TU/e Technische Universiteit Eindhoven University of Technology

Department of Mechanical Engineering
Automotive Technology
Dynamics & Control Group

Distributed Stochastic Model Predictive Control for Intersection Management Using Driver Advices

Master Thesis

Stefan Kojchev

Supervisors:
prof.dr.Henk Nijmeijer
dr.ir. Erjen Lefeber
dr.ir. Mircea Lazar
dr.ir.Alexander Katriniok (Ford-Werke GmbH)

Report number: DC 2018.046

Eindhoven, June 2018

TU/e Technische Universiteit
Eindhoven
University of Technology

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

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

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

Date

29/04/2018

Name

Stefan Kojchev

ID-number

0979770

Signature

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

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

Abstract

Intersections have a notable influence in traffic congestion and traffic accidents, especially in urban environments. A possible solution to these problems is automation of intersections and vehicles. The majority of research into this topic considers fully autonomous vehicles. The focus of this thesis, however, is suggesting appropriate speed advices to the driver in order to pass an unsignalized intersection in a safe manner and to optimize traffic flow.

As the controller expects a reaction from the driver to the speed recommendation, the approach also needs to account for any possible uncertainties that can occur in that reaction. A distributed scenario-based model predictive control (MPC) regime is proposed as a suitable method. The scenario-based MPC draws independent and identically distributed samples of the uncertainty from a bounded interval and calculates an optimal solution over the scenarios. The guarantees on avoiding collisions are directly related to the number of scenarios that is user defined. As the controller needs to account of for all the scenarios, the number of total constraints and therefore computation time significantly increases with the number of scenarios. To reduce the computational burden of the approach the Convex-Concave Procedure and Cutting Planes technique are introduced. With this method only the most restrictive constraints are involved in the optimization.

The scenario-based MPC regime is evaluated through simulations w.r.t. a control scheme that does not consider uncertainties. The results from the simulations display the capability of the approach to ensure safe passage of all interacting agents even under uncertainties from the driver reaction.

Acknowledgement

First of all, I would like to thank prof.dr. Henk Nijmeijer for organizing the project and allowing me to work on it. His guidance and advice during our meetings brought an insight that was crucial to the success of the project. I would also like to thank dr.ir. Erjen Lefeber for his valuable feedback and support that elevated the work in this thesis.

I would also like to express my sincere gratitude to my company supervisor dr.ir. Alexander Katriniok. His mentoring, guidance throughout the project and everlasting patience made this thesis possible. It was truly a pleasure to work together and learn from you. My gratitude also goes to all the colleagues at the Ford Research and Innovation Centre for making me feel welcome and like a part of the team.

I am thankful to all my friends and people that I have met during my Master's studies that have helped taking my mind off work during my spare time. It is an honour to know each and every one of you.

Especially, I would like to express my profound gratitude to my parents, brother and his wife for their continuous encouragement during my studies and the research and writing of this thesis. Your sincere love helps me through any obstacle and motivates me to improve myself on a personal and professional level. Thank you.

Aachen, March 2018 Stefan Kojchev

Contents

Nomenclature

Acronyms

CCP	Convex-Concave Procedure
EU	European Union
HMI	Human Machine Interface
LMI	Linear Matrix Inequality
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
MT	Minimum Time
OCP	Optimal Control Problem
QCQP	Quadratically Constrained Quadratic Problem
SCMPC	Scenario-based Model Predictive Control
SDP	Semi-Definite Problem
TTR	Time To React
TTS	Time To Stop
US	United States
V2V	Vehicle-to-Vehicle communication
V2X	Vehicle-to-Everything
ZOH	Zero Order Hold

Symbols

$s_x^{[i]}$	Position of agent i
$v_x^{[i]}$	Longitudinal speed of agent i
$a_x^{[i]}$	Longitudinal acceleration of agent i
$L_x^{[i]}$	Length of agent i
$W_x^{[i]}$	Width of agent i
d_{safe}	Safety distance
$d_{col}^{[i]}$	Distance to collision point for agent i
$d_{brake}^{[i]}$	Braking distance for agent i
H_p	MPC prediction horizon
H_u	MPC control horizon
p_{col}	Collision point
$p_{col,in}$	Entry point of collision region
$p_{col,out}$	Exit point of collision region

S_x	Safety coefficient
$u^{[i]}$	Control input for agent i
Δv_d	Model disturbance
K_d	Driver gain
τ	Driver delay
x	Agent state
x_τ	Delayed state
J	Cost function
$\underline{v}_{mean}$	Minimum mean velocity
K	Number of scenarios
T_{ax}	Dynamic powertrain constant
T_s	Sampling time

Chapter 1

Introduction

The number of vehicles in the European Union is rapidly increasing over the last decades, which results in an increasing amount of traffic congestion [36]. Intersections in particular are known to decrease the traffic flow, as current regulators (stop signs, roundabouts and traffic lights) require vehicles to stop and wait their turn to cross the intersection. Furthermore, intersections are prone to traffic accidents since individual vehicle trajectories cross. Statistics from the EU show that 43% of all road injury accidents are related to intersections, while 96% of those accidents are attributed to drivers [45]. Similar numbers can be seen in the US [13]. Intersections would be safer if they are controlled by an autonomous cooperative intersection algorithm that ensures no collision between vehicles happens. Furthermore, the algorithm would decide the crossing order and vehicle's speed for maximum time and energy consumption efficiency. Although the potential for improvement is highest for highly or fully automated vehicles, their series production is still in the not so near future. For that reason it is investigated how to issue recommendations to the driver in terms of vehicle speed in order to cross the intersection in a safe and optimal manner.

1.1 Problem definition and research objectives

An ideal intersection management system ensures safe passage of all vehicles, while accounting for maximum time and energy efficiency, and comfort. The main contribution of this thesis is designing a controller for intersection management that gives a driver advices in terms of recommended speed. To do so, it is needed to:

- Develop an appropriate representation of the driver reaction on speed recommendation and implement it into the controller and the simulation environment.

- Define and model the possible and relevant uncertainties that come from the driver. The reaction of the driver consists of the necessary time for the driver to react to the controller suggestion and the action of the driver. Furthermore, deviations from the suggested speed are expected as humans cannot accurately achieve and maintain an exact speed.

- Create a computationally efficient controller that takes into account these uncertainties while satisfying the safety related non-convex constraints. This is a challenging task, as the number of non-convex constraints is expected to grow.

- Perform simulation-based evaluation of the control regime.

In terms of optimal control, the control objectives are formalized as objectives to be optimized. Then, the optimization problem might consider the following objectives:

- Minimize absolute accelerations to optimize fuel consumption
- Minimize jerk to increase comfort level
- Minimize the deviation of the velocity from its set point (e.g., speed limit)
- Minimize non-smooth velocity recommendations (being the control action)

subject to the following constraints:

- Avoid collisions with other agents (non-convex constraints)
- Minimum and maximum vehicle recommended speed
- Minimum and maximum vehicle velocity
- Minimum and maximum vehicle accelerations

The control problem is to be distributed among the agents in accordance to [29], where every agent has to solve its local Optimal Control Problem (OCP). The work in this report leverages the scenario-based MPC as an appropriate method to account for the modelled uncertainties. To reduce the computational burden of the local OCP, a different technique for solving the problem is proposed. Simulation results show the effectiveness of the algorithm.

1.2 Outline

The thesis is comprised of several chapters and appendices and a brief description of each of them is given below.

Chapter 2 provides a review of the available literature related to the research topic. The chapter is structured such that first different control approaches for intersection automation are presented. Furthermore, methods for modelling the driver reaction are introduced as well as control algorithms that can deal with parameter uncertainties.

Chapter 3 covers the modelling of the intersection collision points, inter-vehicle distances and the prediction model. The coordinate systems of the vehicles and the necessary safety gap are explained in the intersection collision point section, as well as the vehicle geometry. The other section in this chapter covers the composition of the prediction model.

In Chapter 4 first the control problem is described for the distributed case where time delay is not considered. In order to solve the non-convex OCP, the SDP relaxation of the optimization problem is presented as a way to solve the problem. This control method is denominated as the nominal control algorithm as it does not directly account for any uncertainties.

Chapter 5 presents the scenario-based MPC approach as an uncertainty-aware control method. First, the definition of the control method and algorithm which extends the approach in chapter 4 are presented. Furthermore, the technique that reduces the computational time is described. Finally, the research approaches in dealing with time delay in the system and the difficulties that occur are described.

In Chapter 6 the simulation setup and results are shown. First a comparison between the nominal and the scenario control methods is presented, showing the necessity for an uncertainty-aware approach and its effectiveness in dealing with the uncertainties. The second study analyses the capability of the approach to account for time delay in the model.

Finally, Chapter 7 concludes the thesis with a summary and the general conclusions of all chapters and gives suggestions for future research topics.

Chapter 2

Literature review

One way the automation of intersections is conducted is by considering only full autonomous vehicles, while another way could be to give advices to each driver how to pass the intersection. For the case where fully automated vehicles are considered the algorithm demands acceleration from the vehicle. In the other case the driver translates the speed advice to vehicle acceleration. Due to potentially time varying driver reaction, the driver is identified as an uncertain part of the control loop. Model Predictive Control algorithms for application of intersection automation with the driver as uncertain part of the control loop have not been analysed in previous works. For that reason it is beneficial to first investigate approaches that only consider highly automated/autonomous vehicles as they have been greatly covered and provide a good basis of the problem (intersection collision point modelling, how to solve the non-convex problem, priority management etc.). In order to extend such an algorithm it is investigated how the driver reaction can be modelled as part of the control loop. Finally, algorithms that can consider parameter uncertainties are presented.

2.1 Highly automated/autonomous vehicles

The automation of intersections is a topic that has been greatly covered in recent literature. A general overview of the challenges that come from intersection management and proposed algorithms can be found in [11] and [49]. One of the common approaches is using Model Predictive Control (MPC). Some of the advantages that MPC provides are its ability to consider multiple objectives, anticipation capabilities and that the constraints are considered in the design phase of the controller. In general MPC can either be centralized, where every vehicle sends information to one control unit which solves the control problem and sends the information back or decentralized/distributed where the vehicles solve the control problem by themselves and communicate between each other. According to [17] there are three apparent advantages of a distributed approach: (1) Agents can coordinate in order to trade-off their own objectives and a global goal, while ensuring safety; (2) Robustness guarantees to a single agent failure; (3) Only the control inputs of a single agent are considered instead of having the control inputs of all agents in the Optimal Control Problem (OCP); (4) Low data rates, dropouts or proximity-based communication can be sufficiently handled.

The main task in automated intersection management is to ensure safety of all involved agents. Formulating and applying this safety constraint is what varies most in the available literature. In [8] the safety constraint is expressed as a critical set that needs to be avoided by the agent. The constraint is of non-convex nature. In order to formulate this constraint as convex, the problem is divided into two sub-problems. Problem A states that an optimal control policy needs to be found such that agent i enters the intersection after all preceding agents, whereas problem B states that agent i exits the intersection before any preceding agents. Both problems can be handled as convex problems. At each time t, agent i solves problems A and B and chooses the control with the lowest cost. Furthermore, the distributed control problem is solved sequentially where every agent takes his decision for intersection crossing according to an a priori fixed decision order. The main advantages are its low complexity and scalability.

In [34] the safety constrains are based on a minimum distance between the agents that ensures no collision, while the priority assignment is based on arrival time. The arrival time for agent i is calculated as: $\tau_i = \frac{d_{i,0}}{v_i(0)}$, where $d_{i,0}$ is the distance between vehicle i and the intersection, and $v_i(0)$ is the agent speed. The safety constraints are formulated based on this priority assignment. The approach applies soft-constrained MPC to ensure feasibility and reduce the conservatism of the collision constraint. Furthermore, the paper provides a scenario when one of the vehicles is replaced by a platoon of vehicles and modifies the cost function to ensure optimal crossing.

A different approach is presented in [38] where a transformation from the original time domain to spatial domain is performed. The safety constraint is formulated such that the time when vehicle k exits the critical set must be less than or equal to the time when the following vehicle in the sequence enters the critical set. As there is no easy way to obtain the optimal entry/exit times due to the variable speed, the domain transformation is used. In addition to the domain transformation, a variable change is performed by replacing the vehicle speed with its inverse. With this the time dependent non-convex safety constraints are translated into convex position dependent linear constraints. This centralized approach provides a minimization of the sum of all cost functions for all agents.

In [42] the authors build on their previous work [23] where a bang-bang priority preserving control law is proposed. The shortcomings of this approach is that it leads to non-smooth vehicle behaviour and increased fuel consumption due to the requirements from the control law to either maximally brake or accelerate the vehicle. This is thus the reason why the authors looked into solving the problem with MPC. The paper suggests two sequentially solved MPCs. Based on a predetermined priority order the highest prioritized agent (agent 1) solves the problem first and transmits its future states to the lower prioritized agents. After that the second prioritized agent solves the optimization problem by enforcing a priority-preserving constraint w.r.t agent 1 and transmits its future states and so on. The first MPC approach is based on a simple linear prediction scheme to estimate future states, also assuming that prior vehicles maintain constant velocities during the considered time horizon. The second MPC approach predicts the future states of prior vehicles by using the results from the previous time step. From the simulations that are conducted it is concluded that the second MPC approach performs better in terms of fuel consumption.

Apart from MPC strategies, papers like [18] use constrained Extended Kalman Filter (EKF) to estimate the states of a nonlinear 5-state vehicle model. The proposed sparse traffic optimization algorithm uses the coupled EKFs estimates to determine the optimal controls for each agent by minimizing the average delay of the agents that approach the intersection. The delay is defined as the amount of additional time required for a vehicle to traverse the intersection due to the predefined priority of other vehicles.

The work presented in [35] suggests a Cooperative Intersection Control (CIC) that relies on V2V communication. The CIC methodology consists of two control levels: (1) execution level which controls the vehicle dynamics; (2) supervisory level that manages the access to the intersection. The control of the vehicle dynamics is based on two concepts: path following for lateral control and virtual platooning for longitudinal control. The virtual platooning control expands the functionality of the Cooperative Adaptive Cruise Control (CACC) into two dimensions by using coordinate transformations. For intersecting vehicles a collision region is defined and collision is avoided if one vehicle passes first. The passing order is determined from the supervisory level control which is based on First Come First Serve crossing sequence. To assess the functionality and illustrate the benefits from CIC two simulations are performed. In the first scenario two vehicles cross the intersection, while in the second scenario a comparison between an intersection automated with CIC and an intersection controlled by traffic lights is given. It is concluded that the CIC maintains a higher average velocity and a lower average delay than traffic lights control.

In [19] all constraints and cost functions are based on time of access. In order to solve the problem with Mixed Integer Linear Programming (MILP), any discontinuity in the cost functions and constraints needs to be removed. Depending on the discontinuity either a conversion from OR logic to AND logic is performed or a slack variable is introduced. Although the approach in [4] is not about managing intersections, it provides a different collision avoidance strategy. The paper presents the agents on a collision course as circles and by using Minkowski addition transforms one agent in a geometric point. A sufficient and necessary condition for two agents to be on a collision course is:

$$\begin{gathered} \dot{l}_{ij}(t_0) < 0 \\ \dot{\chi}_1(t) \cdot \dot{\chi}_2(t) \leq 0, \end{gathered} \tag{2.1}$$

where, $\dot{l}_{ij}$ is the change of relative distance between the two agents and $\dot{\chi}_1, \dot{\chi}_2$ are the change in the limit angles. Collision avoidance can thus be ensured if the sign of one of the conditions is changed. The problem is then formulated as a constrained optimization problem and the optimal solution is obtained by a Lagrangian function.

Other approaches like [26], [25], [27] exploit hybrid systems theory and [2], [7], [15], [14] a scheduling based approach.

In the previous work, described in [29], a parallelized distributed MPC is created to automate intersections. The approach takes the vehicle as a point mass model and therefore models its dynamics as a double integrator. The drivetrain dynamics are modelled as a first order lag element.

The distributed optimal control problem for each vehicle is subjected to the system dynamics and input, state and safety constraints. The non-convex safety constraint needs to be prioritized in order to avoid deadlock-like situations from happening. For that reason the time to react (TTR) is used, which indicates how much time is left for the system until a braking manoeuvre with the maximum deceleration cannot avoid a collision at all times. Less reaction time results in higher priority. In order to solve the non-convex quadratically constrained quadratic problem (QCQP), it is reformulated into a Rank Constrained Semidefinite Problem. The problem is then transformed into a convex one by using semidefinite relaxation. This relaxed problem is solved by using known and developed techniques, such as the randomization technique described in [16]. To evaluate the approach a simulation intersection scenario involving four agents is created. The conclusion from the simulation is that the approach is real-time capable. The overall advantages of the suggested approach are: (1) multiple vehicles can enter the intersection simultaneously; (2) the distributed optimal control problem is solved in parallel without any nested iterations; (3) constraint prioritization is a generic approach without prescribing any passing order; (4) the number of required non-convex safety constraints is reduced to a minimum.

2.2 Driver as part of the control loop

The above mentioned approaches, however consider only fully autonomous vehicles. Including the driver as a part of the system raises additional challenges. One of those challenges is how to appropriately model the behaviour of the driver. This is a topic that many different authors have investigated over the years. In [1] the author gives a detailed historic review of driver acceleration models. The work itself focuses on car-following scenarios and the suggested acceleration model is either in a car-following regime (i.e., the speed is adjusted to the vehicle in front) or free-flow regime (i.e., no vehicle in front). The regime is decided based on a time headway threshold value. A driver model that is based on driving data analysis is presented in [48]. This model represents turning vehicles at signalized intersections. Based on observations from the data a speed profile model is created as: $v = c_1t^3 + c_2t^2 + c_3t + c_4$, where c_1, c_2, c_3, c_4 are coefficients that are adjusted towards the data. The speed profile model is however not influenced by signals, pedestrians or other vehicles. In [39] a detailed control modelling of humans' sensory dynamics is presented. The paper takes into account the visual, vestibular and somatosensory systems and gives transfer functions of each system. The gain coefficients of the transfer functions are also provided which are based on previous work conducted by the group. The dissertation [40] suggests a speed control method based on anticipated acceleration reference. The driver is modelled as a proportional gain combined with time delay. This is the simplest way of modelling the driver. A similar approach is seen in [41]. In [9] two driver modelling approaches are presented. The first approach is a black box model which function has the following form:

$$u_k^d = f^d(x_k^d, x_k^e, d_k^u). \tag{2.2}$$

This function directly maps the current features of the vehicle and environment $[x_k^d, x_k^e]$ to the driver inputs u_k^d with uncertainty d_k^u. The second approach is using stochastic hybrid system models where the state of the driver is represented by a random variable with an associated transition probability function. The driver commands are dependent on the driver state, and this dependency is modelled by a stochastic process. Black box methods are used for applications that require a driver model for a specific manoeuvre, while stochastic hybrid systems are used to represent transition between manoeuvres in a systematic manner.

2.3 Uncertainty-aware Model Predictive Control

Another challenge is how to include driver uncertainties in a computationally efficient control method. In [20] and [21] the same author presents two similar approaches of implementing the driver's uncertainties for a lane keeping and obstacle avoidance scenario. The controller in both papers is designed to only apply the correcting control action that is necessary to avoid violation of the safety constraints. The uncertainty that comes from the driver is presented as a deviation in the steering angle. One approach takes the uncertainties as the maximum deviation between the estimate of the steering angle and the actual steering angle, while in the other approach the uncertainty lies in a normally distributed interval. The approaches then create a feedback equation for the driver model that includes the uncertainty and solves the problem by using a Robust Model Predictive Control or a Stochastic Model Predictive Control.

In [9] two control approaches that handle uncertainties are summarized, with one of them being similar to the approach in [21]. The other approach that the paper suggests is a scenario-based MPC (SCMPC) which includes the uncertainty in the system via samples; also described in [44]. Any model-based or data-based approach can be used for generating a sufficient number of samples for the uncertainties. Moreover, the uncertainties can follow an arbitrary distribution which does not need to be known explicitly. A scenario is defined as an independent and identically distributed sample of the uncertainty variables. Each scenario can lead to a different outcome of the vehicle dynamics, driver actions and/or the traffic environment. Therefore, the corresponding predictions of the vehicle states, driver states and the environment states become deterministic, but dependent on the control inputs and the scenario. SCMPC is highly intuitive, flexible in handling many types of uncertainties and generally easy to implement. For a higher number of scenarios, the likelihood of a constraint violation decreases. Scenario MPC has been applied in centralized control schemes for lane change assistance [43] and powertrain control of hybrid electric vehicles [28].

2.4 Conclusions

By including the driver, the control problem translates into giving direct recommendations in terms of suggested speed instead of demanding an acceleration from the vehicle. The driver itself then translates the suggestion to vehicle accelerations. The challenge of such an approach comes from the fact that the reaction of the driver is time varying and can be recognized as an uncertain part of the control loop. From the conducted literature review, a MPC based approach for intersection automation where the diver reaction is involved and is uncertain is not investigated. The work in this thesis presents an approach that accounts for the modelled driver uncertainties and satisfies the control objectives.

Chapter 3

Modelling

This chapter covers the modelling of the intersection collision points, inter-vehicle distances and the model that is used for control purposes. The report considers only non-signalized four-way intersections with vehicles that are moving straight forward, as shown in Figure 3.1. Traffic signs and traffic rules such as vehicle on the left has to yield to the vehicle on the right do not hold any more, as it is left to the algorithm to decide the passing order for optimal and safe transition. Further assumptions that are made are:

- All vehicles are driven by a human driver
- The desired route and velocity of every agent passing the intersection are a priori known and do not change during the manoeuvre
- All vehicles are equipped with Vehicle-to-Vehicle (V2V) communication
- Data that has been send out at time k is available to every other vehicle at time $k + 1$
- System states are measurable and not subject to uncertainty

The vehicle's geometry is modelled as a rectangle with its width and length denoted as $W^{[i]}$ and $L^{[i]}$. The geometrical centre defines the current vehicle position and is used as a reference point in calculating the distances between agents.

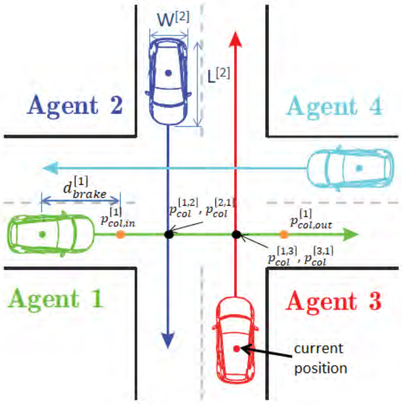

Figure 3.1: Four-way intersection as considered in this thesis

3.1 Intersection collision points

In order to ensure a safe transition for all participants, it is necessary that every agent knows the collision points with other agents in the intersection. The collision point between two agents i and j is defined as the intersection point between their respective trajectories and is denoted as $p_{\text{col}}^{[i,j]}$. The areas that are dangerous lie before and after the collision point.

The intersection infrastructure, origin and destination of each vehicle are defined in a global coordinate system with its origin being in the centre of the intersection. The control problem, however, is solved within the vehicle's own coordinate system and thus the collision points have to be transformed from the global to the vehicle's coordinate system. With this transformation a collision point between two vehicles can have different values for each vehicle. First the vehicle's origin is specified w.r.t the global coordinate system and with that the starting position of each agent is known. Knowing the origins and routes of all vehicles, the exact position of all collision points are calculated in the global coordinate system. Depending on the route of a vehicle, the distance between the geometrical centre of an agent and its collision point is computed. The collision point with the smallest distance to the considered agent is chosen as the origin of the vehicle's own coordinate system. Thus, the initial position of an agent is transformed into a negative value. The positive direction of this coordinate system is along the vehicle's driving path.

3.1.1 Safety distance gap

To ensure a safe passage between two vehicles that have a common collision point, a minimum safety distance gap (d_{safe}) is defined. By knowing the distances to the collision point for each vehicle, it can be defined that two vehicles are not colliding as long as the following equation holds.

$$\left|d_{\text{col}}^{[i]}\right| + \left|d_{\text{col}}^{[j]}\right| \geq d_{\text{safe}}. \tag{3.1}$$

In order to ensure safety, the safety distance gap is additionally multiplied by a safety coefficient S_x. Different kinds of interaction between two vehicles lead to different safety distance gaps. As the scope of this thesis is only to consider straight passing vehicles the calculation of d_{safe} is performed only for this scenario. The largest distance between two vehicles for which a collision occurs is shown in Figure 3.1.1.

From the figure the safety distance gap is defined as:

$$d_{\text{safe}} = S_x \cdot (d_{\text{col},I}^{[i]} + d_{\text{col},I}^{[j]}), \tag{3.2}$$

with $d_{\text{col},I}^{[i]}$ and $d_{\text{col},I}^{[j]}$ being equal to:

$$\begin{aligned} d_{\text{col},I}^{[i]} &= \frac{L^{[i]}}{2} + \frac{W^{[j]}}{2} \\ d_{\text{col},I}^{[j]} &= \frac{L^{[j]}}{2} + \frac{W^{[i]}}{2}. \end{aligned} \tag{3.3}$$

The calculated safety distance gap is used in the safety constraints which are further elaborated in section 4.1.

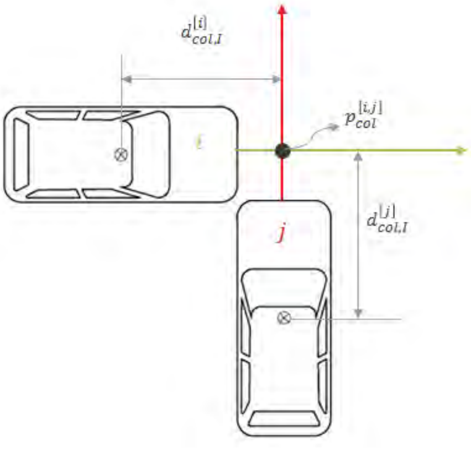

Figure 3.2: Largest distance leading to collision for two straight passing vehicles

3.2 Prediction model

By considering the driver as a part of the control loop, the prediction model consists of two parts: the driver reaction to a speed recommendation and the vehicle dynamics. Generally, the control systems issues a speed recommendation $v_{x,\mathrm{ref}}$ to the driver who transfers that recommendation into a reference acceleration. That reference acceleration is the input to the vehicle dynamics model. Figure 3.3 presents a schematic overview of the proposed model. As can be seen, the driver reaction to the recommended speed consists of a gain (K_d) and time delay parameter (τ). This approach is extended by considering the fact that drivers cannot exactly follow a speed advice, meaning that the recommended speed is subjected to additive disturbance Δv_d. From a Human-Machine-Interface (HMI) point of view, by introducing this additive disturbance it allows for showing an allowed speed interval instead of an exact speed which is more convenient for a human driver. In this chapter first the vehicle dynamics are specified after which the model dynamics including the driver reaction with and without time delay are described.

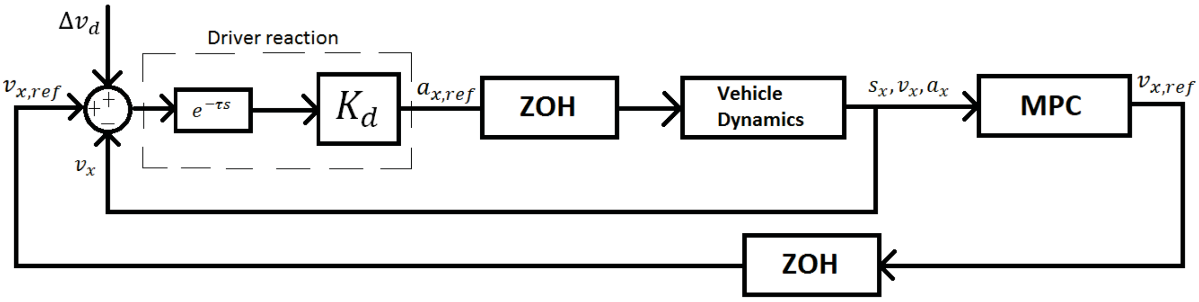

Figure 3.3: Prediction model scheme

3.2.1 Vehicle dynamics

The vehicle is taken as point mass and thus its longitudinal dynamics are simplified as a double integrator, while the drivetrain dynamics are presented by a first order lag element.

This behavior is mathematically expressed as:

$$\begin{aligned}
\dot{a}_x^{[i]} &= -\frac{1}{T_{ax}} \cdot a_x^{[i]} + \frac{1}{T_{ax}} \cdot a_{x,\text{ref}}^{[i]} \\
\dot{v}_x^{[i]} &= a_x^{[i]} \\
\dot{s}_x^{[i]} &= v_x^{[i]},
\end{aligned} \tag{3.4}$$

where T_{ax} is the powertrain time constant, a_x is the acceleration, v_x is the speed, s_x is the path coordinate and $a_{x,\text{ref}}$ is the reference acceleration. Although the simplest representation of vehicle and drivetrain dynamics, it is sufficient for the intersection scenario where no deviation from the a priori known trajectory is assumed. Due to the double integrator the controlled system is marginally stable. In the closed-loop MPC setting, however, the model can directly be used without any pre-stabilizing feedback gain [32].

3.2.2 Driver reaction to the recommended speed without time delay

When no time delay is present ($\tau = 0$) the reference acceleration is written as: $a_{x,\text{ref}}^{[i]}(t) = K_d^{[i]} e(t)$ and substituted in (3.4), where $K_d^{[i]} \in [\underline{K}_d, \overline{K}_d]$, $e(t) = \left(v_{x,\text{ref}}^{[i]}(t) + \Delta v_d^{[i]} - v_x^{[i]}(t)\right)$ and $\Delta v_d^{[i]} \in [\underline{\Delta v}_d, \overline{\Delta v_d}]$. A continuous time state-space representation of the model is of the following form:

$$\begin{cases} \dot{x}^{[i]} = A^{[i]} x^{[i]} + B^{[i]} u^{[i]} + E^{[i]} w^{[i]} \\ y^{[i]} = C^{[i]} x^{[i]} + D^{[i]} u^{[i]} \end{cases} \tag{3.5}$$

or

$$\begin{cases} \begin{bmatrix} \dot{a}_x^{[i]} \\ \dot{v}_x^{[i]} \\ \dot{s}_x^{[i]} \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_{ax}} & -\frac{K_d^{[i]}}{T_{ax}} & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_x^{[i]} \\ v_x^{[i]} \\ s_x^{[i]} \end{bmatrix} + \begin{bmatrix} \frac{K_d^{[i]}}{T_{ax}} \\ 0 \\ 0 \end{bmatrix} u^{[i]} + \begin{bmatrix} \frac{K_d^{[i]}}{T_{ax}} \\ 0 \\ 0 \end{bmatrix} w^{[i]} \\ y^{[i]} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_x^{[i]} \\ v_x^{[i]} \\ s_x^{[i]} \end{bmatrix}, \end{cases} \tag{3.6}$$

with $D^{[i]} = 0$, $u^{[i]}$ being equal to the recommended speed $v_{x,\text{ref}}^{[i]}$ and $w^{[i]}$ denotes the exogenous disturbance $\Delta v_d^{[i]}$ and refers to the allowed velocity offset to the speed advice. This system is discretized with the zero-order hold discretization technique. In essence:

$$x_{k+1}^{[i]} = e^{A^{[i]}(T_s)} x_k^{[i]} + \int_{k\cdot T_s}^{(k+1)\cdot T_s} e^{A^{[i]}(T_s\cdot(k+1)-s)} ds B u_k^{[i]} + \int_{k\cdot T_s}^{(k+1)\cdot T_s} e^{A^{[i]}(T_s\cdot(k+1)-s)} ds E w_k^{[i]}. \tag{3.7}$$

The discrete $A^{[i]}$, $B^{[i]}$ and $E^{[i]}$ matrix are: $A_d^{[i]} = e^{A^{[i]}T_s}$; $B_d^{[i]} = \int_{k\cdot T_s}^{(k+1)\cdot T_s} e^{A^{[i]}(T_s\cdot(k+1)-s)} ds B^{[i]}$; $E_d^{[i]} = \int_{k\cdot T_s}^{(k+1)\cdot T_s} e^{A^{[i]}(T_s\cdot(k+1)-s)} ds E^{[i]}$, with T_s being the sampling time and $C_d^{[i]} = C^{[i]}$. Finally, the discrete-time state space model is:

$$\begin{cases} \begin{bmatrix} a_{x,k+1}^{[i]} \\ v_{x,k+1}^{[i]} \\ s_{x,k+1}^{[i]} \end{bmatrix} = A_d^{[i]} \begin{bmatrix} a_{x,k}^{[i]} \\ v_{x,k}^{[i]} \\ s_{x,k}^{[i]} \end{bmatrix} + B_d^{[i]} u_k^{[i]} + E_d^{[i]} w_k^{[i]} \\ y_k^{[i]} = C_d^{[i]} \begin{bmatrix} a_{x,k}^{[i]} \\ v_{x,k}^{[i]} \\ s_{x,k}^{[i]} \end{bmatrix} \end{cases} \tag{3.8}$$

3.2.3 Driver reaction to the recommended speed with time delay

As the system has a feedback loop around the time delay, an arbitrary time delay (τ is irrational multiple of the sample time) would result in an infinite dimension of the system dynamics [5]. To cope with this issue the time delay is taken as an integer multiple of the sampling time that in the system is represented as a shift in the input. The reference acceleration in this case is:

$$a_{x,ref}(t) = K_d(v_{x,ref}(t-\tau) + \Delta v_d(t-\tau) - v_x(t-\tau)). \tag{3.9}$$

The continuous time state space can be written as:

$$\begin{cases} \dot{x}(t) = A_1 x(t) + A_2 x(t-\tau) + Bu(t-\tau) + Ew(t-\tau) \\ y(t) = Cx(t), \end{cases} \tag{3.10}$$

where $A_1 = \begin{bmatrix} -\frac{1}{T_{ax}} & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, $A_2 = \begin{bmatrix} 0 & -\frac{K_d}{T_{ax}} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} \frac{K_d}{T_{ax}} \\ 0 \\ 0 \end{bmatrix}$ and$E = B$. The A_2 matrix can be written as: $A_2 = A_2{}'\xi^T$ with $A_2{}' = \begin{bmatrix} -\frac{K_d}{T_{ax}} \\ 0 \\ 0 \end{bmatrix}$ and$\xi = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$. It can be noticed that $A_2{}' = -B$. Using this representation and by defining $e(t-\tau) = u(t-\tau) + w(t-\tau) - \xi^T x(t-\tau)$ to represent the delayed velocity tracking error, the $\dot{x}(t)$ equation is written as:

$$\dot{x}(t) = A_1 x(t) + Be(t-\tau), \tag{3.11}$$

The zero order hold is applied to both vehicle dynamics and driver reaction model. In the following $A_{1,d}$ and $B_{1,d}$ denote the discretized version of A_1 and B computed with the zero-order hold technique. The discrete time state space is of the following form:

$$\begin{cases} \bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{e}_k \\ \bar{y}_k = \bar{C}\bar{x}_k, \end{cases} \tag{3.12}$$

or

$$\begin{cases} \bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k + \bar{E}\bar{w}_k \\ \bar{y}_k = \bar{C}\bar{x}_k, \end{cases} \tag{3.13}$$

with $\bar{A} = \begin{bmatrix} \Gamma_1 & \Gamma_2 \\ \Gamma_3 & \Gamma_4 \end{bmatrix}$, $\bar{B} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and has (time delay+3,1) size, $\bar{E} = \bar{B}$, and $\bar{C} = \begin{bmatrix} \Gamma_5 & 0 & 1 & 0 \end{bmatrix}$.

$\Gamma_1 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}$ and is of (time delay, time delay) size;

$\Gamma_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & -1 & 0 \end{bmatrix}$ and is of (time delay,3) size;

$\Gamma_3 = \begin{bmatrix} B_{1,d} & 0_{(3,\text{time delay}-1)} \end{bmatrix}$ and has (3,time delay) size;
$\Gamma_4 = A_{1,d}$ and is of (3,3) size;
and Γ_5 is of (1,time delay) size and of $\Gamma_5 = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}$ form.

Example of the system dynamics for time delay of 3 time steps is:

$$\begin{cases} \begin{bmatrix} x_{\tau 3,k+1} \\ x_{\tau 2,k+1} \\ x_{\tau 1,k+1} \\ x_{1,k+1} \\ x_{2,k+1} \\ x_{3,k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ B_{1,d}(1,1) & 0 & 0 & A_{1,d}(1,1) & A_{1,d}(1,2) & A_{1,d}(1,3) \\ B_{1,d}(2,1) & 0 & 0 & A_{1,d}(2,1) & A_{1,d}(2,2) & A_{1,d}(2,3) \\ B_{1,d}(3,1) & 0 & 0 & A_{1,d}(3,1) & A_{1,d}(3,2) & A_{1,d}(3,3) \end{bmatrix} \begin{bmatrix} x_{\tau 3,k} \\ x_{\tau 2,k} \\ x_{\tau 1,k} \\ x_{1,k} \\ x_{2,k} \\ x_{3,k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u_k + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} w_k \\ y_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{\tau 3,k} \\ x_{\tau 2,k} \\ x_{\tau 1,k} \\ x_{1,k} \\ x_{2,k} \\ x_{3,k} \end{bmatrix} . \end{cases} \tag{3.14}$$

In the later phases of this project a system representation that can consider arbitrary time delay is found. This representation exploits the structure of the system dynamics and its full form is shown in Appendix A. In this report the current representation (where time delay is a integer multiple of the time sample) is kept and used and the formulation mentioned in the Appendix A can be considered as a future extension of the model.

With the presence of time delay in the system it is expected that unstable poles can be obtained. This issue raises the need for a pre-stabilizing gain in the MPC formulation.

3.3 Conclusions

This chapter presents the definition of the intersection collision points and inner-vehicle distances that are necessary to define the safety distance. Furthermore, it covers the model used for control purposes, which is consisted of vehicle dynamics and driver reaction model, in the cases when time delay is and is not present.

Chapter 4

Distributed Model Predictive Control

This chapter presents the first control approach that was taken which does not directly account for the possible uncertainties in the driver reaction. The approach is denoted as the nominal MPC scheme. The idea of MPC is to obtain optimal control actions w.r.t. the applied system model by solving an optimal control problem subject to constraints. In a MPC environment usually a linear or linearised discrete-time process model is used. When using MPC, at every time step k a finite-time optimal control problem is solved over a finite prediction horizon of length H_p. The controls are applied over a control horizon H_u of length $H_u \leq H_p$. After optimization only the first control input from the sequence is applied and then at the next time step the optimization is executed over a shifted horizon.

The optimal control problem is formed and solved in a distributed way by applying a primal decomposition technique to the centralized OCP in the same manner as in [29]. With this technique the centralized OCP is translated into local OCPs with coupled constraints. The collision avoidance constraint is a global objective that couples the agents. In the following, the notation $\{\cdot\}_{(\cdot|k)}$ refers to the full prediction horizon data of the variable $\{\cdot\}$.

On a general level, the full control approach can be summarized with the following algorithm:

Algorithm 1 Distributed MPC, agent i at time k

```
1. **Receive data via V2X**. Receive the trajectories ($d^{[j]}_{i,(\cdot|k)}$) of all agents.
2. **Optimize**. Formulate and solve the OCP of agent $i$ and obtain optimal control sequence $u^{[i],*}_{(\cdot|k)}$.
3. **Broadcast predicted trajectory via V2X**. Compute the distances of agent $i$ to collision points with other agents $j$ ($d^{[i,j]}_{\text{col},(\cdot|k)}$) and broadcast information. This distance is defined as: $d^{[i,j]}_{\text{col},(\cdot|k)} = \left| s^{[i]}_{x,(\cdot|k)} - p^{[i,j]}_{\text{col},(\cdot|k)} \right|$
4. **Apply control**. Apply the first element of the computed optimal control sequence $u^{*}_{(k|k)}$.
5. **Increment time**. $k = k + 1$. Go to step 1.
```

The key steps are how to form and solve the OCP. For that reason in this chapter first the optimal control problem is stated along with the constraints that are imposed in order to satisfy the control objectives. Following, the intersection prioritization is explained and the method for solving the Optimal Control Problem is described. Finally, the conclusions from this chapter are shown.

4.1 Optimal Control Problem

The calculation of the control signal is based on the minimization of a cost function, which is formed based on the main objectives. The velocity v_x of every agent i at the predicted time should be close to the speed limit v_{limit} in order to optimize traffic flow. The speed limit refers to the actual speed limit of the intersection or a virtual speed limit and is constant for straight passing vehicles. The changes of the advised speed Δu should be kept small in order to achieve convenient speed advices to the driver. Finally, an increase of fuel efficiency and to provide comfortable driving experience the longitudinal acceleration and jerk should be minimized. These objectives are summarized in the following cost function:

$$\begin{aligned} J^{[i]}(x_0^{[i]}, u_k^{[i]}) \triangleq & Q^{[i]} \sum_{m=1}^{H_p} (v^{[i]}_{\text{limit},(k+m|k)} - v^{[i]}_{x,(k+m|k)})^2 + R^{[i]} \sum_{j=0}^{H_u-1} \Delta u^{[i],2}_{(k+m|k)} + \\ & S^{[i]} \sum_{m=1}^{H_p} a^{[i],2}_{x,(k+m|k)} + T^{[i]} \sum_{m=1}^{H_p} \Delta a^{[i],2}_{x,(k+m|k)}, \end{aligned} \tag{4.1}$$

where $Q^{[i]}$, $R^{[i]}$, $S^{[i]}$ and $T^{[i]}$ are positive weighting coefficients.

In addition to the objectives in the cost function, objectives that are formed as constraints are incorporated in the problem. First, the minimum and maximum advised speed is bounded such that the upper bound corresponds to the legal speed limit and the lower bound is equal to zero. With these bounds the vehicle is not allowed to drive backwards and has to drive slower than the intersection speed limit. This objective translates in the following input constraint:

$$0 \leq u^{[i]}_{(k+m|k)} + \Delta v^{[i]}_{d,(k+m|k)} \leq \bar{u}^{[i]}_{(k+m|k)}. \tag{4.2}$$

Furthermore, the actual vehicle velocity and absolute accelerations that result from the driver reaction to the speed advice are constrained to a reasonable range to encourage safe driving. These two objectives are formulated in the following state constraints:

$$0 \leq v^{[i]}_{x,(k+m|k)} \leq \bar{v}^{[i]}_x, \tag{4.3}$$

$$\underline{a}^{[i]}_x \leq a^{[i]}_{x,(k+m|k)} \leq \bar{a}^{[i]}_x. \tag{4.4}$$

For feasibility reasons, it needs to be ensured that the prediction horizon covers the coordinate set $S_c \triangleq [p_{col,in}, p_{col,out}]$ in which potential collisions may occur between agents. If the optimization problem is not feasible, then the agent conducts an emergency braking manoeuvre in a brake safe distance d_{brake}. In essence, this is when the agent is entering the set $S_{cb} \triangleq [p_{\text{col,in}} - d_{\text{brake}}, p_{\text{col,out}}]$. To achieve this objective, the minimum mean velocity is bounded by:

$$\frac{1}{H_p+1}(v^{[i]}_{x,k}+\sum_{m=1}^{H_p} v^{[i]}_{(k+m|k)}) \geq \underline{v}^{[i]}_{\text{mean}} \tag{4.5}$$

If $s_{x,k} \in S_{cb}$, i.e., if the agent has approached the brake safe distance, and the agents with higher priority have not left the conflict region, the minimum mean velocity ($\underline{v}_{\text{mean}}$) is calculated by dividing the remaining distance to $p^{[i]}_{\text{col,out}}$ by the preview time covered by the prediction horizon. Otherwise, $\underline{v}_{\text{mean}}$ is set to zero. Further details on the prioritization of the agents is given in section 4.2.

Finally, the control regime must assure collision avoidance which is the most important control objective. Theoretically, the safety constraints are defined such that the sum of the distances between two interacting agents and their common collision point must be greater than the safety distance gap. The safety distance gap is defined in section 3.1.1. The distance between the controlled agent i and the collision point is defined as the distance between the vehicle's position $s^{[i]}_{x,(\cdot|k)}$ and the position of the common collision point $p^{[i,j]}_{\text{col}}$. The distance of the colliding agent and the collision point is denoted as $d^{[j]}_{\text{col},(\cdot|k)}$ and is broadcasted via V2X communication. Mathematically the safety constraint is written as:

$$\left| s^{[i]}_{x,(k+m|k)} - p^{[i,j]}_{\text{col}} \right| + d^{[j,i]}_{cocoll,(k+m|k)} \geq d^{[i,j]}_{\text{col}}, m \in 1, \ldots, H_p. \tag{4.6}$$

This constraint is rearranged in the form of a non-convex quadratic inequality constraint:

$$\left(s^{[i]}_{x,(k+m|k)} - p^{[i,j]}_{\text{col}} \right)^2 \geq \left(d^{[i,j]}_{\text{safe}} - d^{[j]}_{\text{col},(k+m|k)} \right)^2, d^{[j,i]}_{\text{col},(k+m|k)} \leq d^{[i,j]}_{\text{safe}}, m \in 1, \ldots, H_p. \tag{4.7}$$

The safety constraint is imposed only when the non-squared right side of the inequality is larger than zero $\left(d^{[i,j]}_{\text{safe}} - d^{[j]}_{\text{col},(k+m|k)} \right)$, as in the other cases it is satisfied anyway. No rear end collision avoidance constraints are considered in addition, as it is assumed that the driver is capable of avoiding rear end collisions with other agents at intersection crossings.

All of these objectives can be summarized as a non-convex quadratically constrained quadratic optimization problem that is solved independently by each agent:

$$\begin{aligned} &\min J^{[i]}(x^{[i]}_0, u^{[i]}_{(\cdot|k)}) \\ &\text{s.t.} \quad \text{system dynamics (3.12)} \\ &\qquad \text{input constraints (4.2)} \\ &\qquad \text{state constraints (4.3), (4.4) \& (4.5)} \\ &\qquad \text{safety constraints (4.7).} \end{aligned} \tag{4.8}$$

4.2 Agent prioritization

For non-convex problems there is no unique way on how to deal with coupling constraints. By simply imposing safety constraints pairwise it might lead to deadlock situations or collisions when solving the problem in parallel. The reason is the fact that the agents are not finally aware of what the other agents' purpose might be. In order to overcome this issue instead of solving pairwise safety constraints, the safety constraints are prioritized.

The prioritization of the agents is determined by the Time To React (TTR). TTR is a function of the Minimum Time (MT) and the Time To Stop (TTS). Minimum time is defined as the minimum time in which an agent reaches its nearest collision point without taking into consideration other agents. Time To Stop is defined as the minimum time an agent needs to stop the vehicle such that it does not pass the collision point. The TTS is computed with the prediction model taking into consideration the maximum allowed velocity and deceleration. Finally, TTR is the subtraction of TTS from MT. Figure 4.1 illustrates this relation. In essence, TTR is the time in which the agent could start a braking action in order to stop the vehicle in front of the first common collision point.

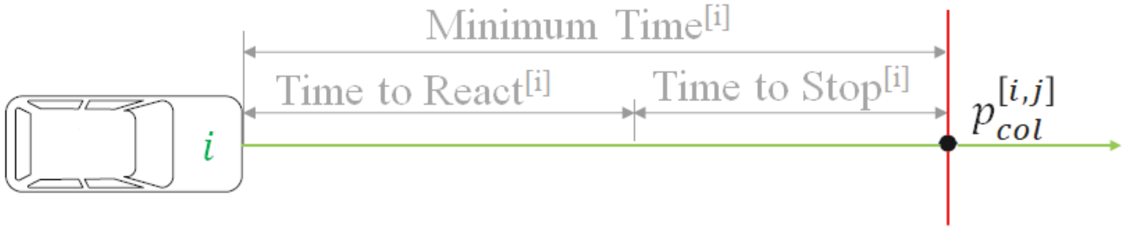

Figure 4.1: Time to react

The agent with the shortest Time To React gets the lowest value (i.e., priority equal to 1) and is the highest prioritized agent. Additionally, agents without a common collision point with any other agent, meaning vehicles on a non-collision route, do not calculate their TTR. They are not considered in the priority order and receive a special integer equal to the highest ranked vehicle. It is important to note that prioritized safety constraints do not imply any intersection crossing order, but a priority order in case of a potential conflict. An agent with lower priority has to yield for the higher prioritized agent, however an agent can transverse the intersection first if it satisfies the given safety constraint. To be independent of any centralized priority coordination regime, it is assumed that priorities are negotiated in a distributed way among the agents once when the scenario is established at time t_0 and then remain constant for the entire manoeuvre.

4.3 Solving the non-convex distributed optimal control problem

The non-convex quadratically constrained quadratic problem (QCQP) can be written as:

$$\begin{aligned}
\min_{u^{[i]}_{(\cdot|k)}} \; & u^{[i],T}_{(\cdot|k)} P_0 u^{[i]}_{(\cdot|k)} + q^{[i],T}_0 u^{[i]}_{(\cdot|k)} + r^{[i]}_0 \\
\text{s.t. } & u^{[i],T}_{(\cdot|k)} P_{j,(k+m|k)} u^{[i]}_{(\cdot|k)} + q^{[i],T}_{j,(k+m|k)} u^{[i]}_{(\cdot|k)} + r^{[i]}_{j,(k+m|k)}, \; d^{[j,i]}_{\text{col}} < d_{\text{safe}} \; \& \\
& \qquad\qquad\qquad\qquad\qquad\qquad \text{priority}(j) < \text{priority}(i) \\
& \text{input constraints (4.2)} \\
& \text{state constraints (4.4) \& (4.5)}
\end{aligned} \tag{4.9}$$

The $P^{[i]}_0$ matrix is a positive semidefinite matrix, while the $P^{[i]}_{j,(k+m|k)}$ indicate the non-convex part of the safety constrains and because of that is a negative semidefinite matrix.

Furthermore, $q_0^{[i]}, q_{j,(k+m|k)}^{[i],T}$ are vectors while $r_0^{[i]}, r_{j,(k+m|k)}^{[i]}$ are affine terms.

By defining $U^{[i]} \triangleq u_{(\cdot|k)}^{[i]} u_{(\cdot|k)}^{[i],T}$ as described in [6], the QCQP with prioritized constraints is reformulated as a rank constrained semidefinite program:

$$\begin{aligned}
&\min_{U^{[i]}, u_{(\cdot|k)}^{[i]}} \; Tr(U^{[i]} P_0^{[i]}) + q_0^{[i],T} u_{(\cdot|k)}^{[i]} + r_0^{[i]} \\
&\text{s.t. } Tr(U^{[i]} P_{j,(k+m|k)}^{[i]}) + q_{j,(k+m|k)}^{[i],T} u_{(\cdot|k)}^{[i]} + r_{j,(k+m|k)}^{[i]} \leq 0, \; d_{\text{col}}^{[j,i]} < d_{\text{safe}} \; \& \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{priority}(j) < \text{priority}(i) \\
&\qquad S^{[i]} \triangleq \begin{bmatrix} U^{[i]} & u_{(\cdot|k)}^{[i]} \\ u_{(\cdot|k)}^{[i]} & 1 \end{bmatrix} \geq 0, \text{rank}(S^{[i]}) = 1 \\
&\qquad \text{input constraints (4.2)} \\
&\qquad \text{state constraints (4.4) \& (4.5)}
\end{aligned} \tag{4.10}$$

$S^{[i]}$ is the Schur complement matrix of $U^{[i]} - u_{(\cdot|k)}^{[i]} u_{(\cdot|k)}^{[i],T}$, and with $S^{[i]} \geq 0$ and $\text{rank}(S^{[i]}) = 1$ it is stated that $U^{[i]} = u_{(\cdot|k)}^{[i]} u_{(\cdot|k)}^{[i],T}$. The $Tr(\cdot)$ stands for the trace of a square matrix.

The problem is solved by dropping the $\text{rank}(S^{[i]}) = 1$ constraint and with that a semidefinite relaxation (SDR) of the original problem is obtained, [6]. Mature optimization techniques already exist for such problems that provide efficient solutions. The technique that is adopted to obtain a feasible solution of the OCP is a variant of the randomization technique shown in [16]. The development of this approach was part of previous work performed by the group at the Ford Research and Innovation centre. The full approach and algorithm, as well as the feasibility and optimality are mentioned in [29]. The same approach is used in solving this OCP.

4.4 Conclusions

This chapter shows how to form the intersection management problem into an Optimal Control Problem embedded in a MPC-based framework. Furthermore, it showed the SDP relaxation method as a technique to solve the originally non-convex, now convex problem while ensuring feasibility.

The nominal MPC approach considers the driver parameters to be constant. It is assumed that the value of these constants are determined and adjusted based on previous behaviour of the driver. However, drivers often change behaviour due to external reasons (tiredness, mood, environmental occurrences, etc) or when a different person is driving the same vehicle (spouse or friend). Therefore, the controller model parameters might mismatch with the actual driver parameters present in the system. This mismatch between the predicted and actual behaviour can lead to violation of the imposed constraints, especially critical for safety constraints.

Chapter 5

Distributed Scenario-based Model Predictive Control

The approach described in the previous section is not able to consider any uncertainties that can come from the driver reaction model. Thus, it is expected that a miss-match between the predicted value in the control model and the actual driver can occur. According to literature [9], [44] the scenario-based MPC is an uncertainty-aware control scheme. The approach gives a probabilistic guarantee on constraint satisfaction by sampling the uncertainties in the system in multiple scenarios.

Let $\delta_{(k|k)}^{[\kappa]}$ be independent and identically distributed sample of the uncertainty at time k. A scenario $\sigma_k^{[\kappa]}$ is defined as the full horizon sample: $\sigma_k^{[\kappa]} \triangleq \left\{\delta_{(k|k)}^{[\kappa]}, \ldots, \delta_{(k+H_p-1|k)}^{[\kappa]}\right\}$. These randomly selected scenarios reflect the potential driver reaction.

5.1 Scenario Optimal Control Problem

The optimal solution of the problem using scenario-based MPC is found by optimizing over all scenarios. To implement this approach the formulation of the OCP, step 2) of the nominal control approach algorithm (algorithm 1), is modified. In essence:

Algorithm 2 Scenario OCP, agent i at time k

1. **Scenario Generation**. Sample K scenarios.
2. **Scenario Constraints**. For every scenario impose input, state and safety constraints.
3. **Scenario Optimization**. Solve a single OCP that finds an optimal solution over K scenarios s.t. scenario constraints.

It is assumed that the driver reaction does not change over a short period of time. This translates into fixing the driver gain $K_d^{[i,\kappa]} \in [\underline{K}_d^{[i]}, \overline{K}_d^{[i]}]$ and time delay $\tau^{[i,\kappa]} \in [\underline{\tau}^{[i]}, \overline{\tau}^{[i]}]$ parameters over the prediction horizon. Every scenario still covers a different realization of the parameters. On the other hand it is expected that the offset to the suggested speed $\Delta v_d^{[i,\kappa]}$ varies over a short period of time. As such it is sampled from the bounded interval $[\Delta \underline{v}_d^{[i]}, \Delta \overline{v}_d^{[i]}]$

for each time step of the prediction horizon. As a result of the varying parameters, a different system model is obtained for every scenario

$$\begin{cases} x^{[i,\kappa]}_{k+1} &= A^{[i,\kappa]}_{d} x^{[i,\kappa]}_{k} + B^{[i,\kappa]}_{d} u^{[i]}_{k} + E^{[i,\kappa]}_{d} w^{[i,\kappa]}_{k} \\ y^{[i,\kappa]}_{k} &= C^{[i]}_{d} x^{[i,\kappa]}_{k}. \end{cases} \tag{5.1}$$

For each of these scenarios separately, the state and safety constraints of the nominal OCP have to be implied. The input and the state constraints on the velocity and acceleration are expanded to:

$$0 \leq u^{[i,\kappa]}_{(k+m|k)} + \Delta v^{[i,\kappa]}_{d,(k+m|k)} \leq \overline{u}^{[i]}_{(k+m|k)} \tag{5.2}$$

$$\underline{v}^{[i]}_{x,(k+m|k)} \leq v^{[i,\kappa]}_{x,(k+m|k)} \leq \overline{v}^{[i]}_{x,(k+m|k)} \tag{5.3}$$

$$\underline{a}^{[i]}_{x,(k+m|k)} \leq a^{[i,\kappa]}_{x,(k+m|k)} \leq \overline{a}^{[i]}_{x,(k+m|k)}. \tag{5.4}$$

In the same manner, the minimum mean velocity constraint is imposed for every scenario. In essence:

$$\frac{1}{H_p + 1} \left(v^{[i]}_{x,k} + \sum_{m=1}^{H_p} v^{[i,\kappa]}_{(k+m|k)} \right) \geq v^{[i]}_{\text{mean}}. \tag{5.5}$$

Defining the safety constraints for the Scenario MPC is more challenging as they depend on the sampled trajectories of other agents j. In order to avoid every agent transmitting all of its sampled trajectories for all scenarios, causing a large amount of data that is transmitted through V2V, a different approach is taken. First the agent j calculates its path coordinate trajectories $s^{[j,\kappa]}_{x,(\cdot|k)}$ for every scenario. Due to the uncertainties, the predicted trajectories vary per scenario. Then, agent j determines its minimum and maximum path coordinate at the predicted step $k + m$, i.e.,

$$\begin{aligned} \underline{s}^{[j]}_{x,(k+m|k)} &\triangleq \min_{\kappa \in K} s^{[j,\kappa]}_{x,(k+m|k)} \\ \overline{s}^{[j]}_{x,(k+m|k)} &\triangleq \max_{\kappa \in K} s^{[j,\kappa]}_{x,(k+m|k)}. \end{aligned} \tag{5.6}$$

To account for these trajectories, the length of the agents is artificially enlarged as:

$$\Delta L^{[j]}_{(k+m|k)} \triangleq \overline{s}^{[j]}_{x,(k+m|k)} - \underline{s}^{[j]}_{x,(k+m|k)}, \tag{5.7}$$

and a new geometrical centre of the vehicle is calculated

$$\hat{p}^{[j]}_{(k+m|k)} \triangleq \frac{1}{2} \left(\underline{p}^{[j]}_{(k+m|k)} + \triangleq \overline{p}^{[j]}_{(k+m|k)} \right). \tag{5.8}$$

The artificial enlargement and the new geometrical centre are shown in Figure 5.1.

The trajectory $\hat{d}^{[j]}_{\text{col},(\cdot|k)}$ that is broadcasted via V2V communication is determined based on the artificial vehicle position trajectory $\hat{s}^{[j]}_{x,(\cdot|k)}$. Furthermore, the artificially enlarged length is broadcasted to the other agents as it is considered in the safety constraints. Having calculated these parameters, the safety constraints for agent i and scenario κ are stated as:

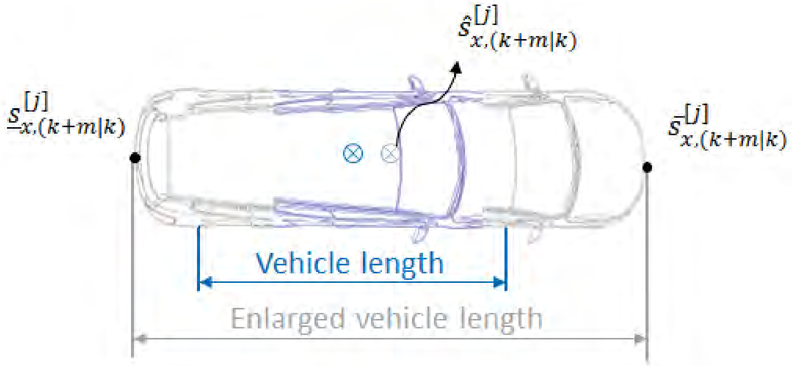
$\hat{s}^{[j]}_{x,(k+m|k)}$
$\underline{s}^{[j]}_{x,(k+m|k)}$
$\bar{s}^{[j]}_{x,(k+m|k)}$
Vehicle length
Enlarged vehicle length

Figure 5.1: Artificially enlarged vehicle and new geometrical centre

$$\left(s^{[i,\kappa]}_{x,(k+m|k)} - p^{[i,j]}_{\text{col}}\right)^2 \geq \left(\hat{d}^{[i,j]}_{\text{safe}} - \hat{d}^{[j]}_{\text{col},(k+m|k)}\right)^2, \tag{5.9}$$

with $\hat{d}^{[i,j]}_{\text{safe}} \triangleq d^{[i,j]}_{\text{safe}} + \Delta L^{[j]}_{(k+j|k)}$.

The scenario OCP is obtained by formulating the cost function (4.8) for every scenario. The distributed scenario OCP finds the optimal solution to the problem by optimizing on average over all scenarios subject to all scenario constraints. In essence:

$$\begin{aligned} \min_{u^{[i]}_{(\cdot|k)}} \ & \frac{1}{K}\sum_{\kappa=1}^{K} J^{[i,\kappa]}(x^{[i]}_0, u^{[i]}_{(\cdot|k)}) \\ \text{s.t. } & \text{system dynamics (5.1)} \\ & \text{safety constraints (5.9)} \\ & \text{input constraints (5.2)} \\ & \text{state constraints (5.3), (5.4) \& (5.5).} \end{aligned} \tag{5.10}$$

For a centralized scheme in [43], [44] it is proven that the scenario constraints are satisfied by chance. In the distributed setting, the uncertainties of every agent are independent of each other and as such, samples and scenarios are generated independently. A direct correlation between number of scenarios and the chance of constraint violation is presented in [9].

Let $S^{[i,\kappa]}_{\text{safe},(k+m|k)}$ denote a set of path coordinates at time $k+m$ that satisfy the safety constraint (5.9) for an arbitrary scenario $\tilde{\kappa}$. The probability of constraint violation is denoted as:

$$Pr\left\{ s^{[i,\tilde{\kappa}]}_{k+m|k} \notin \bigcap_{\kappa \in K} S^{[i,\kappa]}_{safe,(k+m|k)} \right\} \leq \frac{1}{1+K}. \tag{5.11}$$

By broadcasting the worst case scenarios to the other agents, in the worst case at least the same upper probability bound on constraint violation as in the centralized case is achieved. The control problem is solved with the same approach as described in section 4.3.

The scenario MPC considers a significantly larger number of constraints and solving such a problem has a high computational burden. As one of the goals of the thesis is to create a computationally efficient algorithm, it is investigated how this goal can be achieved. The following subsection outlines a different numerical optimization approach to efficiently solve the OCP.

5.2 Convex-Concave Procedure

The main computational burden comes from the non-convex safety constraints as these require the optimization problem to be formulated as a SDP. The Convex-Concave Procedure, described more in detail in [31], retains all the information from the convex part and linearises the non-convex part. In this thesis in particular, the Penalty Convex-Concave Procedure is used as this approach removes the necessity for an initial feasible point. The optimization problem is relaxed by adding slack variables and penalizing the sum of the violations. In essence, by initially putting a low penalty on violations, it is allowed for constraints to be violated so that a region with lower objective value is found. In standard Penalty CCP, a slack is used for every non-convex constraint. An addition that is made to this algorithm is to leverage the MPC setup and instead of imposing slack variables for the full prediction horizon, slacks are only introduced for the steps where safety constrains are enforced. This way some computation time is reduced. The complete algorithm of the Penalty CCP is presented below. Solutions coming from this approach are local.

Algorithm 3 Penalty Convex Concave Procedure [31]

```
given an initial point $x_0, \tau_0 > 0, \tau_{\max}$, and $\mu > 1$.
$n := 0$.
repeat
    1. Convexify. Form $\hat{g}_i(x; x_n) \triangleq g_i(x_n) + \bigtriangledown g_i(x_n)^T(x - x_n)$ for $i = 0, \dots, m$
    2. Solve. Set the value of $x_{n+1}$ to a solution of
           minimize     $f_0(x) - \hat{g}_0(x; x_n) + \tau_n \sum_{i=1}^m s_i$
           subject to:  $f_i(x) - \hat{g}_i(x; x_n) \leq s_i, \quad i = 0, \dots, m$
                        $s_i \geq 0, \quad i = 0, \dots, m$
    3. Update $\tau$. $\tau_{n+1} := \min(\mu\tau_n, \tau_{\max})$.
    4. Update iteration. $n := n + 1$.
until stopping criterion is satisfied.
```

In this algorithm, s_i is the slack variable, τ_n is the slack variable multiplication coefficient, $\tau_{\max}$ is the user defined maximum slack multiplication coefficient and μ is a user defined coefficient that increases τ_n each iteration. As the cost function is linear the function $\hat{g}_0(x; x_n)$ is equal to zero. Equivalently, the function $\hat{g}_i(x; x_n)$ is equal to zero for the input and state constraints and exists only for the non-convex constraints. The convergence of the algorithm is also shown in [31]. To further reduce computation time this algorithm is extended with the Cutting Planes technique. The method keeps track of a set of active constraints while ignoring constraints that are well satisfied, which usually is a large set. In this thesis instead of including all the active constraints, the technique is modified to include only the constraints that are violated the most. This technique is applied in the first step of the Penalty CCP and its procedure is explained in algorithm 4.

With this algorithm it is ensured that the technique takes only the most restrictive con-

Algorithm 4 Cutting Planes Technique

```
1. Find most severe violations. After step 1. of algorithm 3 separate the linear constraints (input (5.2) and state (5.3), (5.4) constraint), the non-convex constraints (5.9) and the minimum mean velocity constraint (5.5) from each other. From each of these sets find what constraints are violated the most by:
   if $n = 1$
       Based on the optimal solution found in the previous time step $u^*(k-1)$ calculate which state and safety constraints of the current time step $k$ are going to be violated the most for each step of the MPC prediction horizon $H_p$.
   else
       Based on the optimal solution found in the previous CCP iteration $u^*(n-1)$ calculate which state and safety constraints of the current CCP iteration $n$ are going to be violated the most for each step of the MPC prediction horizon $H_p$. If violated constraints are found add them to the ones found in the previous CCP iteration.
   end
2. Optimize. In step 2 of algorithm 3 use only the calculated most restrictive constraints to calculate optimal solution.
```

straints and calculates an optimal solution based on them. Figure 5.2 visually shows how the most restrictive constraints are selected per iteration. In the figure, the x-axis shows the value of the safety constraint (5.9) which is modified as:

$$\left(\hat{d}^{[i,j]}_{\text{safe}} - \hat{d}^{[j]}_{\text{col},(k+m|k)}\right)^2 - \left(s^{[i,\kappa]}_{x,(k+m|k)} - p^{[i,j]}_{\text{col}}\right)^2 \leq 0. \tag{5.12}$$

Value of a point greater than zero means that the safety constraint is violated. The y-axis indicates each scenario. Every column of points (indicated with a different colour) are the steps of the prediction horizon where safety constraints are imposed. Each step of the figure (n = 1,2,3) show the satisfaction of the safety constrain for one agent before the OCP is solved and the red circles indicate the most restrictive constraints that are found and are included. It can be seen that the algorithm keeps the most restrictive constraints found in the previous time step and adds them to the current ones. The figure step (n = 3*) is the safety constraint after the final step of the method using the optimal solution that is later on applied to the plant. This step illustrates that no constraints are violated while not sacrificing optimality as the most restrictive constraints are closer to the borders than in step n = 3.

By reducing the number of constraints that are included in the optimization process, the computation time is significantly reduced especially when compared to the SDP relaxation optimization procedure. To illustrate the computation benefits, the execution times of the SDP relaxation, the Penalty CCP and Penalty CCP extended with the Cutting Planes technique are compared in Table 5.1.

The time advantages of the Penalty CCP with Cutting Planes approach are immense compared to the SDP relaxation algorithm while providing very similar results, shown in Appendix B. In the appendix a comparison between the optimization results coming from both methods is presented. The Penalty CCP optimization method is used in the simulation results described in Chapter 6.

Table 5.1: Execution time comparison

	SDP relaxation		Penalty CCP		Penalty CCP with Cutting Planes	
	mean [s]	max [s]	mean [s]	max [s]	mean [s]	max [s]
Vehicle 1	0.418	4.897	0.146	0.187	0.036	0.049
Vehicle 2	2.105	13.443	0.165	0.406	0.042	0.067
Vehicle 3	2.137	10.512	0.173	0.558	0.044	0.067
Vehicle 4	4.140	**21.123**	0.206	**0.661**	0.052	**0.102**

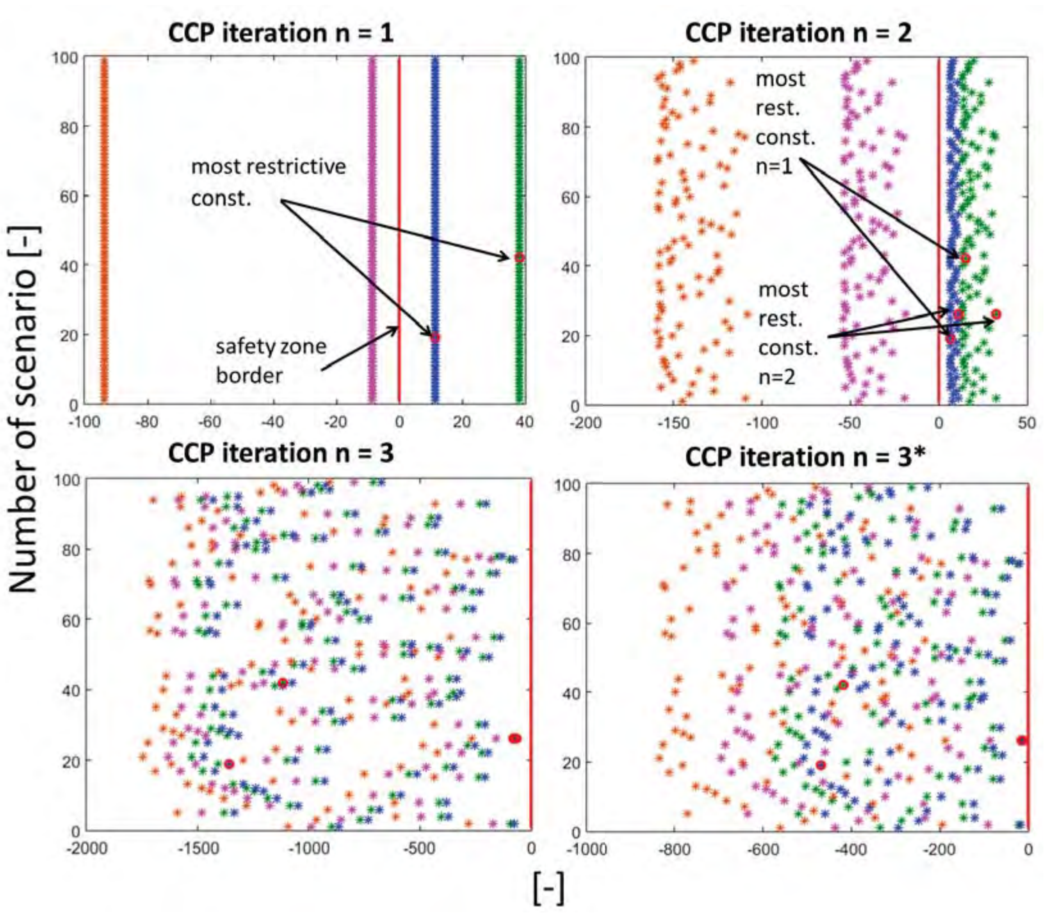

Figure 5.2: Cutting Planes technique for each CCP iteration

5.3 Considering time delay

The final extension of the prediction model is to include an uncertain time delay. The equations shown in section 3.2.3, indicate that the time delay is present in both state and input of the system. Time delay in the model represents the required time for the driver to react to the recommended speed and take the necessary action (brake or accelerate). Appropriate literature is analysed in order to specify the bound from which the time delay is randomly sampled. Based on naturalistic driving data in [1], [22], [46], and [47] it is found that a human driver needs from 0.5 seconds to 2.5 seconds to react, with the 90th percentile usually being at approximately 1.5 seconds. Attentive drivers on the other hand, have a reaction time from 0.5 seconds up to 1 second.

It is known that time delay is frequently a source of instability and deteriorates performance in many systems. A way to counteract these effects is to make closed-loop instead of open-loop predictions. Some stabilization methods have been developed for state and input delay systems via state feedback controllers. In [3] and [30] the problem is reduced to a delay-free system by the Artstein model reduction. [37] propose an LMI-based iterative algorithm in order to design the state feedback stabilization controller. Other approaches such as [24], [33], [51] create a guaranteed cost controller for uncertain time delay systems for both continuous and discrete time systems. However, apart from having an uncertain time delay in both state and input, the prediction model has also additional uncertainties coming from the driver reaction. Dealing with such systems is not greatly covered in the available literature. The two approaches that are found and tried [12], [50] develop a memoryless state feedback guaranteed cost control law. In the approaches first a sufficient condition for the existence of a guaranteed cost control law is derived and then proven that this condition is equivalent to the feasibility of a certain LMI. The feasible solutions to this LMI are used to construct the guaranteed cost controllers. A convex optimization problem is introduced to select the optimal guaranteed cost controller that minimizes the upper bound of the cost function. The difference between [12] and [50] is the LMI that they use, as well as the cost function and the constraints it is subject to. The difference is due to that [12] has a different derived sufficient condition.

5.3.1 Considered approaches and their shortcomings

Both of the above mentioned methods are implemented and in order to validate the implementation it is attempted to replicate the results from the numerical examples. Regarding the approach in [50], the results from the example are closely replicated. When the approach is tried with our system, the results are infeasible, as the method does not seem to be applicable to our type of system. In [12] it is mentioned that the approach in [50] could give infeasible solutions to certain systems and state that their approach overcomes this issue. However, it is not possible to replicate the results from their numerical examples. Therefore, it cannot be guaranteed that the solution with this method for our system is valid. Since no error was found in the implementation, a different approach is considered.

A different approach that is tried is to calculate an optimal gain $\bar{K}$ using the infinite horizon Linear-quadratic regulator for discrete-time state-space system. The calculated gain $\bar{K}$ is such that the state feedback law $u[n] = -\bar{K}x[n]$ minimizes the quadratic cost function:

$$J(u) = \sum_{n=1}^{\infty} \left(x[n]^T Q x[n] + u[n]^T R u[n] + 2x[n]^T N u[n] \right), \tag{5.13}$$

subject to the system dynamics.

A feedback controller causes all system states to approach the equilibrium point, i.e., the acceleration, velocity and position converge to zero. The position and velocity are not desired to be zero as that means the vehicle comes to a standstill. For that reason the system is reduced to a two state system by dropping the position state. The acceleration and velocity state are substituted by the desired operating points ($a_x - 0$ and $v_x - v_{x,lim}$). In essence the modified system states that are used to find a feedback gain are:

$$x_k = \begin{bmatrix} x_{\tau,n|k} \\ x_{\tau,n-1|k} \\ \vdots \\ x_{\tau,1|k} \\ a_x - 0 \\ v_x - v_{x,lim} \end{bmatrix} \tag{5.14}$$

with the delayed states $x_{\tau,n|k}, \ldots, x_{\tau,1|k}$ being the error between the recommended and actual speed and v_{lim} equal to the desired speed to cross the intersection.
The feedback control law is of the following form:

$$u_k^{[i]} = \bar{K}^{[i]} x_k^{[i]} - \bar{K}^{[i]} \begin{bmatrix} 0^{[i]}_{1,\text{time-delay}+1} & v^{[i]}_{x,\text{lim}} \end{bmatrix}^T + u^{[i]}_{k-1} + \Delta u_k^{[i]}, \tag{5.15}$$

with $u^{[i]}_{k-1}$ being the control input from the optimization in the previous time step, or equal to $v^{[i]}_{\text{lim}}$ in the first computation and $\Delta u_k^{[i]}$ being the input compensation computed with the optimization process, i.e. the new control variable that is set by the MPC regime. The $-\bar{K}^{[i]} \begin{bmatrix} 0^{[i]}_{1,\text{time-delay}+1} & v^{[i]}_{x,\text{lim}} \end{bmatrix}^T + u^{[i]}_{k-1}$ is an affine part.

When designing the scenario-based MPC, the system dynamics change with every scenario (κ) due to the random values of the driver parameters and offset driving, it is necessary to calculate the gain $\bar{K}$ for every scenario in order to find the most suitable value per scenario specifications. After computing the gain the closed-loop state matrices are obtained as:

$$\begin{aligned} A_c =& (A_d + B_d \bar{K})^{\kappa} \\ B_c =& B_d \\ E_c =& E_d \\ C_c =& C_d. \end{aligned} \tag{5.16}$$

The controller input u in the original system (3.10) is substituted by (5.15). The affine part is inserted in the prediction of the next free state variable:

$$x_{k+1} = A_c x_k - B_c \bar{K} \begin{bmatrix} 0_{(1,time-delay+1)} & v_{x,\text{lim}} \end{bmatrix}^T + E_c w_k + B_c u_{k-1} + B_c \Delta u_k. \tag{5.17}$$

The state affine feedback controller computed with the LQR method stabilizes the plant. However, as there is a different $\bar{K}$ for every scenario it means that no unique control solution can be found. In essence, the control action that should be applied to the plant cannot be determined unambiguously as there are different control inputs for every scenario:

$$\begin{aligned} u_k^{[1]} &= u_{k-1} - \bar{K}^{[1]} x_0 + \bar{K}^{[1]} x_k + \Delta u_k \\ u_k^{[2]} &= u_{k-1} - \bar{K}^{[2]} x_0 + \bar{K}^{[2]} x_k + \Delta u_k \\ &\vdots \\ u_k^{[N]} &= u_{k-1} - \bar{K}^{[N]} x_0 + \bar{K}^{[N]} x_k + \Delta u_k. \end{aligned} \tag{5.18}$$

In an attempt to calculate the uniform feedback gain for the discrete-time system the problem is formulated as a LMI. The general problem is stated as:

$$eig(A(m) + B(m)\bar{K}) \text{ are inside unit disk,} \tag{5.19}$$

where, $m = 1, \dots, K$ is the respective sampled scenario. To find the feedback gain vector $\bar{K}$, the problem is stated for each realization of the system as a LMI derived from the Lyapunov stability function. In essence:

$$\begin{aligned}
&V(k+1) - V(k) < 0\\
&x(k+1)^T P x(k+1) - x(k)^T P x(k) < 0;\;\; P > 0\\
&x(k)^T A^T P A x(k) - x(k)^T P x(k) < 0;\;\; P > 0\\
&A^T P A - P < 0;\;\; P > 0
\end{aligned} \tag{5.20}$$

Since the closed loop of the system is analysed, the A matrix is equal to $A_c = (A + B\bar{K})$. The inequality problem (5.20) is thus:

$$(A + B\bar{K})^T P (A + B\bar{K}) - P < 0;\;\; P > 0 \tag{5.21}$$

Using the Schur complement formula this inequality is transformed to:

$$\begin{aligned}
&\begin{bmatrix} -P^{-1} & A + B\bar{K} \\ (A + B\bar{K})^T & -P \end{bmatrix} < 0;\;\; P > 0 \Leftrightarrow\\
\Leftrightarrow &\begin{bmatrix} I & 0 \\ 0 & P^{-1} \end{bmatrix} \begin{bmatrix} -P^{-1} & A + B\bar{K} \\ (A + B\bar{K})^T & -P \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & P^{-1} \end{bmatrix} < 0;\;\; P^{-1} > 0 \Leftrightarrow\\
\Leftrightarrow &\begin{bmatrix} -P^{-1} & AP^{-1} + B\bar{K}P^{-1} \\ P^{-1}A^T + P^{-1}\bar{K}^T B^T & -P^{-1} \end{bmatrix} < 0;\;\; P^{-1} > 0
\end{aligned} \tag{5.22}$$

By defining $X_1 = P^{-1}$ and $X_2 = \bar{K}P^{-1}$ (5.22) is written as:

$$\begin{bmatrix} -X_1 & AX_1 + BX_2 \\ X_1 A^T + X_2^T B^T & -X_1 \end{bmatrix} < 0;\;\; X_1 > 0 \tag{5.23}$$

After solving the LMI for X_1 and X_2 the feedback gain is calculated as:

$$\bar{K} = X_2 X_1^{-1} \tag{5.24}$$

The difficulties that occur from this approach are twofold. When implemented, the computational demand of the approach is immense and the current platform runs out of computational memory for more than fifteen scenarios. Furthermore, when the number of scenarios is reduced in order for the computation to be possible, the solver gives an infeasible solution for multiple unstable systems.

As it is not succeeded in creating a feedback gain from the commonly known approaches and the ones from available literature, the scenario MPC is unchanged and makes open-loop predictions. Since creating an approach that is not covered in literature is not in the scope of the thesis, further separate research should be conducted in this topic. The scenario MPC with open-loop predictions can still account for time delay when the drivers are considered to be attentive without violating safety constraints. This is one of the studies shown in the following chapter 6.

5.4 Conclusions

This chapter presents the scenario-based MPC as a control method that is able to account for the parametric uncertainties of the driver reaction model. The Penalty Convex-Concave Procedure which is extended with the Cutting Planes technique significantly improves the computational efficiency without altering the results. Finally, methods for dealing with time delay in the system by means of feedback gain are shown, as well as their shortcomings w.r.t the used system dynamics. It is concluded that further research is needed in this specific topic under these conditions and that the scenario MPC can be used for attentive drivers without feedback gain.

Chapter 6

Simulations

This chapter presents the simulation setup and results of different studies using the control methods described in Chapter 4 and Chapter 5. The first study shows a comparison between the nominal MPC and the Scenario-based MPC in order to illustrate the issues that can occur if an uncertainty unaware control algorithm is used. Furthermore, it shows the benefits of using the Scenario-based MPC in terms of control objectives stated in Chapter 1. In the second study the result from the case when also time delay is present in the prediction model is shown.

The software package where the simulation model is implemented is Matlab and Simulink. The definition of initial conditions and calculation of agent priority and safety distances is done offline. The initial conditions are summarized in Table 6.1. Based on these initial conditions the priority order of the agents and the safety distances are calculated offline based on the methods in section 3.1.1 and in section 4.2. From this calculation the following time-invariant priorities are obtained: $\gamma(1) = 1; \gamma(2) = 2; \gamma(3) = 3; \gamma(4) = 4$.

Table 6.1: Initial conditions

Initial conditions		
Description	**Variable**	**Value**
Initial velocity agent 1-4	$v_0^{[1-4]}$	13.9 [m/s]
Initial position agent 1	$s_0^{[1]}$	-68.3 [m]
Initial position agent 2	$s_0^{[2]}$	-69.0 [m]
Initial position agent 3	$s_0^{[3]}$	-72.3 [m]
Initial position agent 4	$s_0^{[4]}$	-81.3 [m]

The other general parameters are presented in Appendix C. These parameters are used for all simulation scenarios that are analysed.

6.1 Study 1: Nominal MPC vs Scenario-based MPC

The goal of this study is to show that the nominal MPC approach cannot satisfy the safety constraints and with that motivate the necessity of implementing the Scenario-based MPC. Furthermore, it manifests the benefits of using the Scenario-based MPC algorithm when dealing with parameter uncertainties.

6.1.1 Simulation setup

Table 6.2 provides the parameter values used in the nominal MPC algorithm for this particular study. It is considered that no time delay (τ) is present and thus this parameter is equal to zero for both control algorithms. As the nominal MPC cannot account for uncertainties that can occur in the prediction model, a miss-match between the simulation and controller setup in the driver reaction gain and speed offset parameters is expected. The simulation parameters for the gain and speed offset are part of the general parameters shown in Appendix C. To solve the local OCPs, YALMIP with SDPT3 is applied as SDP solver.

Table 6.2: Nominal MPC parameters

Nominal MPC		
Description	**Variable**	**Value**
MPC driver reaction gain- agent 1	$K_d^{[1]}$	0.6 [-]
MPC driver reaction gain- agent 2	$K_d^{[2]}$	1.2 [-]
MPC driver reaction gain- agent 3	$K_d^{[3]}$	0.5 [-]
MPC driver reaction gain- agent 4	$K_d^{[4]}$	1.1 [-]
MPC driver speed offset- agent 1-4	$\Delta v_d^{[1-4]}$	0 [m/s]

The parameters used in the Scenario-based MPC are shown in Table 6.3. By choosing the number of scenarios to be 99, a constraint violation probability of at most 1% can be guaranteed. The assigned speed offset corresponds to a range of $\approx \pm 5$ km/h which is suitable to account for possible uncertainties as well as for HMI design when suggesting a speed range. Regarding the driver gain parameter, [10] suggests a range which in this work is extended in order to consider a more comprehensive reaction possibility. With the range that is used in this thesis, for example an error $v_{\text{ref}} - v$ of 3 m/s with the selected lower and upper bound corresponds to an acceleration of $a_{x,\text{ref}}$ ranging from 0.3 m/s^2 to 3.6 m/s^2. The solver used is qpOASES.

Table 6.3: Scenario-based MPC parameters

Scenario-based MPC		
Description	**Variable**	**Value**
Number of scenarios	K	99 [-]
MPC driver reaction lower bound	$\underline{K}_d^{[i]}$	0.1 [-]
MPC driver reaction upper bound	$\overline{K}_d^{[i]}$	1.2 [-]
MPC driver speed offset lower bound	$\Delta\underline{v}_d^{[i]}$	-1.5 [m/s]
MPC driver speed offset upper bound	$\Delta\overline{v}_d^{[i]}$	1.5 [m/s]

6.1.2 Results

Figure 6.1 illustrates the results coming from the nominal MPC. Each row of the figure represents the results of an agent in terms of path coordinate trajectory, velocity and acceleration shown in each column of the figure respectively. In the velocity column, the actual, maximum (solid black line) and minimum mean velocity are displayed along with the recommended speed and the speed limit. Regarding the path coordinate, a coordinate of zero refers to the first collision point in the reference frame of the agent in consideration. Furthermore, every other agent that might potentially be in conflict with the considered agent is depicted in the same reference frame. In case when the considered agent is in conflict with a higher priority agent, a coloured polygon is displayed that indicates an area that must not be intersected by the trajectory of the considered agent. The fifth row provides an enlarged illustration into these conflict regions.

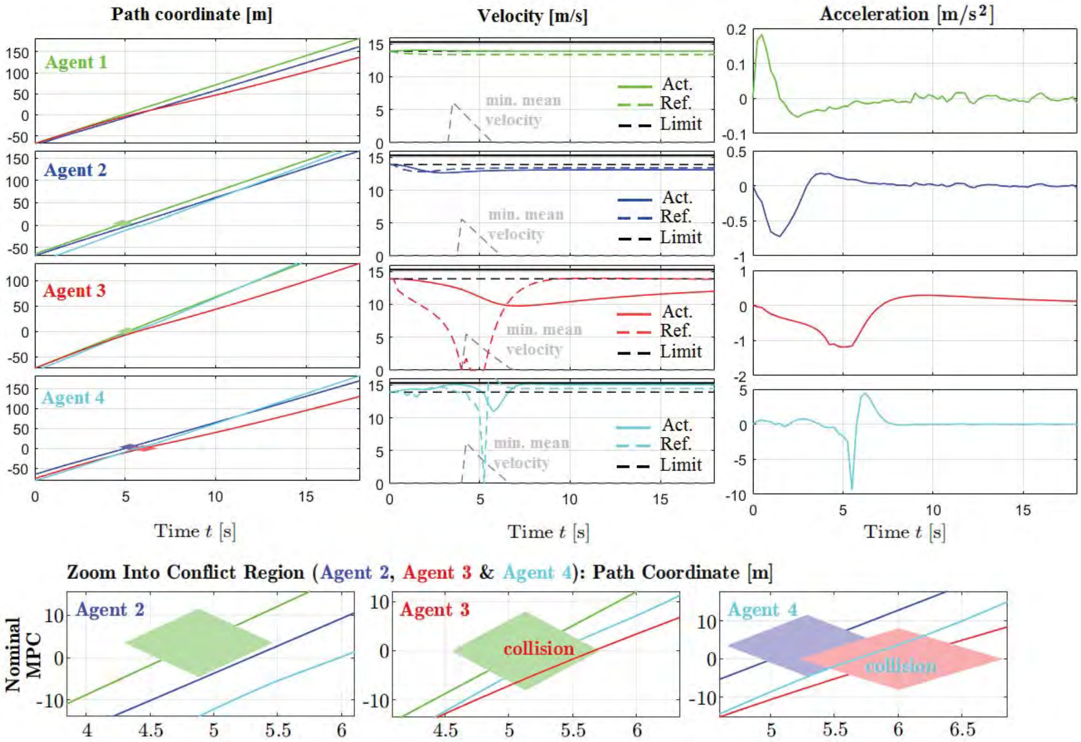

Figure 6.1: Nominal MPC with parametric uncertainties in driver reaction gain and speed offset

Since agent 1 has the highest priority, it crosses the intersection first without the need to react on any other vehicle. The defined speed offsets in the simulation setup result in agent 1 driving slightly faster than the advised speed. Agent 2 and agent 3 both are on collision course with agent 1 and thus have to maintain a safe distance to it. In the fifth row a closer insight into the conflict regions for agent 2 and agent 3 is shown in the first and second column respectively. The trajectory of agent 3 intersects the conflict polygon which indicates that a collision between agent 1 and agent 3 occurs. The cause for the collision is due to the uncertain behaviour of the driver that deviates from the predicted one by the controller. The controller suggests for the vehicle to stop as a possible way to avoid the collision, however since the driver of agent 3 is reacting in a moderate way ($K_d = 0.1$) to the suggested speed it does not manage to avoid the collision. Furthermore, suggesting to make a full stop at that particular time violates the minimum mean velocity constraint. Agent 4 is on a collision course with both agent 2 and agent 3 and since it has the lowest priority it has to yield to both. Due to the parameter uncertainties, the controller suggests increasing the velocity as a way to avoid collision with both agents. However, due to the lethargic reaction of agent 3 the position of that agent is not where the controller predicted and thus a collision between the two agents occurs. In an attempt to avoid the collision the controller abruptly suggests a full stop. This suggestion violates the minimum mean velocity constraint and furthermore its sudden nature results in not satisfying the control objectives in terms of input change and acceleration change.

In summary the nominal MPC causes collisions between multiple agents due to the fact that it cannot account for any possible uncertainties in the drivers behaviour. Furthermore, the control algorithm suggests speeds that are not suitable for driver comfort and efficient crossing in terms of fuel economy.

The results coming from the Scenario-based MPC approach are illustrated in Figure 6.2 with the rows and columns indicating the same parameters as the previously analysed Figure 6.1. As the approach is able to consider uncertainties in the driver reaction, it successfully gives speed recommendations to the drivers of each agent, such that collisions are avoided. A consequence of this approach, however, is reduced performance as the agents keep larger separation from the safety region (polygons in the fifth row) than what is actually required. However, as the control approach needs to account for all potential driver reactions that are specified on the intervals $[\underline{K}_d^{i]}, \overline{K}_d^{[i]}]$ and $[\Delta\underline{v}_d^{i]}, \Delta\overline{v}_d^{[i]}]$ this behaviour is reasonable and expected. Furthermore, the speed recommendations do not have any rapid changes and as such can be followed by a human driver. Moreover, comfortable driving is assured as the resulting accelerations are smooth.

To conclude the results from this simulation study, the scenario-based MPC successfully avoids collisions even when uncertainties are present in the drivers' behaviour. Furthermore, the recommendations in terms of suggested driving speed are smooth and possible to be followed by a human driver. Likewise, the accelerations coming from this approach are such that they do not cause discomfort to the driver. In essence, the scenario approach satisfies all of the control requirements mentioned in section 1.1 and therefore is a suitable approach when dealing with model uncertainties in terms of K_d and Δv_d.

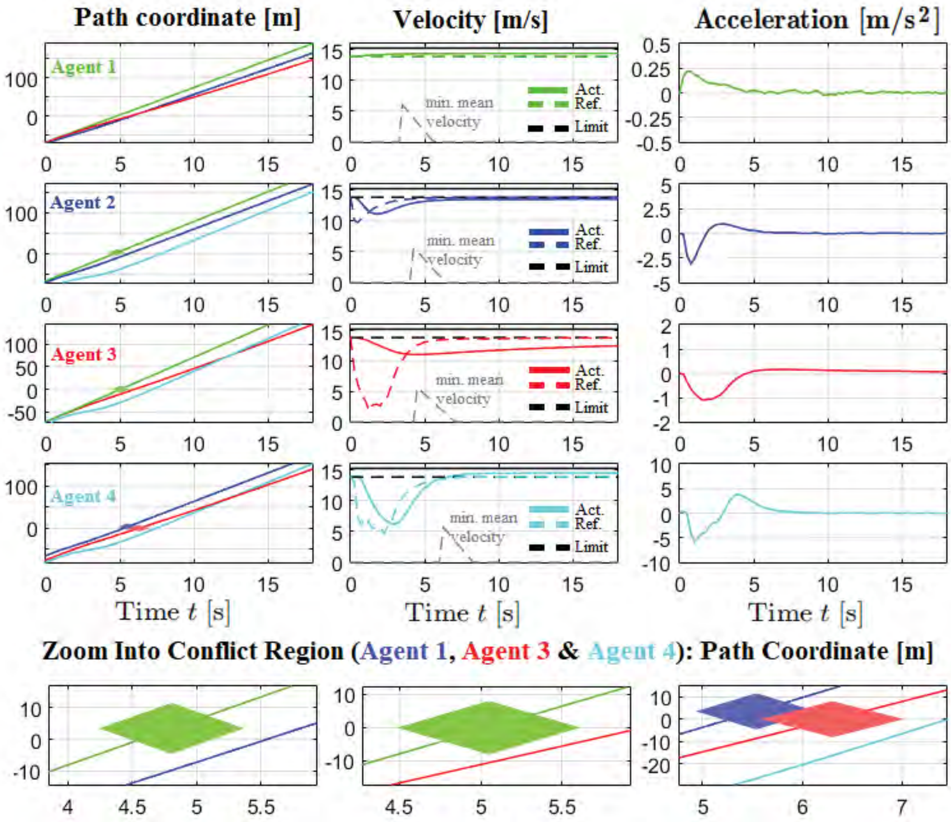

Figure 6.2: Scenario-based MPC with parametric uncertainties in driver reaction gain and speed offset

6.2 Study 2: Scenario-based MPC with uncertain $K_p, \Delta v_d$ and τ

This study shows the ability of the scenario-based MPC to also deal with time delay besides the other parametric uncertainties. An assumption is made however, that only attentive drivers are present, thus limiting the driver reaction to maximum one second. Due to the issues discussed in section 5.3, the result of considering non-attentive drivers and solving the OCP with the same approach is shown in Appendix F.

6.2.1 Simulation setup

The setup for this study differs from the previous such that it is extended by including the ability of the MPC to account for uncertain time delay. The delay parameters used in the simulation and controller are shown in Table 6.4.

6.2.2 Results

Figure 6.3 depicts the results coming from the scenario-based MPC that also considers the delayed reaction of the drivers. The setup of the figure is the same as the previous two figures in this chapter. Due to the fact that the controller needs to account for this delayed reaction,

Table 6.4: Time delay parameters

Time delay parameters		
Description	**Variable**	**Value [s]**
Agent 1 simulation time delay	$\tau^{[1]}$	0.5
Agent 2 simulation time delay	$\tau^{[2]}$	0.75
Agent 3 simulation time delay	$\tau^{[3]}$	1
Agent 4 simulation time delay	$\tau^{[4]}$	1
MPC time delay lower bound	$\underline{\tau}^{[i]}$	0.5
MPC time delay upper bound	$\overline{\tau}^{[i]}$	1

the algorithm in the first time steps suggests a high deceleration to all agents that are on collision course and have lower priority (i.e., agent 2, 3 and 4). By advising to the vehicles to slow down in the beginning of the simulation, the controller allows itself to have more time in the future steps to adjust the speed recommendations such that no collision occurs. As visible in the fifth column of this figure, the algorithm successfully manages to avoid collision for all vehicles. However, when closely looking at the speed recommendations it is noticeable that they have abrupt changes and thus can be difficult to be followed by a human driver-especially for agent 4. The cause for these changes is the satisfaction of the safety constraint and the effect cannot be mitigated by modifying the cost function weights. Furthermore, the resulting accelerations from this approach would cause discomfort and would decrease fuel efficiency.

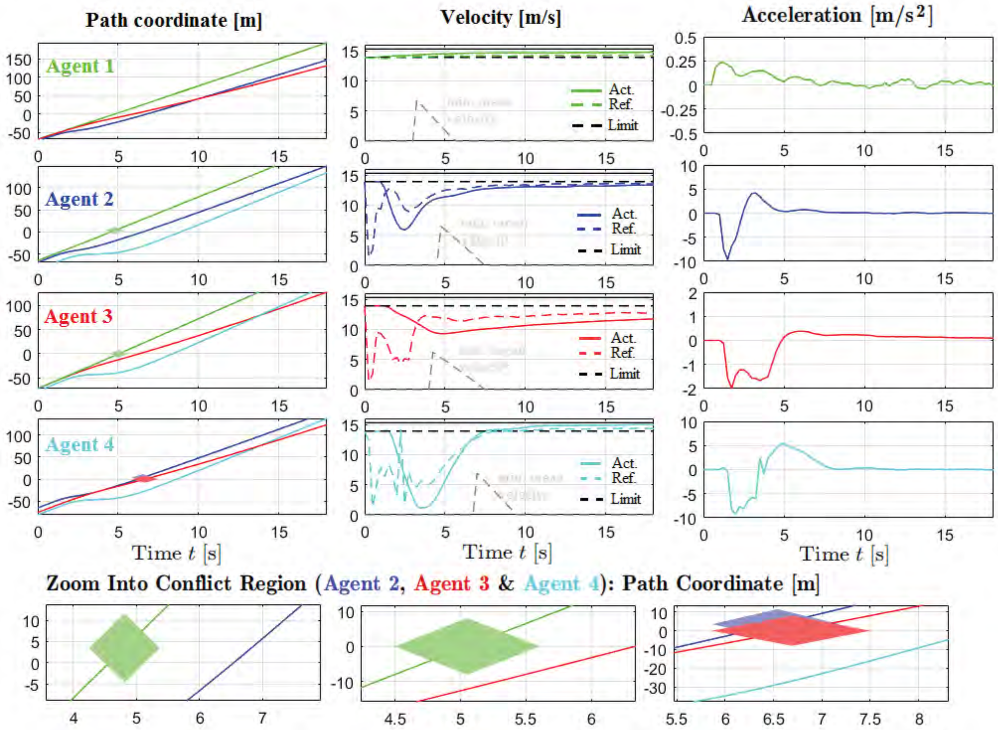

Figure 6.3: Scenario-based MPC with parametric uncertainties in driver reaction gain and time delay, and speed offset

A conclusion that comes from this study is that the scenario-based MPC can avoid collision for all involved agents when the driver reaction time is also considered. However, the acceleration and recommended speed to the driver can have sudden changes. This room for improvement thus presents a suitable motivation for further research into this topic.

Chapter 7

Conclusions and Recommendations

The aim of this research project is to create a distributed stochastic controller for intersection management that would give advices to the driver in terms of vehicle speed. Additionally, the controller needs to be able to consider parametric uncertainties that come from the driver reaction to the suggested speed and to be computationally efficient. This chapter recapitulates the conclusions drawn throughout this report and is followed by recommendations for future research into the topic.

7.1 Conclusions

In this report non-signalized four-way intersections with straight passing vehicles are considered. To improve the safety and lower congestions at such intersections, a stochastic distributed Model Predictive Control method is proposed. As the vehicles are driven by a human driver, this controller issues speed recommendations. The vehicle dynamics are implemented as a double integrator, and the vehicles geometry is modelled as a rectangle. The reaction of the driver consists of a gain parameter that reflects how the driver translates the recommended speed to acceleration (i.e., how hard he/she presses the acceleration/brake pedal) and of a time delay parameter that indicates the time the driver needs to react to the suggestion. Furthermore, to resemble human driving, an offset to the recommended speed is implemented as an additional disturbance to the model.

The control approach that is developed is a scenario-based MPC as this method is able to account for parametric uncertainties that may occur due to the uncertain driver reaction. Each scenario is an independent identically distributed sample of the uncertainty, sampled from a defined bounded interval. The control objectives that need to be met are summarized in a control function and subject to constraints. In order to reduce the computational burden of the algorithm, the Optimal Control Problem is solved using the Penalty Convex-Concave Procedure extended with the Cutting Planes Technique. With this method all of the non-convex constraints are linearised and only the most restrictive constraints are included when solving the problem. The results from this method are indistinguishable when compared to a solver that includes all constraints without linearising them, while immense improvement is achieved w.r.t the computation time. From the first simulation study the advantages of the scenario-based MPC are shown when compared to a uncertainty unaware control algorithm. It is shown that the scenario-based MPC is able to avoid collisions under the considered uncertainties and also satisfy the other control objectives.

It is noticed that introducing time delay into the system significantly decreases performance and could lead into system instability. If only attentive drivers are considered, then the scenario-based MPC can avoid collision for all agents, without the need to design an additional feedback gain as the prediction model is stable. However, a deterioration of performance in terms of suitability of the suggested speed to be followed by a human driver is noticed. This is one of the tasks that can be further investigated in the future and can improve the control algorithm. Further research suggestions are made in the following section.

7.2 Recommendations

A possible extension and improvement for the current scenario-based MPC would be to design a stabilizing feedback gain for the cases when larger time delay is present. The challenge of this task is, as mentioned in the report, to find a uniform gain that stabilizes all realizations of the system. The difficulty of finding such gain is due to the fact that the time delay is present in both state and input of the system and simultaneously the system has uncertain driver reaction parameters in terms of driver gain and speed offset.

Another topic that can be investigated is to make a real world driving study in order to improve the parametrization of the driver reaction model. Drivers with different characteristics (age, driving experience, sex, etc) are needed to cover the range of possible reactions. Furthermore, the study should be conducted at intersection crossings, in order for the results to be more applicable to the algorithm.

Ideally the created control algorithm would be experimentally tested and validated for real world conditions. Implementing such an algorithm to a real-time platform raises additional challenges. Some of the common problems that occur is that running the algorithm on the platform has higher computation time than in the simulation setup. Furthermore, delay or loss of V2X messages can happen and the position from the GPS signal needs to be filtered.

Considering recent developments in the field of Machine Learning and Deep Learning techniques, it is possible to leverage those techniques in this algorithm. The way that these techniques can be implemented is to predict the uncertainty in the driver reaction parameters and then hand over those parameters to the controller.

Bibliography

[1] Kazi Iftekhar Ahmed. *Modeling drivers' acceleration and lane changing behavior*. PhD thesis, Massachusetts Institute of Technology, 1999. 7, 26

[2] Heejin Ahn, Alessandro Colombo, and Domitilla Del Vecchio. Supervisory control for intersection collision avoidance in the presence of uncontrolled vehicles. In *American Control Conference (ACC), 2014*, pages 867–873. IEEE, 2014. 6

[3] Zvi Artstein. Linear systems with delayed controls: a reduction. *IEEE Transactions on Automatic control*, 27(4):869–879, 1982. 27

[4] Fethi Belkhouche. An optimal time strategy for collision avoidance between collaborative agents. In *American Control Conference (ACC), 2017*, pages 1328–1333. IEEE, 2017. 6

[5] Bo Bernhardsson. Sampling of state space systems with several time delays. *IFAC Proceedings Volumes*, 26(2):469–472, 1993. 13

[6] Stephen Boyd and Lieven Vandenberghe. Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization. In *Communications, Computation, Control, and Signal Processing*, pages 279–287. Springer, 1997. 20

[7] Leonardo Bruni, Alessandro Colombo, and Domitilla Del Vecchio. Robust multi-agent collision avoidance through scheduling. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3944–3950. IEEE, 2013. 6

[8] Gabriel R Campos, Paolo Falcone, Henk Wymeersch, Robert Hult, and Jonas Sjöberg. Cooperative receding horizon conflict resolution at traffic intersections. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 2932–2937. IEEE, 2014. 4

[9] Ashwin Carvalho, Stéphanie Lefévre, Georg Schildbach, Jason Kong, and Francesco Borrelli. Automated driving: The role of forecasts and uncertainty–a control perspective. *European Journal of Control*, 24:14–32, 2015. 7, 8, 21, 23

[10] Robert E Chandler, Robert Herman, and Elliott W Montroll. Traffic dynamics: studies in car following. *Operations research*, 6(2):165–184, 1958. 32

[11] Lei Chen and Cristofer Englund. Cooperative intersection management: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, 2016. 4

[12] Wu-Hua Chen, Zhi-Hong Guan, and Xiaomei Lu. Delay-dependent guaranteed cost control for uncertain discrete-time systems with both state and input delays. *Journal of the Franklin institute*, 341(5):419–430, 2004. 27

[13] EH Choi. Crash factors in intersection-related crashes: An on-scene perspective (no. hs-811366). *National Center for Statistics and Analysis. National Highway Traffic Safety Administration (NHTSA), US Department of Transportation*, 2010. 1

[14] Alessandro Colombo and Domitilla Del Vecchio. Efficient algorithms for collision avoidance at intersections. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 145–154. ACM, 2012. 6

[15] Alessandro Colombo and Domitilla Del Vecchio. Least restrictive supervisors for intersection collision avoidance: A scheduling approach. *IEEE Transactions on Automatic Control*, 60(6):1515–1527, 2015. 6

[16] Alexandre dAspremont and Stephen Boyd. Relaxations and randomized methods for nonconvex qcqps. *EE392o Class Notes, Stanford University*, (1):1–16, 2003. 6, 20

[17] Gabriel Rodrigues de Campos, Paolo Falcone, and Jonas Sjoberg. Autonomous cooperative driving: a velocity-based negotiation approach for intersection crossing. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 1456–1461. IEEE, 2013. 4

[18] Ashil Sayyed Farahmand and Lamine Mili. Cooperative decentralized intersection collision avoidance using extended kalman filtering. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 977–982. IEEE, 2009. 5

[19] S Alireza Fayazi, Ardalan Vahidi, and Andre Luckow. Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via milp. In *American Control Conference (ACC), 2017*, pages 4920–4925. IEEE, 2017. 6

[20] Andrew Gray, Yiqi Gao, J Karl Hedrick, and Francesco Borrelli. Robust predictive control for semi-autonomous vehicles with an uncertain driver model. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 208–213. IEEE, 2013. 7

[21] Andrew Gray, Yiqi Gao, Theresa Lin, J Karl Hedrick, and Francesco Borrelli. Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2329–2334. IEEE, 2013. 7, 8

[22] Marc Green. ”how long does it take to stop?” methodological analysis of driver perception-brake times. *Transportation human factors*, 2(3):195–216, 2000. 26

[23] Jean Gregoire, Silvère Bonnabel, and Arnaud De La Fortelle. Priority-based coordination of robots. *CoRR*, abs/1410.0879, 2014. 5

[24] X Guan, Z Lin, and G Duan. Robust guaranteed cost control for discrete-time uncertain systems with delay. *IEE Proceedings-control theory and applications*, 146(6):598–602, 1999. 27

[25] Michael Hafner, Drew Cunningham, Lorenzo Caminiti, and Domitilla Del Vecchio. Automated vehicle-to-vehicle collision avoidance at intersections. In *Proceedings of world congress on intelligent transport systems*, 2011. 6

[26] Michael R Hafner, Drew Cunningham, Lorenzo Caminiti, and Domitilla Del Vecchio. Cooperative collision avoidance at intersections: Algorithms and experiments. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1162–1175, 2013. 6

[27] Michael R Hafner and Domitilla Del Vecchio. Computational tools for the safety control of a class of piecewise continuous systems with imperfect information on a partial order. *SIAM Journal on Control and Optimization*, 49(6):2463–2493, 2011. 6

[28] Martina Joševski, Alexander Katriniok, and Dirk Abel. Scenario mpc for fuel economy optimization of hybrid electric powertrains on real-world driving cycles. In *American Control Conference (ACC), 2017*, pages 5629–5635. IEEE, 2017. 8

[29] Alexander Katriniok, Peter Kleibaum, and Martina Joševski. Distributed model predictive control for intersection automation using a parallelized optimization approach. *IFAC-PapersOnLine*, 50(1):5940–5946, 2017. 2, 6, 16, 20

[30] Wook Kwon and A Pearson. Feedback stabilization of linear systems with delayed control. *IEEE Transactions on Automatic control*, 25(2):266–269, 1980. 27

[31] Thomas Lipp and Stephen Boyd. Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17(2):263–287, 2016. 24

[32] Jan Marian Maciejowski. *Predictive control: with constraints*. Pearson education, 2002. 12

[33] Magdi S Mahmoud and Lihua Xie. Guaranteed cost control of uncertain discrete systems with delays. *International Journal of Control*, 73(2):105–114, 2000. 27

[34] Laleh Makarem and Denis Gillet. Model predictive coordination of autonomous vehicles crossing intersections. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 1799–1804. IEEE, 2013. 5

[35] Alejandro Ivan Morales Medina, Nathan van de Wouw, and Henk Nijmeijer. Cooperative intersection control based on virtual platooning. *IEEE Transactions on Intelligent Transportation Systems*, 2017. 6

[36] Peter Mock. European vehicle market statistics. *Pocketbook: Lugano, Switzerland*, 2014. 1

[37] Young Soo Moon, Poogyeon Park, Wook Hyun Kwon, and Young Sam Lee. Delay-dependent robust stabilization of uncertain state-delayed systems. *International Journal of control*, 74(14):1447–1455, 2001. 27

[38] Nikolce Murgovski, Gabriel Rodrigues de Campos, and Jonas Sjöberg. Convex modeling of conflict resolution at traffic intersections. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 4708–4713. IEEE, 2015. 5

[39] Christopher J Nash, David J Cole, and Robert S Bigler. A review of human sensory dynamics for application to models of driver steering and speed control. *Biological cybernetics*, 110(2-3):91–116, 2016. 7

[40] Kazutoshi Nobukawa. *A model based approach to the analysis of intersection conflicts and collision avoidance systems.* PhD thesis, University of Michigan, 2011. 7

[41] Günther Prokop. Modeling human vehicle driving by model predictive online optimization. *Vehicle System Dynamics*, 35(1):19–53, 2001. 7

[42] Xiangjun Qian, Jean Gregoire, Arnaud De La Fortelle, and Fabien Moutarde. Decentralized model predictive control for smooth coordination of automated vehicles at intersection. In *Control Conference (ECC), 2015 European*, pages 3452–3458. IEEE, 2015. 5

[43] Georg Schildbach and Francesco Borrelli. Scenario model predictive control for lane change assistance on highways. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 611–616. IEEE, 2015. 8, 23

[44] Georg Schildbach, Lorenzo Fagiano, Christoph Frei, and Manfred Morari. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica*, 50(12):3009–3018, 2014. 8, 21, 23

[45] MC Simon, Thierry Hermitte, and Yves Page. Intersection road accident causation: A european view. In *21st International Technical Conference on the Enhanced Safety of Vehicles*, pages 1–10, 2009. 1

[46] Heikki Summala. Brake reaction times and driver behavior analysis. *Transportation Human Factors*, 2(3):217–226, 2000. 26

[47] George T Taoka. Brake reaction times of unalerted drivers. *ITE journal*, 59(3):19–21, 1989. 26

[48] Axel Wolfermann, Wael KM Alhajyaseen, and Hideki Nakamura. Modeling speed profiles of turning vehicles at signalized intersections. In *3rd International Conference on Road Safety and Simulation RSS2011, Transportation Research Board TRB, Indianapolis*, 2011. 7

[49] Henk Wymeersch, Gabriel Rodrigues de Campos, Paolo Falcone, Lennart Svensson, and Erik G Ström. Challenges for cooperative its: Improving road safety through the integration of wireless communications, control, and positioning. In *Computing, Networking and Communications (ICNC), 2015 International Conference on*, pages 573–578. IEEE, 2015. 4

[50] Li Yu and Furong Gao. Optimal guaranteed cost control of discrete-time uncertain systems with both state and input delays. *Journal of the Franklin Institute*, 338(1):101–110, 2001. 27

[51] Li Yu, Furong Gao, and Anke Xue. Guaranteed cost control of uncertain discrete linear time-delay systems. In *American Control Conference (ACC)*, volume 4, pages 2481–2485. IEEE, 2000. 27

Appendix A

Prediction model when delay τ is not a multiple of the sample time T_s

This appendix gives a representation of the prediction model where the time delay is not multiple of the sample time and is an extension of the presented model in Section 3.2.3.

The vehicle dynamics in continuous time for $x(t) = [a_x(t), v_x(t), s_x(t)]^T$ and $\tilde{u}(t) = a_{x,ref}$ are written as:

$$\dot{x}(t) = \begin{bmatrix} -\frac{1}{T_{ax}} & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} \frac{1}{T_a x} \\ 0 \\ 0 \end{bmatrix} \tilde{u}(t) \tag{A.1}$$

Assuming that $u(t) = u(k)$ for $k \leq t \leq k+1$ and $k = kT_s$ it is obtained:

$$\begin{aligned} x(k+1) =& e^{AT_s} x(k) + \int_0^{T_s} e^{As} ds B\tilde{u}(k) \\ =& \begin{bmatrix} e^{-T_s/T_{ax}} & 0 & 0 \\ T_{ax}[1 - e^{-T_s/T_{ax}}] & 1 & 0 \\ T_s T_{ax} - T_{ax}^2[1 - e^{-T_s/T_{ax}}] & T_s & 1 \end{bmatrix} x(k) + \int_0^{T_s} \begin{bmatrix} \frac{1}{T_{ax}} e^{-s/T_{ax}} \\ 1 - e^{-s/T_{ax}} \\ s - T_{ax}[1 - e^{-s/T_{ax}}] \end{bmatrix} ds\tilde{u}(k) \\ =& \begin{bmatrix} e^{-T_s/T_{ax}} & 0 & 0 \\ T_{ax}[1 - e^{-T_s/T_{ax}}] & 1 & 0 \\ T_s T_{ax} - T_{ax}^2[1 - e^{-T_s/T_{ax}}] & T_s & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 - e^{-T_s/T_{ax}} \\ T_s - T_{ax}[1 - e^{-T_s/T_{ax}}] \\ \frac{1}{2}T_s^2 - T_s T_{ax} + T_{ax}^2[1 - e^{-T_s/T_{ax}}] \end{bmatrix} \tilde{u}(k) \end{aligned} \tag{A.2}$$

Let $\tau = nT_s - \tilde{\tau}$ for some positive integer n, where $0 \leq \tilde{\tau} \leq T_s$. Taking the complete reaction of the driver the input to the vehicle dynamics is formulated as:

$$\tilde{u}(k) = a_{x,ref}(k) = K_d[v_{x,ref}(k - nT_s + \tilde{\tau}) - v_x(k - nT_s + \tilde{\tau})] = K_d[v_{x,ref}(k - n + \tilde{\tau}) - v_x(k - n + \tilde{\tau})] \tag{A.3}$$

Since:

$$\begin{bmatrix} e^{-\tilde{\tau}/T_{ax}} & 0 & 0 \\ T_{ax}[1 - e^{-\tilde{\tau}/T_{ax}}] & 1 & 0 \\ \tilde{\tau} T_{ax} - T_{ax}^2[1 - e^{-\tilde{\tau}/T_{ax}}] & T_s & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 - e^{-\tilde{\tau}/T_{ax}} \\ \tilde{\tau} - T_{ax}[1 - e^{-T_s/T_{ax}}] \\ \frac{1}{2}\tilde{\tau}^2 - \tilde{\tau} T_{ax} + T_{ax}^2[1 - e^{-\tilde{\tau}/T_{ax}}] \end{bmatrix} \tilde{u}(k) \tag{A.4}$$

the velocity is written as:

$$v_x(k+\tilde{\tau}) = \begin{bmatrix} T_{ax}(1-e^{-\tilde{\tau}/T_{ax}}) & 1 & 0 \end{bmatrix} x(k) + [\tilde{\tau} - T_{ax}(1-e^{-\tilde{\tau}/T_{ax}})]\tilde{u}(k) \tag{A.5}$$

or:

$$v_x(k-n+\tilde{\tau}) = \begin{bmatrix} T_{ax}(1-e^{-\tilde{\tau}/T_{ax}}) & 1 & 0 \end{bmatrix} x(k-n) + [\tilde{\tau} - T_{ax}(1-e^{-\tilde{\tau}/T_{ax}})]\tilde{u}(k-n) \tag{A.6}$$

Using the extended state $\bar{x}(k)^T = [a_x(k), v_x(k), s_x(k), v_x(k-1+\tilde{\tau}), v_x(k-2+\tilde{\tau}), \dots, v_x(k-n+\tilde{\tau})]^T$ it is obtained:

$$\begin{aligned}\bar{x}(k+1) = &\begin{bmatrix} e^{-T_s/T_{ax}} & 0 & 0 & 0 & \dots & 0 & -K_d[1-e^{-T_s/T_{ax}}] \\ T_{ax}[1-e^{-T_s/T_{ax}}] & 1 & 0 & 0 & \dots & 0 & -K_dT_s + K_dT_{ax}[1-e^{-T_s/T_{ax}}] \\ T_sT_{ax} - T_{ax}^2[1-e^{-T_s/T_{ax}}] & T_s & 1 & 0 & \dots & 0 & -K_d(\frac{1}{2}T_s^2 - T_sT_{ax} + T_{ax}^2[1-e^{-T_s/T_{ax}}]) \\ T_{ax}[1-e^{-\tilde{\tau}/T_{ax}}] & 1 & 0 & 0 & \dots & 0 & \tilde{\tau} - T_{ax}[1-e^{-\tilde{\tau}/T_{ax}}] \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \bar{x}(k) + \\ &K_d \begin{bmatrix} 1-e^{-T_s/T_{ax}} \\ T_s - T_{ax}[1-e^{-T_s/T_{ax}}] \\ \frac{1}{2}T_s^2 - T_sT_{ax} + T_{ax}^2[1-e^{-T_s/T_{ax}}] \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} v_{x,ref}(k-n+\tilde{\tau})\end{aligned} \tag{A.7}$$

or by using the extended state $\bar{x}(k)^T = [a_x(k), v_x(k), s_x(k), K_d(v_{x,ref}(k-1+\tilde{\tau}) - v_x(k-1+\tilde{\tau})), K_d(v_{x,ref}(k-2+\tilde{\tau}) - v_x(k-2+\tilde{\tau})), \dots, K_d(v_{x,ref}(k-n+\tilde{\tau}) - v_x(k-n+\tilde{\tau}))]^T$:

$$\begin{aligned}\bar{x}(k+1) = &\begin{bmatrix} e^{-T_s/T_{ax}} & 0 & 0 & 0 & \dots & 0 & 1-e^{-T_s/T_{ax}} \\ T_{ax}[1-e^{-T_s/T_{ax}}] & 1 & 0 & 0 & \dots & 0 & T_s + T_{ax}[1-e^{-T_s/T_{ax}}] \\ T_sT_{ax} - T_{ax}^2[1-e^{-T_s/T_{ax}}] & T_s & 1 & 0 & \dots & 0 & \frac{1}{2}T_s^2 - T_sT_{ax} + T_{ax}^2[1-e^{-T_s/T_{ax}}] \\ -K_dT_{ax}[1-e^{-\tilde{\tau}/T_{ax}}] & -K_d & 0 & 0 & \dots & 0 & -K_d(\tilde{\tau} - T_{ax}[1-e^{-\tilde{\tau}/T_{ax}}]) \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \bar{x}(k) + \\ &\begin{bmatrix} 0 \\ 0 \\ 0 \\ K_d \\ 0 \\ \vdots \\ 0 \end{bmatrix} v_{x,ref}(k+\tilde{\tau})\end{aligned} \tag{A.8}$$

Appendix B

CCP and SDP solver comparison

To illustrate the similarity of the two approaches (Convex Concave Procedure with Cutting Planes and SDP relaxation) that solve the OCP, a comparison between the input signals is conducted. The setup of the simulation is the same as shown in section 6.1, where also the full results are shown. In the figure B.1 below in the first column the input signal of the controller which solves its OCP with the CCP and SDP procedures is shown for each agent involved in the intersection. The second column depicts the difference between the two solvers.

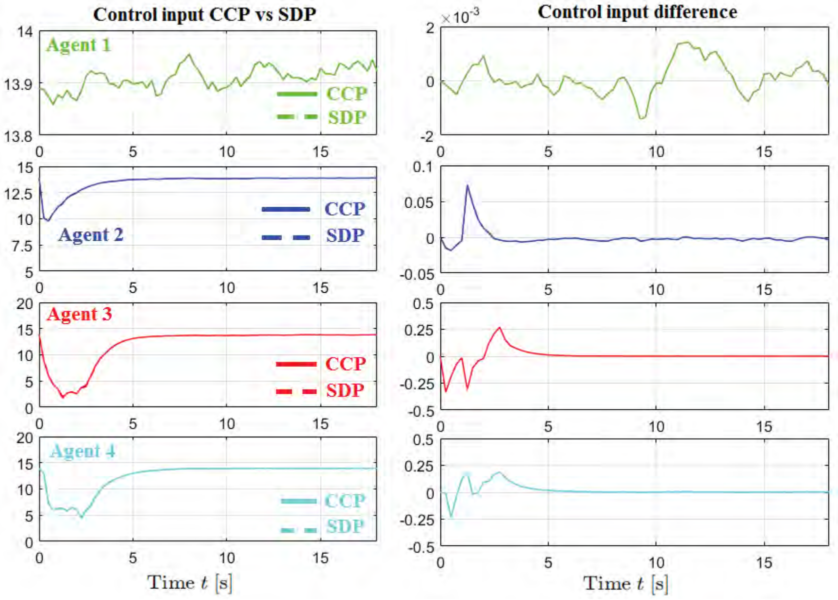

Figure B.1: Control input of CCP and SDP approach and their difference

The control input difference between these two approaches is negligible and thus it can be concluded that the complete results when the control problem is solved with the CCP approach are similar to the SDP results. This analysis supports the motivation to use the CCP with Cutting Planes method as an approach for all further simulation studies as it is more computationally efficient than the SDP and manages to have identical results.

Appendix C

Simulation parameters

In this appendix the simulation parameters that are used for all simulation studies are shown in the table C.1 below. By choosing the sample time to be equal to 0.25 seconds and prediction horizon length of 20 steps, a preview time of 5 seconds is obtained. The length and width of the vehicle correspond to a Ford Mondeo.

Table C.1: Simulation parameters

General parameters		
Description	**Variable**	**Value**
Sampling time	T_s	0.25 [s]
Vehicle length	$L^{[i]}$	4.87 [m]
Vehicle width	$W^{[i]}$	1.85 [m]
Dynamic powertrain constant	T_{ax}	0.3 [s]
Length of prediction horizon	H_p	20 [-]
Length of control horizon	H_u	20 [-]
Weighting factor for control outputs	Q	0.5
Weighting factor for changes of control inputs	R	6
Weighting factor for absolute accelerations	S	1
Weighting factor for acceleration changes (jerk)	T	1
Safety coefficient for safety constraints	S_x	1.2 [-]
Safety distance gap	d_{safe}	8.04 [m]
Desired velocity	v_{des}	13.9 [m/s]
Velocity lower bound	v_{lb}	0 [m/s]
Velocity upper bound	v_{ub}	15.18 [m/s]
Acceleration lower bound	a_{lb}	-9 $[m/s^2]$
Acceleration upper bound	a_{ub}	5 $[m/s^2]$
Simulation driver reaction gain- agent 1	$K_d^{[1]}$	0.55 [-]

Simulation driver reaction gain- agent 2	$K_d^{[2]}$	1.0 [-]
Simulation driver reaction gain- agent 3	$K_d^{[3]}$	0.1 [-]
Simulation driver reaction gain- agent 4	$K_d^{[4]}$	1.2 [-]
Simulation driver speed offset- agent 1	$\Delta v_d^{[1]}$	0.5 [m/s]
Simulation driver speed offset- agent 2	$\Delta v_d^{[2]}$	-0.3 [m/s]
Simulation driver speed offset- agent 3	$\Delta v_d^{[3]}$	-0.7 [m/s]
Simulation driver speed offset- agent 4	$\Delta v_d^{[4]}$	0.6 [m/s]

Appendix D

Influence of uncertain driver reaction parameters

This appendix investigates the uncertainty influence of each driver reaction parameter (driver gain, time delay and speed offset) in order to show the effects that can arise if they are not considered by the control algorithm. Therefore, three simulation studies are shown simulated with the nominal MPC.

D.1 Study 1: Uncertain driver gain K_d

The driver gain in this study for the simulation environment is the same as shown in Table C.1, while the gain in the MPC of each agent is shown in table 6.2. The time delay and speed offset for this study are zero for both MPC and simulation. This way only the influence of a mismatch in driver gain is presented. Figure D.1 presents the results. The rows and columns of the figure indicate the same parameters as described in section 6.1.2.

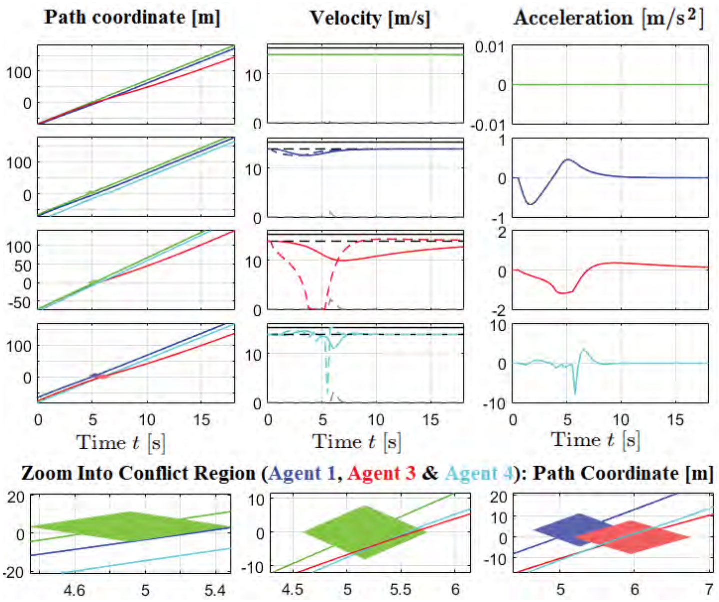

Figure D.1: Influence of uncertain driver gain

As seen from the figure, agent 3 and agent 4 violate the safety constraint and a collision occurs for both agents.

D.2 Study 2: Uncertain reaction time τ

In this study the driver gain is identical for the simulation environment and respective MPC and have values as shown in table C.1, while the speed offset is zero. Each agent have the following time delay parameters:

Table D.1: Time delay parameters

Study 2 parameters		
Description	**Variable**	**Value [s]**
MPC driver delay agent 1	$\tau_{MPC}^{[1]}$	1.5
MPC driver delay agent 2	$\tau_{MPC}^{[2]}$	2.25
MPC driver delay agent 3	$\tau_{MPC}^{[3]}$	1.5
MPC driver delay agent 4	$\tau_{MPC}^{[4]}$	1.25
Simulation driver delay agent 1	$\tau_{Sim}^{[1]}$	1.25
Simulation driver delay agent 2	$\tau_{Sim}^{[2]}$	1.75
Simulation driver delay agent 3	$\tau_{Sim}^{[3]}$	2.5
Simulation driver delay agent 4	$\tau_{Sim}^{[4]}$	1.5

Figure D.2 illustrates the results from this study. As seen, due to the time delay, oscillating behaviour in the control input of agent 2 is visible which is a cause for violation of the velocity and acceleration constraints. Furthermore, agent 2 and agent 3 violate the safety constraints.

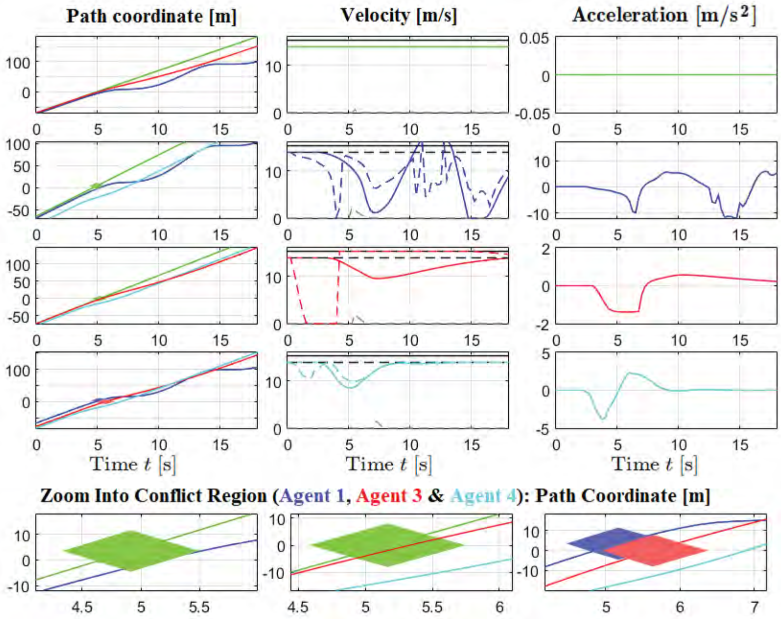

Figure D.2: Influence of uncertain driver reaction time

D.3 Study 3: Uncertain speed offset Δv_d

The last study of this appendix investigates the effects of uncertain speed offset. For that reason mismatch occurs only in the Δv_d parameter. The simulation parameters are indicated in table C.1, while the MPC parameters are as follows: $\Delta v_d^{[1]} = 0.5, \Delta v_d^{[2]} = 0.2, \Delta v_d^{[3]} = 0.4, \Delta v_d^{[4]} = -0.3$. Figure D.3 illustrates the results. When compared to the previous two studies, the effects of mismatch in speed offset has the least severe effects on the overall behaviour of the agents. However, it can still lead to a safety constraint violation that occurs for agent 4, which is highly undesirable.

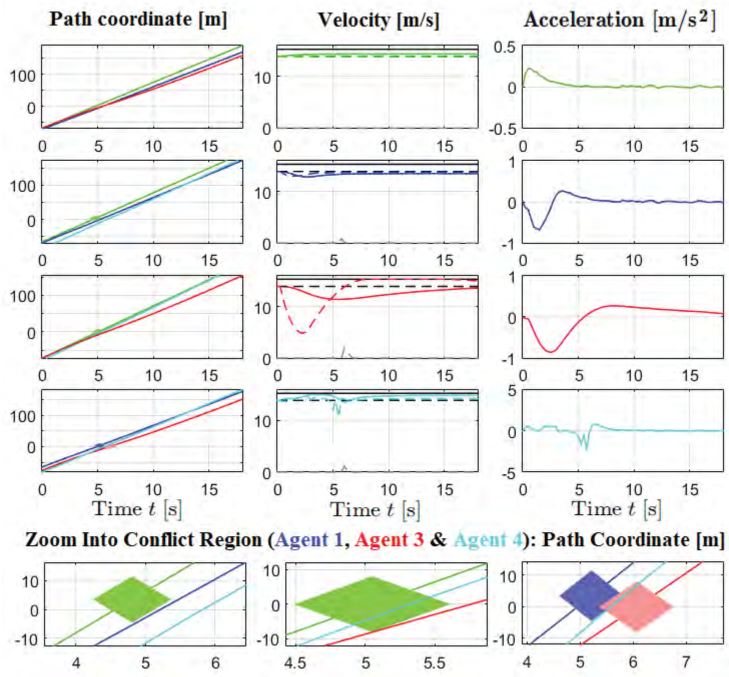

Figure D.3: Influence of uncertain speed offset

Appendix E

Cost function weights sensitivity study

This study analyses the influence of the cost function weighting parameters on the optimisation results. The cost function (4.1) has four parameters (Q, R, S and T) that weight the deviation from the desired speed ($v_{\text{lim}} - v_x$), the input change deviation ($0 - \Delta u$), the acceleration deviation ($0 - a_x$) and jerk deviation ($0 - \Delta a_x$) respectively. The sensitivity analysis is preformed by investigating six cases where the weighting parameters are varied (Table E.1). The first case is a base case where all parameters have equal values. In the second, third and forth case only one parameter is varied in order to investigate how does each parameter influence the results. The fifth case shows one selection where all parameters are different, while the sixth case shows the chosen parameters that are used in the simulation studies and are selected based on the results from the previous case.

Table E.1: Weighting parameters per case

	Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**
Q	1	1	1	1	1	0.5
R	1	1	1	4	4	6
S	1	3	1	1	2	1
T	1	1	4	1	2	1

To investigate the performance of each case the minimum, maximum, mean and root mean square value are computed for each of the weighting objectives. Table E.2 shows the results of this computation. The ideal study has a minimum and maximum value that are closest to zero and as low as possible mean and root mean square value. The green fields on the table indicate the most ideal value for each case. As can be seen the sixth case has the most green fields and excels in performance especially compared to the other case where all parameters are varied.

Table E.2: Parameter sensitivity study

Deviation from speed ($v_{lim} - v_x$)						
	Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**
Minimum	0.0111	0.0111	-0.0021	0.0111	0.0111	0.0111
Maximum	6.8340	6.3894	7.1279	6.2696	6.5565	6.6554
Mean	2.4498	2.7282	2.6235	2.9335	2.8457	2.9377
RMS	3.3839	3.4264	3.7337	3.6682	3.6284	3.7170
Acceleration deviation ($0 - a_x$)						
	Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**
Minimum	-3.4627	-2.7351	-3.3474	-3.1141	-2.8431	-2.4716
Maximum	4.8606	4.8803	7.2956	4.0409	4.1244	4.4224
Mean	0.0134	0.0388	-0.0153	0.0137	0.0222	0.0308
RMS	2.2490	1.9717	2.5412	2.2090	2.0145	1.9245
Jerk deviation ($0 - \Delta a_x$)						
	Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**
Minimum	-0.7221	-0.5933	-1.2281	-0.76430	-0.6686	-0.3243
Maximum	0.5230	0.3994	0.4159	0.3690	0.3038	0.2277
Mean	0.0192	0.0139	-0.0038	0.0137	0.0075	0.0240
RMS	0.2081	0.1693	0.3049	0.2292	0.1851	0.1174
Input change deviation ($0 - \Delta u$)						
	Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**
Minimum	-3.4627	-2.7351	-3.3474	-3.1141	-2.8431	-2.4716
Maximum	4.8606	4.8803	7.2956	4.0409	4.1244	4.4224
Mean	0.0134	0.0388	-0.0153	0.0137	0.0223	0.0308
RMS	2.2490	1.9717	2.5412	2.2090	2.0145	1.9245

Appendix F

Scenario-based MPC with parametric uncertainties under larger time delays

This appendix illustrates the influence of time delay grater than 1 second to the system. More detailed, the setup of this simulation study is shown in table F.1. The selected time delay range is appropriate to a normal human reaction time that occurs and as it is often considered in literature.

Table F.1: Time delay parameters

Time delay parameters		
Description	**Variable**	**Value** [s]
Agent 1 simulation time delay	$\tau^{[1]}$	0.5
Agent 2 simulation time delay	$\tau^{[2]}$	1.25
Agent 3 simulation time delay	$\tau^{[3]}$	1.25
Agent 4 simulation time delay	$\tau^{[4]}$	1.5
MPC time delay lower bound	$\underline{\tau}^{[i]}$	0.5
MPC time delay upper bound	$\overline{\tau}^{[i]}$	2.5

As seen from Figure F.1 the larger time delay causes oscillating behaviour in the control input (agent 2) and is the reason for serious performance deterioration. The created scenario-based MPC cannot counteract these occurrences with the current setup. Therefore, the results from this study motivate the need for further research on how to create a feedback gain that stabilizes the prediction model and resolves these issues.

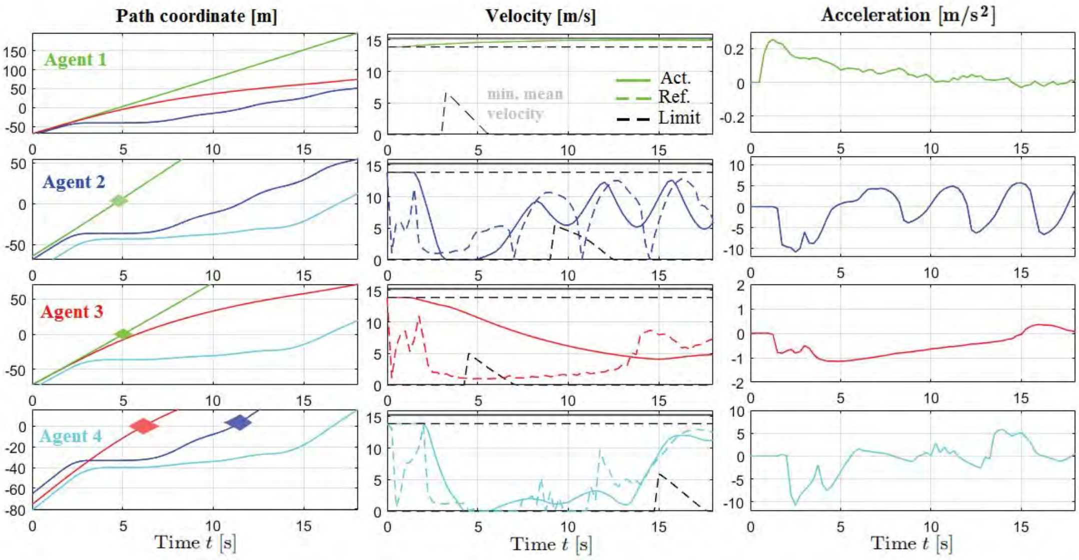

Figure F.1: Influence of larger time delays to the system