similar to Heinrich et al. (2018), Wang et al. (2018) proposes a study that analyzes three different methods of how to design a speed planner that mainly considers passenger comfort and computational efficiency for real-time computations. The objective of the three different optimal control based methods was to minimize the trip time, the norm of the jerk and the square of the jerk. The results of the objectives demonstrated that the minimum square of the jerk obtained the smoothest speed profiles while the minimum time lacked comfort, however, was the fastest to complete the task.

Interactions between drivers, decision making and courtesy behaviour is an essential topic in the field of automated vehicles. Menéndez-Romero et al. (2018) presents a study of the driving behaviour between drivers and highly automated vehicles in highways. The objective of the study is to predict and adapt to the behaviour of other traffic participants by a courtesy behaviour based system. The predictions are determined by a so called Gentle Boost classifier and Monte Carlo samplings that gives the probability of the behaviour outcomes that are unlikely to happen while the decision making approach is based on a utility function where safety and comfort are the cost functions. The results of applying this model into speed planners show an improvement in the interactions between the automated vehicles and other traffic participants as well as reduction of collisions.

Lima et al. (2015) has developed a decoupled speed planner and longitudinal controller using a clothoid-based path approach with the motivation of describing the nonholonomic movement of a vehicle as a set of clothoids. The problem is formulated as a convex optimization problem with the purpose of producing a smooth speed profile with an objective function where the acceleration and the difference between the vehicle velocity and the maximum velocity allowed are minimized. The longitudinal controller solves the optimal solution in a receding horizon fashion. Results show that, as the penalty factor in the acceleration term increases, the smoother the speed profile becomes.

In Held et al. (2018) a speed planner is designed to minimize the fuel consumption of a heavy duty vehicle in urban driving, where the environment is exposed to constant change. The problem is formulated as an optimal control problem (OCP) that is solved within a certain update frequency using a model predictive control (MPC) framework in order to have the information needed to adapt to the surrounding changes throughout the path.

One common approach to speed planning for a fixed path is to model the problem as a minimum-time optimal control problem. Verscheure et al. (2008) presents this framework by essentially parameterizing the path as a single variable where a trajectory planning model is made for robots. The problem is formulated as a nonlinear optimal control problem which is reformulated and solved as a convex second order cone program (SOCP) which in turn yields global optimal solutions. Both Lipp and Boyd (2014) and Zhang et al. (2018) base their studies on the framework presented in Verscheure et al. (2008). Lipp and Boyd (2014) gives a general formulation of the optimal control problem that can be used depending on which type of vehicle is modelled. The speed profiles along the new path representation can be determined once the formulation of the vehicle dynamics and the road constraints are made. The decoupled speed planner in Zhang et al. (2018) base their methods similarly to Verscheure et al. (2008) and Lipp and Boyd (2014), but extends the study to include comfort box-and time window constraints. Other features included are friction and acceleration limitations and vehicle dynamics.
1.2 Problem Formulation

The studies described in the previous section, shows great development in the area of speed planning. [Zhang et al. (2018)] is one of the more significant articles for this thesis, where the speed planner governs multiple interesting features and performances, thus it has had a great contribution to the approaches made in the thesis. One main goal to accomplish in this thesis is to design a speed planner that firstly, provides comfortable and time efficient travels with respect to the road constraints, the vehicle dynamics and dynamic obstacles in the surroundings and secondly, is able to be used in a receding horizon fashion (real-time) and lastly to being able to adapt the speed of the HDV to the speed of a vehicle driving in front. This leads to the thesis objectives defined below:

- Design a decoupled speed planner, based on a convex optimization approach, for autonomous heavy duty vehicles that satisfy the requirements of [ISO (1997)] standard of a comfortable ride for a human to experience while still retaining time efficiency.
- Is computationally efficient enough to be able to run in real time.
- Design an algorithm that adjusts the speed of the heavy duty vehicle to the velocity of vehicles in front of it.

1.2.1 Limitations

The main limitation of this thesis is the chosen approach to use a decoupled speed planner. To use a decoupled path and speed planner makes it more difficult to adapt to a dynamic environment however, reduces the complexity of the overall problem compared to methods that plan both path and speed at the same time. Since computational efficiency and the ability to run the planner in real time is of main importance in this thesis, the decoupled approach is thus chosen.

Other limitations set are that, the vehicle is assumed to drive on a single lane with no intersections and with no gear changes taken into account. These assumptions are primarily made upon the motivation to reduce the complexity of the vehicle dynamics formulation and to achieve the requirements set within the time scope employed for this master thesis.

1.3 Theory

The following section will present the fundamental theory that has been the backbone of the mathematical formulations in the thesis.

1.3.1 Time Optimal Control Problem

Optimal control is a systematic tool that has been widely used since the space race in the 1950s. It has successively been widespread to other fields such as autonomous systems, economics and aeronautics. The method is systematic in a way that it facilitates the work of control design and has the advantages of finding optimal solutions subject to the system dynamics and the cost criterion. [Ulf T. Jönsson and Ögren (2010)] An optimal control problem can be formulated
in continuous-or discrete-form and includes the System dynamics, Boundary conditions, Control variables and Cost function. A description of the categories is presented below.

**System dynamics:** is the equation that describes the dynamics of the system in state space form, i.e. a description of how the system changes over time. The general formulation of the system dynamics is often defined as a first order differential equation as shown below,

\[ \dot{x} = f(t, x, u). \] (1.1)

Where the functions \( f \in \mathbb{R}^n \) are the vector fields that gives the descriptions of the system in each state \( x \in \mathbb{R}^n \) and \( u \in \mathbb{R}^m \) are the control variables.

**Boundary conditions:** The state vector is constrained by the initial and final times, \( t_i \) and \( t_f \), respectively. A representation of the boundary conditions is shown below,

\[ x(t_i = 0) = x_i \]
\[ x(t_f = T) = x_f. \] (1.2)

**Control variables:** The variable \( u \) in eq. (1.1) is the unknown variable that is desired to be controlled w.r.t. the system and belongs to the set \( U \in \mathbb{R}^m \) and is required to be a piece-wise continuous function.

**Cost function:** The constraints stated above give in general an infinite amount of solutions, thus in order to obtain the optimal/best suited solution to a given problem, a cost function is required. The general formulation of the cost function \( J \) is,

\[ J = \phi(x(t_f)) + \int_{t_i}^{t_f} f_0(t, x(t), u(t))dt \] (1.3)

The first term represents the terminal cost and is added as penalization if there exists a deviation between the desired final state and the actual final state. The second term is the cost that is continuously added as the state changes over time, (Ulf T. Jönnson and Ögren, 2010, page 7-8).

**General time optimal control formulation**

Given the terms described above, the general mathematical formulation of a continuous time optimal control problem can now be presented in eq. (1.4)

\[
\begin{align*}
\text{minimize} & \quad \phi(x(t_f)) + \int_{t_i}^{t_f} f_0(t, x(t), u(t))dt \\
\text{subject to} & \quad \dot{x} = f(t, x, u), \\
& \quad x(0) = x_i, \\
& \quad x(T) = x_f, \\
& \quad u(t) \in U, \\
& \quad u(\cdot) \text{ is piecewise continuous.}
\end{align*}
\] (1.4)
Optimal control problems are typically solved analytically by Pontryagin’s maximum principle (PMP) and numerically by, e.g., dynamic programming (DP), discretization, first order gradient methods, Newton’s method, consistent approximation etc. which are described in Ulf T. Jönsson and Ögren (2010) for further interest.

1.3.2 Convexity

One of the most desired features when solving optimization problems, is to obtain global optimal solutions. This can be obtained by formulating the problem as a convex optimization problem i.e. that the objective function is convex and the feasible region of constraints is a convex set. By definition, for any \( x, y \in S \) it holds that,

\[
\gamma x + (1 - \gamma) y \in S, \quad \forall 0 \leq \gamma \leq 1 \ & \ & \forall x, y \in S. \tag{1.5}
\]

Equation (1.5) is interpreted as, if two arbitrary points \( x \) and \( y \) are in the set, convexity requires that the line connecting these points must also lay in the set \( S \), and the objective function \( f \) is, by definition, only convex on \( S \) if,

\[
f(\gamma x + (1 - \gamma) y) \leq \gamma f(x) + (1 - \gamma) f(y), \quad \forall 0 \leq \gamma \leq 1 \ & \ & \forall x, y \in S. \tag{1.6}
\]

Where the left hand-side of eq. (1.6) is the graph of the function, while the right hand-side is the line connecting the points \( (x, f(x)) \) and \( (y, f(y)) \) on the function. Thus, in order for \( f \) to be convex, the line segment must lay above or on the function graph, (Griva et al., 2011, page 48-49).

1.3.3 Second Order Cone Programming

Second order cone programming (SOCP) is a method that falls within the category of numerical optimization. The SOCP problem is convex and nonlinear and can be formulated as a linear and a quadratic program. The method has been utilized in different engineering areas, in filter design, truss design and a various applications within robotics, Lobo et al. (1998). The general formulation of SOCP problems shown in eq. (1.7) includes, similar to an optimal control problem, an objective function together with the related constraints.

\[
\begin{align*}
\text{minimize} \quad & f^T x \\
\text{subject to} \quad & ||A_i x + b_i||_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m, \\
& Fx = g \quad i = 1, \ldots, m. \tag{1.7c}
\end{align*}
\]

The objective function is linear where the optimization variables are \( x \in \mathbb{R}^n \) and the remaining problem parameters are \( f \in \mathbb{R}^m, \ F \in \mathbb{R}^{m \times n}, \ g \in \mathbb{R}^m, \ A_i \in \mathbb{R}^{(m_i - 1) \times m}, \ b_i \in \mathbb{R}^{m_i - 1}, \ c_i \in \mathbb{R}^m \) and \( d_i \in \mathbb{R}^m \). The second term in the formulation (1.7b) is the second order cone constraint that considers the quadratic equations and is written in the standard Euclidean norm form. An equivalent reformulation of the constraint is stated below.
\[
\begin{bmatrix}
A_i \\
c_i^T
\end{bmatrix}
\begin{bmatrix}
x \\
b_i
\end{bmatrix} \in Q
\]

This shows that the second order cones belongs to the Lorentz cone domain \( Q \) illustrated in fig. 1.2.

Figure 1.2: Illustration of the Lorentz cone.

Given that the objective function is linear and is minimized over the convex second-order cone eq. (1.7b) and affine constraint eq. (1.7c), it is then guaranteed that the SOCP problem is convex and gives global optimal solutions, as described in the convexity section above. SOCP is closely related to linear (LP) and quadratic programming (QP), by setting the left hand-side of eq. (1.7b) to zero, a LP problem is obtained and by setting the parameter \( c_i = 0 \) and squaring the term, a quadratically constrained linear programming (QCLP) problem is obtained. SOCPs are typically solved by primal-dual interior point algorithms, that in recent years have been developed to solve the problem robustly and efficiently, [Lobo et al., 1998] page 194-195.
Chapter 2

Mathematical Formulation

This chapter presents the mathematical formulations used throughout the thesis. The optimal control problem is divided into two sections, the objective function and the constraints. Firstly, a general formulation of the equations used is described followed by the discretization and lastly the reformulation into SOCP form.

2.1 Path Representation

In this thesis, a decoupled speed planner will be designed, i.e. that the path coordinates are given in advance into the speed planner and henceforth is supposed to find the optimal speed profile for the requirements desired. As previously presented, a path can be modeled by various graph-based algorithms such as A* (Heinrich et al. (2018)) or by representing it as a set of clothoid-curves (Lima et al. (2015)). However, one of the primal requirements set on the desired speed planner is that it only should consider the $x, y, z$ coordinates and the heading $\theta$, i.e. the orientation of the HDV, as inputs. Thus a way of solving the problem is to parameterize the path using only a single waypoint representation. Given that the workspace is defined in the Cartesian coordinates as,

$$w = (x, y). (2.1)$$

A way of representing the distance between each waypoint in the workspace is to define an arc-length parametrization $s$ between the points. The representation becomes,

$$w(s) = (x(s), y(s)). (2.2)$$

This gives a change from $\mathbb{R}^2 \rightarrow \mathbb{R}^1$ and eventually facilitates the upcoming work by only working with one parameter instead of two. The position $s$ is given as a function of time, $s = f(t)$ and is spanned throughout the workspace, $s \in [0, s_f]$.

Since the arc length is a function of time, the path $w$ can now also be represented as a function of time,
CHAPTER 2. MATHEMATICAL FORMULATION

\[ w(t) = (x(t), y(t)), \quad t \in [0, t_f] \quad (2.3) \]

The velocity \( v \) and acceleration \( a \) in the Cartesian coordinates is thus the derivative of eq. (2.2) with respect to the arclength \( \cdot' \) and time \( \cdot' \):

\[ v = \dot{w}(s) = w'(s) \dot{f} \quad (2.4) \]

and the acceleration

\[ a = \ddot{w}(s) = w''(s) \dot{f}^2 + w'(s) \ddot{f}. \quad (2.5) \]

Where \( \dot{f} \) and \( \ddot{f} \) are respectively the longitudinal velocity and acceleration of the vehicle’s inertial frame. Given that the heading of the ego vehicle on the path \( w(s) \) is \( \theta \) gives the derivative of the path, \( w(s) \) w.r.t. the arclength as,

\[ w'(s) = \left( \frac{\partial x}{\partial s}, \frac{\partial y}{\partial s} \right) = (\cos(\theta(s)), \sin(\theta(s))) \quad (2.6) \]

Where \( w'(s) \) describes the direction of the speed in the Cartesian coordinates. While the second derivative represents the curvature \( K(s) \),

\[ w''(s) = \left( \frac{\partial^2 x}{\partial s^2}, \frac{\partial^2 y}{\partial s^2} \right) = K(s). \quad (2.7) \]

An illustrative figure of the parametrization is shown in fig. 2.1.

\[ \text{Figure 2.1: Workspace parameterization.} \]

2.2 Optimal Control Formulation

This section will present the formulation of the optimal control problem that contains the functionalities that should be achieved by the speed planner. The problem formulation is categorized
2.2. OPTIMAL CONTROL FORMULATION

according to objective function, semi-hard constraints and hard constraints. Where the terms stated in the objective function are allowed to be violated in order to meet the requirements of the hard constraints that must hold in all cases to retain feasible solutions. One of the constraints that will be introduced later in this chapter, is the comfort box constraint which is considered as a semi-hard constraint. It means that the constraint is defined in the objective function as well as in the constraints with the primal task of staying active to achieve high performances but can be violated in cases where safety is required. A description of the objective function is firstly presented, followed by the hard- and semi-hard constraints.

2.2.1 Objective Function

Minimization of the time spent on roads and simultaneously retaining a smooth and comfortable ride while driving an HDV in long distances, are essential factors to consider when designing the speed planner. These factors consequently give the main goal of the optimization, i.e. to minimize the travel time while simultaneously achieving a smooth speed profile. The following formulations of the objective function are based on the approaches made in the literature. In Zhang et al. (2018), it was shown that one good approach of maximizing the smoothness of the speed profile is to minimize the derivative of the acceleration, the jerk and in Verscheure et al. (2008) a simple formulation of how to minimize the trip time by integrating over the fixed path. Hence, the variables that will be optimized are the longitudinal velocity squared and acceleration vectors along the path, respectively,

\[ \beta(s) = s^2 = \dot{s}^2 \]
\[ \alpha(s) = \ddot{s} = \ddot{f}. \]  \hspace{1cm} (2.8)

Minimum time objective

The time needed to travel over a fixed path is given as an integral,

\[ T = \int_{0}^{s_f} 1dt. \]  \hspace{1cm} (2.9)

However, since the optimization is made over space a substitution of the time variable with the arclength is made, similar to Lipp and Boyd (2014) and is derived below,

\[ T = \int_{s(0)}^{s(t_f)} \frac{1}{s} ds = \int_{0}^{s_f} \frac{1}{\sqrt{\beta}} ds. \]  \hspace{1cm} (2.10)

Minimization of jerk

Having only the minimization of the time in the objective function will lead to a non-smooth speed profile that switches between maximum acceleration and retardation. This problem can
be solved by adding the minimization of the jerk which is the derivative of acceleration eq. (2.8) with respect to time, and can be calculated as

\[ J(s) = \ddot{f} = \ddot{\alpha}(s) = \alpha'(s) \sqrt{\beta(s)} = \frac{1}{2} \beta''(s) \sqrt{\beta(s)}. \] (2.11)

Unfortunately this is nonlinear and non-convex. This is a consequence of the square root of the variable \( \beta \), which makes it not possible to solve given that the problem formulation is needed to be convex in order to find global optimal solutions. Hence, similarly to [Zhang et al. 2018], a pseudo jerk, \( \alpha'(s) \) that is the derivative of acceleration with respect to arc-length \( s \) is introduced and is guaranteed to be convex in the upcoming Convexity section. The pseudo-jerk becomes,

\[ J = \int_0^{s_f} ||\alpha'(s)||^2 ds. \] (2.12)

The objective function can now be presented as,

\[ \min_{\alpha, \beta} \omega_1 \int_0^{s_f} \frac{1}{\sqrt{\beta}} ds + \omega_2 \int_0^{s_f} ||\alpha'(s)||^2 ds \] (2.13)

Where \( \omega_1 \) and \( \omega_2 \) are the weights introduced in each term.

### 2.2.2 Constraints

Since the speed planner is supposed to generate a speed profile for a collision-free and safe trip, it is essential to design a planner that takes the geographical, environmental and legal aspects as well as the vehicle dynamics into consideration. The constraints that will be presented are categorized into two parts, static and dynamic environments. Initially, the constraints that covers the static environments, such as speed limitations, vehicle dynamics, avoiding side-slip and roll-over are presented followed by the Time window constraint that includes dynamic environments such as collision avoidance between vehicle and pedestrians or other traffic participants and further, interactions in highway merging.

#### Velocity and acceleration dynamics

The derivative of the velocity \( \beta(s) \) w.r.t. the arclength \( s \) gives the acceleration \( \alpha(s) \) and is derived as \( \dot{\beta}(s) = 2\ddot{s} = 2\alpha(s)\dot{s} = \beta'(s)\dot{s} \), resulting in eq. (2.14).

\[ \beta'(s) = 2\alpha. \] (2.14)

#### Speed limitations

The constraint on the speed can be given by either, a speed limit set by law, top speed of the vehicle itself, or limitations due to curvature. The speed limit constraint is then defined as

\[ v_{\text{limit}} = \min\{v_{\text{law}}, v_{\text{max}}, v_{\text{curve}}\}, \] (2.15)
2.2. OPTIMAL CONTROL FORMULATION

and results to the constraint on the speed variable $\beta$,

$$\beta \leq v_{\text{limit}}^2.$$  \hfill (2.16)

The speed limitation in a curve, $v_{\text{curve}}$, can be derived from the centripetal force equation,

$$ma_{\text{max}} = \frac{mv_{\text{curve}}^2}{r} \quad \Rightarrow \quad v_{\text{curve}} = \sqrt{\frac{a_{\text{max}}}{K}}$$ \hfill (2.17)

Where $r = \frac{1}{K}$, and $K$ is the curvature defined in eq. (2.7). The parameter $a_{\text{max}}$ is the maximum centripetal acceleration allowed in a curve to avoid side-slip and can be selected dependent on the vehicle properties and the friction and acceleration limitations allowed in lateral direction. The vector $v_{\text{curve}}$ is thus predefined before entering the optimization formulation, however, will be complemented with the comfort box constraint that sets a further limitation on the lateral acceleration, Chu et al. (2018).

Boundary conditions

As stated in the optimal control formulation, it is necessary to define in which state the problem start in. Given that the optimization variables are the speed and acceleration, the boundary conditions then becomes,

$$\beta(s_{\text{initial}}) = \beta_{\text{initial}}$$

$$\alpha(s_{\text{initial}}) = \alpha_{\text{initial}}.$$ \hfill (2.18)

Where the parameters $\alpha_{\text{initial}}$ and $\beta_{\text{initial}}$ can be selected upon the desired speed and acceleration of the vehicle in the initial state. And the velocity at the final state is defined as,

$$\beta(s_{\text{final}}) = \beta_{\text{final}}.$$ \hfill (2.19)

Vehicle dynamics

The motion of the vehicle is non-holonomic i.e. the lateral motion is coupled with the longitudinal motion. Up until now, the focus of the problem formulation has only been on the longitudinal motions. The lateral speeds needs to be explicitly expressed in order to get the full expression of the vehicle dynamics in both lateral and longitudinal directions. Figure 2.2 shows an illustration of the exposed forces on a HDV.
In Zhang et al. (2018), the dynamics taken into account are only the control forces in longitudinal and lateral directions, however, is extended here to cover inclination forces as well as air drag forces in the longitudinal direction. In order to add the inclination force, the inputs to the optimization needs to be extended with a z coordinate, however, will not be parameterized in the same manner as the \( (x, y) \) coordinates, since the vehicle is still moving along the \( \mathbb{R}^2 \) domain but with an altitude included. By writing the force equilibrium, the vehicle dynamics is obtained according to eq. (2.20),

\[
\begin{align*}
    m \ddot{w} &= \left( R \mathbf{u}^T \right)_\text{Control forces} - R \mathbf{m} \sin \left( \begin{bmatrix} \phi \\ 0 \end{bmatrix} \right) - R \frac{1}{2} \rho A c \left[ \begin{bmatrix} \beta \\ 0 \end{bmatrix} \right] .
\end{align*}
\]  

(2.20)

Where \( m \) is the mass of the vehicle, \( \phi \) is the inclination angle, \( A \) is the front area of the vehicle and \( \ddot{w} \) is the acceleration in Cartesian coordinates, as expressed in equation eq. (2.5). The matrix \( \mathbf{u} = [u_{long}, u_{lat}] \) contains the control forces in longitudinal and lateral directions in the inertial frame of the vehicle and is added as an optimization variable in the problem. However, the left hand-side of the equilibrium is expressed in the global Cartesian coordinates, thus the right hand-side needs to be converted in the same manner by using the rotational matrix defined below,

\[
R = \begin{bmatrix}
\cos(\theta(s)) & -\sin(\theta(s)) \\
\sin(\theta(s)) & \cos(\theta(s))
\end{bmatrix}.
\]

Where \( \theta \) is the heading of the car in each point of the path and is derived in equation eq. (2.6), Zhang et al. (2018).

Friction circle

A friction circle constraint represents the maximum allowable acceleration and retardation limitation between the tires and the road in order to avoid roll-over and side-slip. An illustrative figure of how to interpret the constraint is shown below.
2.2. OPTIMAL CONTROL FORMULATION

Figure 2.3 shows three levels of limitations where, the outer circle represents the maximal allowed longitudinal and lateral acceleration for a certain friction coefficient ($\mu$). The second level is called the comfort box, which gives the comfortable acceleration limits for a human to experience. The third level is the inner friction circle which becomes active if the HDV enters a slippery road where the friction coefficient decreases, where the outer constraints needs to be violated to compensate for the safety and thus, results to a diminution of the acceleration in all directions. The inner and outer friction circle constraint can be expressed as eq. (2.21),

$$||u|| \leq \mu mg.$$  \hspace{1cm} (2.21)

Where $g$ is the gravitation constant and $\mu$ is the friction coefficient, thus as $\mu$ increases, a higher control force is required and vice versa. \cite{Zhang et al. 2018}. The comfort box constraint will be described in more detail in the upcoming section.

**Comfort box**

The smoothness term in the objective function will be complemented with the semi-hard comfort box constraint shown in fig. 2.3. Having the comfort box only as a hard constraint can decrease the mobility of the vehicle in cases where it is necessary to increase the deceleration, such as urgent stops where the comfort constraint needs to be ignored. Thus, by defining parts of the constraint in both the objective function and in the constraints gives a trade-off between the mobility and the safety in emergency cases. The comfort box is henceforth alleviated by adding a slack variable in the objective function that describe the violations of the comfort constraint. The terms added in the objective function are expressed below as,

$$\text{minimize } \sigma_{\text{long}, \sigma_{\text{lat}}} \lambda_1 ||\sigma_{\text{long}}(s)|| + \lambda_2 ||\sigma_{\text{lat}}(s)||.$$  \hspace{1cm} (2.22)
Where \( \sigma_{\text{long}}(s) \) and \( \sigma_{\text{lat}}(s) \) are respectively, the longitudinal and lateral slack variables that describe the constraint violations, with their corresponding weights \( \lambda_1 \) and \( \lambda_2 \). Simultaneously, the comfort box constraints are given as,

\[
\begin{align*}
||\alpha(s)|| & \leq a_{c,\text{long}} + \sigma_{\text{long}}(s) \\
\frac{||u_{\text{lat}}(s)||}{m} & \leq a_{c,\text{lat}} + \sigma_{\text{lat}}(s) \\
0 & \leq \sigma_{\text{long}}(s) \\
0 & \leq \sigma_{\text{lat}}(s).
\end{align*}
\]

(2.23)

Where \( a_{c,\text{long}} \) and \( a_{c,\text{lat}} \) are the threshold limitations of the longitudinal and lateral acceleration and retardation, respectively, [Zhang et al. (2018)](#).

**Time window**

Up until now, only static environments have been covered, however, the following constraint formulation will extend the problem into considering dynamic environments as well. By definition and similar to [Zhang et al. (2018)](#), the time window constraint can be stated as below,

\[
t_i = T(s_i) = \int_0^{s_i} \frac{1}{\sqrt{\beta}}, \quad T(s_i) \in W_T = (0, T_U].
\]

(2.24)

Which in some sense is equal to the time objective function, but is here formulated as a constraint that can be manipulated w.r.t the time and state desired. It can be interpreted as, given that the vehicle passes through the state \( s_i \) between the time window \( W_T = (0, T_U] \) where a detection algorithm has predicted that a pedestrian is occupying the road, it is guaranteed that there will be no collision. The constraint can thus contribute in collision avoidance at certain points on the path and enforce the vehicle to arrive at a desired point \( s_i \) at a certain time-point. This is however, only possible if the behaviour of the surrounding dynamic obstacles have been predicted in advance. Time windows can in total be separated into three intervals, shown below,

\[
W_T^1 = (0, t_1], \quad W_T^2 = [t_2, t_3], \quad W_T^3 = [t_4, \infty).
\]

(2.25)

An illustrative figure of how to interpret the time intervals is shown below,
Figure 2.4: Time window representation.

Suppose that two moving obstacles $V_1$ and $V_2$ are occupying the road in the time intervals $[t_1, t_2]$ and $[t_3, t_4]$, thus eq. (2.25) shows the safe time intervals to pass through the road with no collision. For $W_1^T$, the vehicle is enforced to speed up in order to pass the road before $V_1$, $W_2^T$ is to pass the road after the $V_1$ but before $V_2$, which is risky to accomplish. Time interval, $W_3^T$, suggests a solution where the vehicle decelerate to let both obstacles pass through and cross the road afterwards. The feasible time interval $W_T$ is thus a union of three infeasible parts, $W_1^T, W_2^T, W_3^T$, where only the first time interval, $W_1^T$, is convex and consequently makes it problematic to solve all three cases given the convexity-based framework of this thesis.
## 2.2.3 Complete problem formulation

The complete formulation of the time continuous optimal control problem can now be presented in eq. (2.26).

**minimize** \( \alpha, \beta, u, \sigma_{\text{long}}, \sigma_{\text{lat}} \)

**subject to**

\[
\begin{align*}
\beta'(s) &= 2\alpha(s), \quad (2.26b) \\
\beta(s) &\leq v^2_{\text{limit}}(s), \quad (2.26c) \\
\alpha(s_{\text{initial}}) &= \alpha_{\text{initial}}, \quad (2.26d) \\
\beta(s_{\text{final}}) &= \beta_{\text{final}}, \quad (2.26f) \\
0 &\leq \sigma_{\text{long}}(s), \quad (2.26l) \\
0 &\leq \sigma_{\text{lat}}(s), \quad (2.26l) \\
T(s_i) &= \int_0^{s_i} \frac{1}{\sqrt{\beta}} ds, \quad T(s_i) \in W_T = (0, T_U]. \quad (2.26m)
\end{align*}
\]

### 2.3 Convexity

To ensure that the solutions obtained will be globally optimal, it is required that the objective function in eq. (2.26) to be convex where the feasible region containing the constraints is a convex set. Convexity analysis shows that,

**Objective function**

- An integral of a powered absolute value (\( \int ||x||^\psi \) for \( \psi \geq 1 \)) is convex and thus yields a convex smoothness term \( J \) and slack variables of the comfort box \( ||\sigma_{\text{long}}||, ||\sigma_{\text{lat}}|| \), (Bertsekas 2009 page 71).
- Integration of a power function, \( \int x^\psi \) for \( \psi \leq 0 \) and \( \psi \geq 1 \), is convex. Thus makes the minimum time objective term \( T \) convex, (Bertsekas 2009 page 71).
- \( \omega_1, \omega_2, \lambda_1 \) and \( \lambda_2 \) are all \( \geq 0 \) and do not influence the convexity of the objective function.

**Constraints**

- All affine functions (\( ax + b \)) are convex which makes the constraints eqs. (2.26b) to (2.26g) and eqs. (2.26k) and (2.26l) convex, (Bertsekas 2009 page 27).
- All norm functions in \( \mathbb{R}^n \) are convex, thus makes eq. (2.26h) convex, (Bertsekas 2009 page 72).
2.4. DISCRETIZATION OF OPTIMAL CONTROL PROBLEM

- From the standard formulation of a SOCP eq. (1.7), it was shown that all functions written in second-order cone form are convex which further makes eq. (2.26i) and eq. (2.26j) convex.
- The time window constraint, eq. (2.26m), is formulated similarly to the time objective function, which has already been proven to be convex.

The analysis shows that the problem formulation has a convex objective function and the feasible region limited by the constraints is a convex set. This guarantees that the solutions that will be obtained are global optimums.

2.4 Discretization of Optimal Control Problem

To solve the optimal control problem, a discretization of eq. (2.26) is made and will result in an optimization problem.

2.4.1 Objective Function

Assuming that the velocity of the vehicle does not change discontinuously and is piecewise linear in each point \( i = 1, \ldots, n \) and the acceleration is piecewise constant between each point \( i = 1 \ldots n - 1 \), as represented in fig. 2.5.

![Figure 2.5: Piecewise linear velocity and piecewise constant acceleration.](image)

Hence, the following discrete function of the velocity is obtained,

\[
\beta(s) = \beta_i + (s - s_i) \frac{\beta_{i+1} - \beta_i}{s_{i+1} - s_i}, \quad i = 1, \ldots, n. \tag{2.27}
\]

Time objective

By substituting eq. (2.27) into the time optimal objective function eq. (2.10) gives,

\[
T = \omega_1 \int_{s_i}^{s_{i+1}} \left( \beta_i + (s - s_i) \frac{\beta_{i+1} - \beta_i}{s_{i+1} - s_i} \right)^{-\frac{1}{2}} ds
= 2\omega_1 \frac{(s_{i+1} - s_i)}{\sqrt{\beta_{i+1} + \sqrt{\beta_i}}} \tag{2.28}
\]
Which finally yield the discretized form of time objective function represented in eq. (2.29),

$$
T = 2\omega_1 \sum_{i=0}^{N-1} \frac{s_{i+1} - s_i}{\sqrt{\beta_{i+1} + \sqrt{\beta_i}}}. 
$$

(2.29)

**Smoothness objective**

For the smoothness term, finite differences can be used to approximate $\alpha'(s)$, which yields

$$
J = \int_0^{s_f} ||\alpha'(s)||^2 ds = \sum_{i=0}^{N-1} \left| \frac{\alpha(s_{i+1}) - \alpha(s_i)}{s_{i+1} - s_i} \right|^2 (s_{i+1} - s_i).
$$

(2.30)

### 2.4.2 Constraints

The discretization of the acceleration in Cartesian coordinates, eq. (2.5) that is related to the vehicle dynamics constraint eq. (2.20) is differentiated twice with respect to the arclength $s$ by finite differences approximations, given below,

$$
w'(s) = \frac{w(s_{i+1}) - w(s_i)}{s_{i+1} - s_i}. 
$$

(2.31)

And the second derivative by Runge–Kutta approximation,

$$
w''(s) = \frac{w(s_{i-2}) - w(s_{i-1}) - w(s_i) + w(s_{i+1})}{2(s_{i+1} - s_i)^2}. 
$$

(2.32)

The resulting complete discrete problem is,
2.5 Second Order Cone Program Formulation

Equation (2.33) is rewritten in SOCP form according to the general formulation in eq. (1.7) by making the objective function linear and having second order conic and affine constraints.

2.5.1 Time Objective

The objective function has to be linear according to eq. (1.7). Starting with the time part which is solved by introducing a slack variables \( d \) that covers for the non-linearity in the denominator, which will be rewritten as,

\[
\min \beta \quad \text{subject to} \quad \frac{\beta_{i+1} - \beta_i}{s_{i+1} - s_i} - 2\alpha_i = 0 \\
\beta_i \leq v_{\text{limit,}i}^2 \\
\beta_0 = v_{\text{initial}}^2 \\
\alpha_0 = \alpha_{\text{initial}} \\
\beta_n = v_{\text{final}}^2 \\
m(w'' \beta_i + w' \alpha_i) = R \mathbf{u}_i - R mg \sin \phi_i - R \frac{1}{2} A c d_\beta_i \\
\|\mathbf{u}_i\| \leq \mu mg \quad i = 1, \ldots, n, \\
\|\alpha_i\| \leq a_{c,\text{long}} + \sigma_{\text{long,}i} \quad i = 1, \ldots, n, \\
\|\mathbf{u}_{\text{lat,i}}\|_m \leq a_{c,\text{lat}} + \sigma_{\text{lat,}i} \\
0 \leq \sigma_{\text{long,}i} \quad i = 1, \ldots, n, \\
0 \leq \sigma_{\text{lat,}i} \\
T(s_j) = 2\omega_1 \sum_{i=0}^{n-1} \frac{s_{i+1} - s_i}{\sqrt{\beta_{i+1} + \beta_i}} \\
T(s_j) \in [0, T_U].
\]

(2.33)

Introducing another variable \( c \) that covers the square root terms in eq. (2.34b) gives,

\[
c_i \leq \sqrt{\beta_i}.
\]

(2.35)
CHAPTER 2. MATHEMATICAL FORMULATION

And so, the constraint eq. (2.34b) can be replaced by two second-order cone constraints, by firstly writing eq. (2.35) in second order cone form, according to,

\[ 0 \leq \beta_i - c_i^2 \]
\[ 0 \leq 4\beta_i - 4c_i^2 \]
\[ 0 \leq (1 + \beta_i)^2 - (1 - \beta_i)^2 - 4c_i^2 \]
\[ (1 - \beta_i)^2 + 4c_i^2 \leq (1 + \beta_i)^2 \]
\[ \sqrt{(1 - \beta_i)^2 + 4c_i^2} \leq (1 + \beta_i). \]

This gives the second-order cone constraint

\[ \| 2c_i \|_{2} \leq 1 + \beta_i. \] (2.36)

Next, combining eqs. (2.34b) and (2.35) and yield the second order cone of eq. (2.34b)

\[ 1 \leq d_i(c_i+1 + c_i) \]
\[ 4 \leq 4d_i(c_i+1 + c_i) \]
\[ \{ (c_i+1 + c_i) = x \} \]
\[ 4 \leq 4d_ix \]
\[ 4 \leq (x + d_i)^2 - (x + d_i)^2 \]
\[ 4 + (x - d_i)^2 \leq (x + d_i)^2 \]
\[ \sqrt{4 + (x - d_i)^2} \leq x + d_i, \]

which is the same as

\[ \| c_i+1 + c_i - d_i \|_{2} \leq c_i+1 + c_i + d_i. \] (2.37)

2.5.2 Smoothness Objective

Next a reformulation of the smoothness objective into SOCP-form is made. Shown in eq. (2.30), the smoothness term is quadratic and is required to be linear, thus a new slack- variable \( e \) is introduced and the quadratic term is formulated in SOCP- form as a constraint. The objective is,

\[ \text{minimize} \quad \omega_2 \sum_{i=0}^{n-1} (s_{i+1} - s_i)c_i \] (2.38a)

with the additional constraint,
2.5. SECOND ORDER CONE PROGRAM FORMULATION

\[
\left( \frac{\alpha(s_{i+1}) - \alpha(s_i)}{s_{i+1} - s_i} \right)^2 \leq e_i. \tag{2.39}
\]

And can be reformulated as a SOCP as following,

\[
\alpha^T Q \alpha \leq e_i
\]
\[
\alpha^T Q \alpha - e_i \leq 0
\]
\[
4\alpha^T Q \alpha - 4e_i \leq 0
\]
\[
4\alpha^T Q \alpha + (1 - e_i)^2 - (1 + e_i)^2 \leq 0
\]
\[
4\alpha^T Q \alpha + (1 - e_i)^2 \leq (1 + e_i)^2
\]
\[
\sqrt{4\alpha^T Q \alpha + (1 - e_i)^2} \leq 1 + e_i
\]

Where the square root of \( Q \) is found according to Cholesky factorization. Given that \( Q \) is a positive definite Hessian with all eigenvalues \( \geq 0 \), a factorization in the form of \( Q = L^T L \) can be obtained, where \( L \) is a triangular non-singular and unique matrix. J. Higham (2009). \( Q \) is the positive definite Hessian and is derived as below,

\[
\frac{\alpha_2 - \alpha_1}{s_2 - s_1}^2 + \frac{\alpha_3 - \alpha_2}{s_3 - s_2}^2 + \cdots + \frac{\alpha_{n-1} - \alpha_n-2}{s_{n-1} - s_{n-2}}^2
\]

Where \( \Delta s = (s_{i+1} - s_i) \) for \( i = 1 \ldots n - 1 \) and finally yield \( Q \) as,

\[
Q = \begin{bmatrix}
2/\Delta s^2 & -2/\Delta s^2 & 0 & \ldots & 0 \\
-2/\Delta s^2 & 4/\Delta s^2 & \ddots & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & 4/\Delta s^2 & -2/\Delta s^2 \\
0 & \ldots & 0 & -2/\Delta s^2 & 2/\Delta s^2
\end{bmatrix}
\]

By the factorization the obtained SOCP form is,

\[
\sqrt{4\alpha L^T L \alpha + (1 - e_i)^2} \leq 1 + e_i
\]
\[
(1 + e_i)/2 \geq \left\| \frac{L^T \alpha}{(1 - e_i)/2} \right\|.
\tag{2.40}
\]

The optimization variables are now extended further with the three new slack variables \( c, d, e \). Remaining terms in the objective function and constraints are linear and are kept unchanged. The whole SOCP formulation becomes,
\begin{align*}
\text{minimize} \quad & \alpha, \beta, u, \sigma_{\text{long}}, \sigma_{\text{lat}}, c, d, e \\
& 2\omega_1 \sum_{i=0}^{N-1} (s_{i+1} - s_i)d_i + \omega_2 \sum_{i=0}^{N-1} (s_{i+1} - s_i)e_i + \lambda_1 \|\sigma_{\text{long},i}\| + \lambda_2 \|\sigma_{\text{lat},i}\|
\end{align*}

subject to
\begin{align*}
& \left\| \frac{2}{c_i + 1 + c_i - d_i} \right\| \leq c_i + 1 + c_i + d_i \quad i = 1 \ldots n - 1, \\
& \left\| \frac{2c_i}{1 - \beta_i} \right\| \leq 1 + \beta_i \quad i = 1 \ldots n, \\
& \left\| \frac{L^T \alpha}{(1 - e_i)/2} \right\| \leq (1 + e_i)/2 \quad i = 1 \ldots n - 1, \\
& \frac{\beta_{i+1} - \beta_i}{s_{i+1} - s_i} - 2\alpha_i = 0 \quad i = 1, \ldots, n, \\
& \beta_i \leq v_i^{\text{limit},i} \quad i = 1, \ldots, n, \\
& \beta_0 = v_0^{\text{initial}} \quad i = 1, \\
& \alpha_0 = \alpha_i^{\text{initial}} \quad i = 1, \\
& \beta_n = v_n^{\text{final}} \quad i = n, \\
\end{align*}

\begin{align*}
m \left( w'' \beta_i + w' \alpha_i \right) = Ru_i - Rmg \sin \phi_i - R\frac{1}{2}Ac_i \beta_i & \quad i = 3, \ldots, n - 1, \\
\|u_i\| \leq \mu mg & \quad i = 1, \ldots, n - 1, \\
\|\alpha_i\| \leq a_c + \sigma_{\text{long},i} & \quad i = 1, \ldots, n - 1, \\
\|u_{\text{lat},i}\| \leq a_c + \sigma_{\text{lat},i} & \quad i = 1, \ldots, n - 1, \\
0 \leq \sigma_{\text{long},i} & \quad i = 1, \ldots, n - 1, \\
0 \leq \sigma_{\text{lat},i} & \quad i = 1, \ldots, n - 1, \\
T(s_j) = 2\omega_1 \sum_{i=0}^{j} (s_{i+1} - s_i)d_i & \quad T(s_j) \in (0, Tu]
\end{align*}
Chapter 3

Implementation and Results

To solve the optimization problem the SOCP formulation presented in the last chapter has been implemented using CVX, which is a modelling system for convex optimization in MATLAB. This chapter will aim to show the results of the implementation and explain the functionality of the speed planner.

3.1 Parameters and Test Platform

A table of the fixed parameters used throughout the result execution is shown in table 3.1. The vehicle mass $m$, vehicle front area $A$ and drag coefficient $c_d$ are withdrawn from typical data of HDVs. The $\mu$ value is typically chosen as the friction coefficient between rubber tyres and dry asphalt.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>Vehicle mass</td>
<td>26 000</td>
<td>kg</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravitational acceleration</td>
<td>9.81</td>
<td>m/s²</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Friction coefficient</td>
<td>0.7</td>
<td>1</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Air density</td>
<td>1.22</td>
<td>kg/m³</td>
</tr>
<tr>
<td>$A$</td>
<td>Vehicle front area</td>
<td>12</td>
<td>m²</td>
</tr>
<tr>
<td>$c_d$</td>
<td>Drag coefficient</td>
<td>0.6</td>
<td>1</td>
</tr>
</tbody>
</table>

An arbitrary test path together with its corresponding curvature and speed limitation profile has been generated. It is used to show the functionality of the objective function together with the comfort box, time window constraints and the replanning algorithm. The path is shown in fig. 3.1a. Figure 3.1b is an illustration of the predefined speed limitation $v_{\text{limit}}$ related to the Speed limitation constraint.
3.2 Time and Smoothness Objective

According to [ISO 1997] the comfort level a passenger experience for different acceleration magnitudes are given in table 3.2. A fairly reasonable value of the longitudinal and lateral acceleration and deceleration limit to choose and keep fixed throughout the results is thus 1 m/s².

<table>
<thead>
<tr>
<th>Value</th>
<th>Comfortability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 0.315 m/s²</td>
<td>Not uncomfortable</td>
</tr>
<tr>
<td>0.315 to 0.63 m/s²</td>
<td>A little uncomfortable</td>
</tr>
<tr>
<td>0.5 to 1 m/s²</td>
<td>Fairly uncomfortable</td>
</tr>
<tr>
<td>0.8 to 1.6 m/s²</td>
<td>Uncomfortable</td>
</tr>
<tr>
<td>1.25 to 2.5 m/s²</td>
<td>Very uncomfortable</td>
</tr>
<tr>
<td>Greater than 2 m/s²</td>
<td>Extremely uncomfortable</td>
</tr>
</tbody>
</table>

A test of how the speed planner adapts to the speed limit with having the time objective active and setting the smoothness weight $\omega_2$ to zero is shown in Figure 3.2.
3.2. TIME AND SMOOTHNESS OBJECTIVE

It can be seen that the acceleration profile is discontinuous and suggests maximal acceleration and deceleration in the straight segments of the speed limit. The discontinuity is also shown in the Jerk profile given as the large negative peaks, which mainly occurs as a result of the piecewise linear curvature profile of the generated test-path. This behaviour is however not desired from a comfort point of view and can be solved by adding the smoothness term. Figure 3.3 illustrates the speed profiles executed on the test path with different smoothness weights $\omega_2$. 

Figure 3.2: Speed, acceleration and Jerk profiles with no smoothness objective.
Table 3.3: Travel time over testpath.

<table>
<thead>
<tr>
<th>( \omega_2 )</th>
<th>Travel time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>104s</td>
</tr>
<tr>
<td>1</td>
<td>107s</td>
</tr>
<tr>
<td>10</td>
<td>116s</td>
</tr>
<tr>
<td>100</td>
<td>142s</td>
</tr>
</tbody>
</table>

As shown from Figure 3.3, higher values on \( \omega_2 \) results in a smoother speed profile, continuous acceleration profile and less peaks on the Jerk profile while still retaining time efficiency.

A tabular of the time it takes to travel through the path is shown in table 3.3. It is shown that the time differences between having no smoothness, have small deviations in comparison with adding a smoothness with the weights \( \omega_2 = 1 \) and \( \omega_2 = 10 \).

### 3.3 Semi-hard Comfort Limit

The semi-hard comfort limit is demonstrated by a stopping scenario on a curvy part of the test path, \( s = (400, 430) \). Table 3.4 presents the parameter values used in this scenario.
### Table 3.4: Parameters for stopping scenario

<table>
<thead>
<tr>
<th>$\omega_1$</th>
<th>$\omega_2$</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$a_{c,\text{long}}$</th>
<th>$a_{c,\text{lat}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) Speed profile for stopping scenario

(b) Longitudinal acceleration for stopping scenario

(c) Lateral acceleration for stopping scenario

Figure 3.4: A stopping scenario where the comfort box constraint is violated for different starting velocities.

The scenario in Figure 3.4 presents the longitudinal and lateral acceleration profiles for three different starting velocities $v_0 \geq 0$ and their corresponding speed profiles shown in Figure 3.4a. In Figure 3.4b, starting with $v_0 = 25$ km/h, the vehicle manages to achieve a safe stop at the end of the segment while still retaining the comfort box limitations active. However, when starting with higher velocities $v_0 = 50$ km/h and $v_0 = 70$ km/h, the comfort box limitations are violated where safety is prioritized and the deceleration is increased to the friction circle limits of roll-over and side-slip avoidance. Figure 3.4c shows the lateral acceleration profiles and similarly to the longitudinal accelerations, starting with $v_0 = 25$ km/h, a safe stop where the deceleration stays within the limits of the comfort box is obtained but is violated for the remaining higher starting velocities.
3.4 Time Window

To show the functionality of the time window constraint, a highway merging scenario is set up. The scenario is illustrated in Figure 3.5 and assumes that the autonomous HDV is driving in a highway entrance and will enter the highway after 500 m, at point A. The speed limit is increased after 200 m from 50 km/h to 90 km/h, so the HDV has 300 m distance to accelerate. On the highway there is another vehicle detected and predicted to reach point A in 32.5 s.

Figure 3.5: Highway merging scenario.

If no action is taken and the speed plan is generated without considering the vehicle on the highway, a collision is expected. The speed plan and acceleration with no time window constraint is represented by the blue line in fig. 3.6.

Figure 3.6: Speed profile of the highway merging scenario with and without time window constraint active.

Assuming that a behaviour prediction of the vehicle on the highway has been made with the vehicle having a constant velocity of 90 km/h then from fig. 3.7 it is shown that the vehicle will occupy the path between the points \( s = (500, 1000) \) m and the time \( t = (32.5, 52.5) \) s, this is shown as the thick black line. It can be seen that without any time window constraint a collision would occur at the merging point A.
To avoid the collision the time window constraint is set to enforce the HDV to reach point B before the time \( t = 32.5 \) s. If the HDV is at point B at the same time as the vehicle on the highway is at point A, it is considered safe to enter the highway. Mathematically the constraint is \( T(570) \leq 32.5 \) and the result is seen by the red lines in figs. [3.6] and [3.7]. This shows that the time window constraint adds the functionality to accelerate faster than the original plan in order to adapt to other vehicles.

### 3.5 Replanning

So far, it has only been demonstrated how the speed planner works when generating one single plan. In reality the goal is to use the planner in real time in a receding horizon fashion. In other words, to make one plan, send it to a controller that tries to track the profile and then continuously update the speed profile with the actual speed and acceleration of the vehicle as initial conditions to the planner. One issue when doing replanning is how to set the boundary conditions. It could be desired to have zero speed at the end point of each plan for safety reasons. Or it could be desired to have no specific boundary condition on the final point. Figures [3.8a] and [3.8b] show the result of replanning over the test path Figure [3.1a] in Figure [3.8a] \( \beta(s_f) = 0 \) is used and in Figure [3.8b] \( \beta(s_f) = free \). The horizon is 200 m and the plan is updated every 5 s, assuming that the controller manages to follow the plan perfectly.

The different ways of setting boundary conditions affects the solutions temporal consistency, in other other words, how similar the new plan is to the previous one. It can clearly be seen that with \( \beta(s_f) = free \) end the plans are much more temporal consistent compared with \( \beta(s_f) = 0 \). Anyhow the final executed speed profile of figs. [3.8a] and [3.8b] are quite similar, however it can be noticed that the method with \( \beta(s_f) = 0 \) is a bit more passive in the initial acceleration and the acceleration becomes jerky when switching from one plan to another.

#### 3.5.1 Vehicle Following

One desired functionality of the planner is that it should be able to adapt the speed to a vehicle in front. This is possible to do by setting upper speed limits on the waypoints where the vehicle is. One simple algorithm for adapting the speed to a vehicle in front is described in algorithm [3.1].
CHAPTER 3. IMPLEMENTATION AND RESULTS

Algorithm 3.1 Vehicle following, method 1

\begin{verbatim}
loop
    Check for vehicle in front
    if Vehicle within horizon then
        Read the vehicles position and speed: \( v_{veh_{pos}}, v_{veh_{vel}} \)
        \( \beta(v_{veh_{pos}} : end) \leq v_{veh_{vel}}^2 \)
        Run optimization to generate speed profile
    else
        \( \beta(s_f) = \text{free} \)
        Run optimization to generate speed profile
    end if
    Follow speed profile
end loop
\end{verbatim}

This algorithm sets an upper speed limit on the waypoints from where the vehicle is detected and forward, assuming that the vehicle will continue with constant speed. To demonstrate the performance of the method a scenario when approaching a slow going vehicle is set up. The initial position and speed of the ego vehicle and the vehicle in front are set as in table 3.5 and the vehicle in front is driving with constant speed. Further the scenario is set up on a straight road with a speed limit of 50 km/h.

Table 3.5: Approaching slower vehicle scenario

<table>
<thead>
<tr>
<th></th>
<th>Initial position [m]</th>
<th>Initial speed [km/h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ego vehicle</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>Vehicle in front</td>
<td>400</td>
<td>30</td>
</tr>
</tbody>
</table>

The result of how the speed is adapted to the vehicle in front is shown in Figure 3.9a and the distance to the vehicle in front as function of time can be seen in Figure 3.9b.

The planning horizon used is 200 m and the vehicle in front starts at position 400 m, hence in...
3.5. REPLANNING

The beginning of the scenario no boundary condition are set. But as soon as the vehicle is within 200 m distance, the speed profile adapts to its speed. One issue that appears when setting these boundary conditions is that the speed profiles for each replanning are not temporal consistent. Since the vehicle in front is also moving the previous boundary condition will not be the same for the next plan. This is what causes the oscillations that can be seen in the acceleration profile.

The problem with the oscillating acceleration can be partly solved by using a lower update time. In figs. 3.10a and 3.10b the same scenario is shown but this time the speed profile is updated every 1 s instead of every 5 s.

It can be seen that the acceleration profile becomes smoother thanks to the more frequent replanning.

One drawback with algorithm 3.1 is that it is very conservative and will start slowing down earlier and keep longer distance than may be considered normal driving behaviour. Therefore a
A new method, described in Algorithm 3.2, is proposed.

**Algorithm 3.2 Vehicle following, method 2**

```plaintext
loop
    Check for vehicle in front
    if Vehicle within horizon then
        \( \beta(s_f) = \text{free} \)
        Generate a speed profile as if the vehicle in front was not there
        Read the vehicle's position and speed: \( v_{eoh} \), \( v_{veh} \)
        Collision check, use the generated speed profile and assume constant speed of the vehicle in front
        Save position and speed for the collision: \( c_{pos} \), \( c_{vel} \)
        if Collision predicted within horizon then
            \( \beta(c_{pos} - \text{safety distance} : \text{end}) \leq c_{vel}^2 \)
            Generate a new speed profile
        else
            \( \beta(s_f) = \text{free} \)
        end if
    else
        \( \beta(s_f) = \text{free} \)
    end if
    Follow speed profile
end loop
```

This algorithm will first make a speed plan as if there was no vehicle in front, do a collision check assuming constant speed of the vehicle and then, only if there is a collision predicted within the planning horizon, set boundary conditions and adapt the speed.

Algorithm 3.2 is tested on the same scenario as earlier and the result is shown in figs. 3.11a and 3.11b. Compared with the solution with Algorithm 3.1, this one slows down later and gets closer to the vehicle in front.

Figure 3.11: Vehicle following scenario method 2 with horizon 200 and update time 2 s.
Chapter 4

Discussion

This chapter will analyze the functionality and performance of the implemented speed planner with a corresponding conclusion. A section of how to improve the current speed planner in the future is also presented.

4.1 Objective Function and Constraint Performances

The results demonstrate that as the speed planner takes \( x, y, z \) and \( \theta \) as inputs from the path, while having a starting velocity and smoothness objective of \( \beta_0 = 0 \) and \( \omega_2 = 0 \), respectively, the acceleration profile becomes discontinuous and results into an uncomfortable and jerky ride. However, by activating the smoothness objective with the weights \( \omega_2 = 1 \) and \( \omega_2 = 10 \), results into a fairly reasonable acceleration profile that satisfy the comfort levels of a human to experience according to ISO (1997), which also does not deviate greatly in terms of the trip time in comparison with having \( \omega_2 = 0 \), but deviates with 38 s when \( \omega_2 = 100 \). Having a high value on the weight is however not needed, since the lower values generated good performances, in terms of time and comfort.

Although having the comfort maximization and time minimization as main goals in this thesis, safety while driving is always a prioritization in the case of autonomous driving where it is of great importance to ensure that the trip goes as planned. Since the surroundings are mainly dynamical, it is thus self-evident to include the comfort box constraint in the objective function to be violated in emergency cases. This was illustrated in the scenario of Figure 3.4 which showed great performances given different starting velocities. In the cases where the starting velocities were higher than \( v_0 = 25 \) km/h, the comfortable acceleration and deceleration limitations where violated in order to stop at the given state, which indicates that the speed planner is safe in matter of urgent stops.

The time window constraint makes it possible to go faster to a point than would be the case without the constraint. Unfortunately it is not possible to put corresponding constraint in order to go slower. In reality there are a lot of situations where it is preferable to slow down in order to let someone pass instead of accelerating in front. Zhang et al. (2018) proposed a solution to this where they use the same type of time window constraint but put a very large smoothness weight. The large smoothness weight will make the whole plan a lot slower and then putting a
constraint to be at a given position before a given time is one way to drive slower to that point then what was initially planned. Although this method exists, the choice to not use it were made in this thesis, it requires manipulating the weights in a way that is not a desired interface to the planner.

4.2 Replanning and Vehicle Following

Two cases of replanning where tested, either using boundary conditions on the final point as $\beta(s_f) = 0$ or as $\beta(s_f) = \text{free}$. Results showed that having a zero end state velocity lacked temporal consistency i.e. that the speed and acceleration profiles of the next replanning segment is not consistent with the previous one. This behaviour is not desired and can cause problems in terms of comfortability of the ride. One proposition to achieve temporal consistency is however, by decreasing the update time of the replannings. Nonetheless, this approach is considered as more safe given that the velocity profiles always go to zero at each replanning, thus governs abrupt stops in all cases. In the case of $\beta(s_f) = \text{free}$, temporal consistency is achieved but with less focus on the safety. This is however not an issue since the comfort box constraint should cover for the safe stops, which already has been established to give good performances in the previous section.

The first algorithm implemented, where the functionality of following a vehicle in front is added, shows that the performance for higher update times gives oscillating solutions with lack of temporal consistency. This issue occurred earlier in the replanning part with $\beta(s_f) = 0$, it was however proposed to be resolved by decreasing the update time. This approach were tested in the algorithm which resulted in improvements in terms of temporal consistency, also in smoother speed and acceleration profiles. Another issue was that the HDV decelerates directly after the detection of the vehicle in front, at timepoint $\approx 60$ s. Therefore, to achieve a more reasonable driving behaviour by having a lower distance between the vehicles and avoid sudden deceleration as the vehicle in front is detected, the second algorithm was proposed. The new approach resulted in enabling the HDV to slow down at time $\approx 70$ s, however still achieved to follow the speed of the vehicle perfectly.

Current issues detected with the algorithm of using the speed planner in a real time system, is that it can happen that the controller can not follow the given speed profile exactly. This is not always a big problem but it might be of concern if the controller overshoots for example the speed limit of the road. This results in an infeasible start when trying to replan from the state where the vehicle's actual velocity is above one of the constraints. Another issue encountered with both algorithms related to the "following a vehicle in front functionality", is that the HDV was not able to keep a fixed distance behind the vehicle in front. This problem is mainly a consequence of the optimization formulation, since the arclength $s$ is not defined as an optimization variable and thus can not be controlled in the same manner as the speed variable.

4.3 Conclusions

A speed planner that generates speed profiles that minimizes travel time and jerk has been implemented. Travel time in order to get time efficient plans and jerk for the sake of passenger comfort and better tracking performance for the feedback controller is designed. By tuning the weights on travel time and jerk, either time efficiency or smoothness can be prioritized more. A
semi-hard comfort constraint is used in order to stay within the acceleration limits for comfort most of the time, but still be able to break the comfort constraint and brake or accelerate on the limit of friction if an emergency occurs. The temporal consistency when replanning was investigated with better consistency when having $\beta(s_f) = \text{free}$ and two algorithms for adapting the speed to a vehicle in front is proposed with better results obtained in the second algorithm.

4.4 Future Work

If more time were given to this thesis, effort would be put on mainly two things. First, extending the model to include acceleration limits dependent on which gear the vehicle is in. This would be necessary to make use of the full potential of the vehicle. Secondly, it would be interesting to find a way to use time windows on the form $W_T = [t_L, t_U]$. This for the ability to, in a good way be able to slow down in order to let dynamic obstacle pass a certain waypoint without having to completely stop before. Furthermore, to test the implementations on real trucks to evaluate the performance and functionality of the speed planner in realistic scenarios.
Bibliography


Manne Held, Oscar Flärdh, and Jonas Mårtensson. Optimal Speed Control of a Heavy-Duty Vehicle in Urban Driving, 2018. ISSN 15249050.


