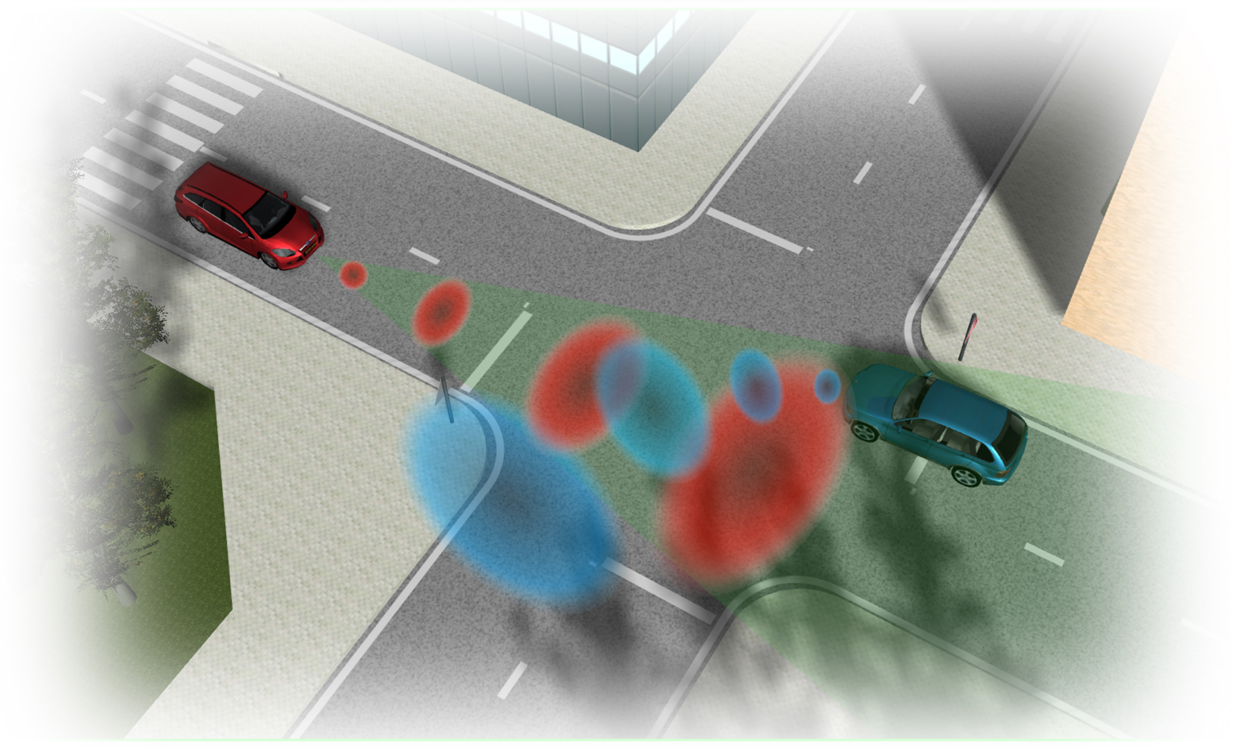
CHALMERS



Collision avoidance at intersections

A probabilistic threat assessment and decision-making system for safety interventions

Adam H. Runarsson F. and Fredrik Granum

Department of Signals & Systems
Chalmers University of Technology
Gothenburg, Sweden 2014
Master's Thesis 2014:9

Title	:	Collision avoidance at intersections - A probabilistic threat assessment and decision-making system for safety interventions
Students	:	Adam H. Runarsson F. and Fredrik Granum
Supervisors	:	Gabriel Rodrigues de Campos, Klas Alenljung and Peter Karlsson
Examiner	:	Paolo Falcone
Test team	:	Jim Lindqvist
Support	:	Denso Sales Sweden and ÅF Technology

Abstract

This thesis has been focusing on developing a collision avoidance system dealing with frontal collisions in cases of intersecting traffic. This has been done by predicting the paths of the included vehicles to generate an occupancy map. This map has been evaluated to define the probability of a collision. Once a collision has been detected the goal has been to avoid it by applying emergency braking as late as possible. These three parts has been solved separately.

The predictions are performed using an Unscented Kalman filter where an effort has been made to capture all possible future paths of a vehicle. The collision detection has been performed by defining a vector space spanning all the possible combinations of positions that could be a collision. From these vectors a probability of a collision could be defined using the joint cumulative distribution. If the calculated probability went high it would trigger the collision avoidance. The collision avoidance is performed using a formal threat assessment algorithm proposed in the thesis which is based on reachability tools. Without loss of generality, no assumptions regarding road geometry has been made.

The developed collision avoidance system was applied to several different scenarios dealing with frontal collisions in cases of intersecting traffic, where the it was able to predict and avoid the collisions.

Acknowledgements

We would like to thank our supervisors Gabriel Rodrigues de Campos, Klas Alenljung and Peter Karlsson for their guidance and support throughout the project. Without their help this thesis would never have come to pass. We would also like to thank Christer Åkerblom for being one of the initiators of the project and Paolo Falcone, our examiner, for his valuable input during our short (read long) and very helpful meetings.

We are also thankful for the support we have gotten from our industrial partners, Denso Sales Sweden and ÅF Technology. Thank you ÅF for the support and thanks to all the staff at Denso for supporting us and making us feel welcome. A special thanks is directed to Jim Lindqvist who was always ready to lend a helping hand, Johan Degerman for listening and providing valuable advise and Yoshihiro Abe who was always working late along side us.

Finally, we would like to thank our girlfriends Inga Rós and Sara Hector for sticking with us, through the good times and the bad.

Adam H. Rúnarson and Fredrik Granum, Göteborg June 2014

Contents

Acronyms

ΔTTC	Collision time interval.
ADAS	Advanced Driver Assistance Systems.
AEBS	Advanced Emergency Braking System.
BVD	Collision avoidance.
CA	Collision avoidance.
CTRA	Constant turn rate and acceleration model.
DM	Decision Making.
DTC	Distance to collision.
EKF	Extended Kalman filter.
ITSC	Intellegent Transportation Systems Conference.
JCD	Joint Cumulative Distribution.
LTAP	Left turn across pat.
SNR	Signal to noise ratio.
TA	Threat Assessment.
TJA	Traffic Jam Assistance.
TTC	Time to collision.
UKF	Unscented Kalman filter.
UT	Unscented transform.
V2V	Vehicle to Vehicle.

Glossary

ΔTTC The time it takes for a vehicle to leave the collision area.

ADAS Systems developed to automate/adapt/enhance vehicle systems for safety and better driving..

AEBS An autonomous safety system which employs sensors to monitor the proximity of vehicles in front and detects situations where the relative speed and distance between the host and target vehicles suggest that a collision is imminent..

BVD A special case of the multivariate normal distribution where only two random variables are considered.

CA Collision avoidance, often used to describe the type of system.

CTRA Constan turn rate and acceleration, describing the assumed behaviour of an object to keep a constant yaw rate and acceleration.

DM Decision Making is refering to the process of evaluating which type of intervention would be the most suiting according to the current threat.

DTC A measurement of how much distance is left until the collision area is reached.

EKF An extension of the Kalman filter to handle non-linear equation by linearizing at each consequitive timstep.

ITSC Annual flagship conference of the IEEE Intelligent Transportation Systems Society.

JCD A mathematical function to calcualte the probability that a random variable $X \sim \mathbb{N}(\mu,\sigma)$ will be found to have a value less than or equal to a certain constant.

LTAP A left turn across path is desribing when a vehicle moving in from the opposite direction makes a left turn which crosses the path of the ego vehicle.

TA Threat Assessment is refering to the process of evaluating how threating a situation is.

TJA An autonomous road vehicle system that keeps distance, regulates speed and can even take control of steering to assist in traffic jams..

TTC The time it takes to reach the collision at given the current conditions.

UKF A variant of the Kalman filter which uses the Unscented Transform to avoid using linearisation of the update and measurement equations.

UT A mathematical function used to estimate a probability density after propagation through a non-linear transformation.

V2V A communication strategy where two or more vehicles can transmit and receive information about each other.

List of Figures

1

Introduction

With the ever growing computational power and sensors becoming cheaper, intelligent transportation systems are expected to become more prominent. In recent years, the focus of such system has mainly been on Advanced Driver Assistance Systems (ADAS). Many car manufacturers have already invested substantial effort in intelligent systems resulting in functionality such as Advanced Emergency Braking System (AEBS) and Traffic Jam Assistance (TJA). However functions dealing with intersecting traffic are still more uncommon in modern vehicles. In 2012, 8.4 % of the lethal collisions and 12.5 % of the collision which led to severe injuries in Sweden [1] can be connected to intersecting traffic situations. Being able to avoid or mitigate these kind of collisions is therefore of high interest. Such a system is a challenging industrial problem and requires extensive theoretical studies to cope with the different aspects that can arise in these kind of situations. The goal of this thesis is to investigate a solution to these problems.

1.1 Context

This project relies on collaboration between Chalmers University of Technology, Denso Sales Sweden AB and the Technology Division at ÅF AB. The idea was to combine their respective knowledge of formal threat assessment, sensors and vehicle active safety systems to develop a robust system for detecting and preventing frontal collisions in cases of intersecting traffic. This initiative resulted in a master thesis that was performed by the two main authors during the first six months of 2014.

1.2 Intersecting traffic

Path intersections in traffic scenarios are daily occurring events for almost all drivers. As common as it is, the risk of a collision is always present. These collision are often avoided by the driver, but in some cases the driver is incapable of reacting in time and hence a collision may occur.

One of the problems encountered, when dealing with intersecting traffic, both for humans and machines, is to determine if a collision will occur. The collision will only occur if both vehicles are at the intersecting sections of their individuals paths during the same time span. These time spans are often short since both vehicles can be moving at high speeds and in opposite directions, which makes it difficult to provide a precise and reliable prediction.

The fast sequence of events leading to a collision can cause other problems as well. Once the collision is detected, it needs to be avoided before becoming inevitable. The safest approach would be to stop the vehicle as soon as a risk of collision is present, but that is not always necessary to avoid the collision. Unnecessary interventions or too conservative ones will never gain acceptance among drivers.

It is clear that the problem of avoiding frontal collisions in cases of intersecting traffic does not have a trivial solution. The problem will hence be divided in to several sub problems. The first challenge will be tracking the vehicles in the surrounding environment to predict the paths of included vehicles to find possible intersections. Once an intersection of paths is found, the next challenge will be focused on finding the most likely time span of a collision. When the time span of the collision has been identified it is necessary to evaluate when to start the intervention, to avoid being at the intersection during this time span.

1.3 General objectives

With the background to the problem given in previous sections the general objectives of this thesis are

1. Track and predict the paths of the surrounding traffic providing a probabilistic, time dependent information of the vehicles positions.
2. Develop an efficient way to detect the risk of a collision by using the acquired information.
3. Assess in a formal and robust way the need for an intervention at the current time instance to minimize the number of unnecessary interventions.
4. Implement the proposed collision avoidance (CA) system in a vehicle.

1.4 Contributions

Considering the general objective presented in the previous section the contributions of this thesis are fourfold and detailed as follows,

- An Unscented Kalman filter (UKF) protocol to predict vehicle trajectories. This has, to the best of the authors knowledge, not been done prior to this project.

- A new way of defining a collision, to calculate the probability of a collision, has been introduced by using vector space.
- A rechability based approach to evaluate when to intervene while accounting for uncertainties of the distance to collision (DTC) and the duration of the collision.
- A CA system for avoiding frontal collisions due to intersecting traffic.

The thesis lead to a conference paper [2] which was presented at the 17^{th} International IEEE Conference on Intelligent Transportation Systems (ITS) in October 2014. It will also serve as a basis for a journal article in 2015 and possibly a follow up thesis proposal.

1.5 Thesis Organization

Chapter 2 provides the reader with information about the objectives and the prerequisites connected to solving the problems in regards to these objectives.Following the description of the problems, the organization of the thesis is presented in a sequential order according to the objectives. Chapter 3 focuses on presenting and explaining the solution used to track and predict the traffic surrounding the ego vehicle. This information is used to generate a probabilistic time dependent position information of the vehicles, which is used to define the risk of a collision. A way to efficiently evaluate this risk of a collision is presented in Chapter 4. Once the collision is detected, an assessment of how immediate the threat is performed, to conclude if an intervention is needed to avoid it. The method for performing this assessment is explained in Chapter 5. The solution to the first three objectives, presented in Chapter 3 to 5, is then be combined and evaluated as a complete CA system. The results of this evaluation are presented in Chapter 6. The thesis will finally be discussed in Chapter 7, where also some suggestions of future work and concluding remarks are stated.

2

Prerequisites

To better understand the problem and the prerequisites, necessary for the technical aspects of this report, this chapter will formulate the problem, define the scenario of interest and describe the desired behaviour of the complete system. Also the vehicle models used throughout the project will be accounted for. As an overview of the chapter, a list of assumptions made will be accounted for in the final section.

2.1 Problem formulation

The main problem is to avoid frontal collisions in cases of intersecting traffic and doing so while minimizing the number of false interventions. The problem can be divided into three sub problems,

- Predict path choices of both the ego and the opposing traffic at such an early time instance that the collision can be avoided.
- Define the most probable time span of a collision.
- Avoid the collision and in an efficient way minimize the number of false interventions.

To verify that the solutions to these problems work together to solve the main problem, they will be combined into one collision avoidance (CA) system. This system will act as a proof of concept of the plausibility and functionality of the solution.

2.2 Test Scenario

To verify the functionality of the system, a test scenario is essential. The suggested scenario is a T intersection where the oncoming traffic can either go left or straight as illustrated in Figure 2.1. The ego vehicle (red) is equipped with the CA system and will evaluate the probability of a collision with the opposing traffic, referred to as the *"other"* vehicle (blue). When a collision is detected, the ego vehicle will need to avoid or, in the worst case, mitigate the collision. Note that no knowledge of the road geometry is assumed here, therefore the tools are developed with generality in mind. However, for the sake if simplicity, it is assumed that the ego vehicle is not turning. Another limitation of the system is introduced in this scenario due to the field of view of frontal sensors. Figure 2.2 illustrates a scenario where a collision is imminent but is outside of sensor range. Due to this limitation, only frontal collision of the ego vehicle, where the ego vehicle collides with the other vehicle's side, are considered.

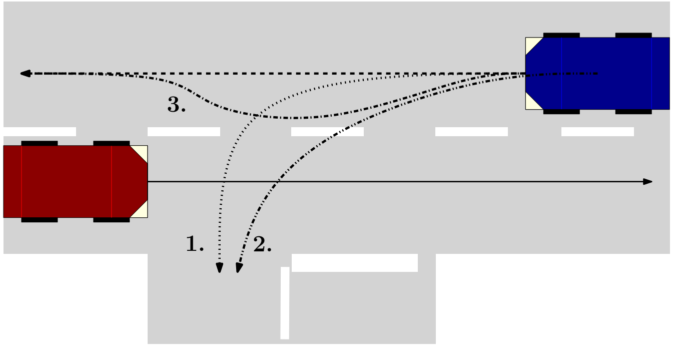


Figure 2.1: Illustration of possible movements of the other (blue) vehicle. Notice the avoidance manoeuvre (dashed-dotted line) which would not lead to a collision but could be detected as one. The ego vehicle is assumed to be going straight and equipped with the CA system. Note, that the road is here displayed for clarity, but no the system has no knowledge of the road geometry.

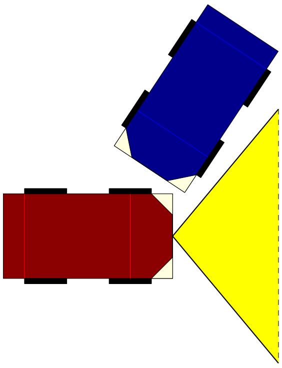


Figure 2.2: Demonstration of a collision that happens outside of sensor range. The sensor range is depicted by the yellow area.

2.3 Desired behaviour of the system

The paths of the other vehicle included in the test scenario, illustrated in Figure 2.1, will test different types of intervention behaviours of the collision avoidance system. Both desired and non-desired ones.

Path 1- The left turn across path (LTAP), initiated late, will give the system a small time to react. If the collision is detected to late, the ego vehicle will only be able to mitigate the collision. This path will hence test the systems ability to predict a collision.

Path 2- If the turn would be initiated earlier, it will likely result in the collision being detected sooner. The detection of a collision does not strictly imply that there exist an immediate need for an intervention. This path will hence test the systems capabilities of triggering an intervention at the correct time instance.

Path 3- The avoidance manoeuvre performed by the other vehicle will generate a situation that is very likely to result in a false intervention. This path will hence test the systems ability to detect collisions in a correct way.

In these three examples, it is clear that there are a number ways to trigger unwanted behaviour of the CA system. In Figure 2.3, the different parts of the system and how they should connect to each other is illustrated.

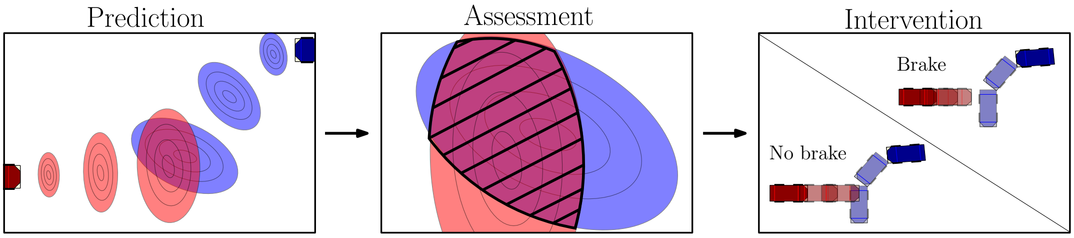


Figure 2.3: Illustration of the different steps needed to be performed for the CA system.

The first part of the system is the path prediction. This part should, as accurately as possible, given the knowledge of the surrounding environment, predict the future paths of the included vehicles. This will provide information that the collision detection should evaluate. The evaluation is done to find out in what time span the collision is most likely to occur. The collision avoidance part, the third part, has the role of assessing if there is a need to start braking or not. When the collision avoidance concludes that the ego vehicle needs to apply emergency braking to avoid the collision, the intervention is triggered. These steps can be visualized by the flowchart in Figure 2.3. For the system to be able to predict the future paths a mathematical description of the vehicles is needed. This description is presented in the next section.

2.4 Vehicle Model

There are many kinds of vehicle models that could be of value, but for the sake of accurate predictions, the model type considered in this thesis assumes constant turn rate and acceleration (CTRA). According to [3], it is a reasonable assumption, that a driver will generally drive with a constant turn rate and acceleration. In this thesis, a well known unicycle model with CTRA to model both cars is used. A more accurate model for the vehicles could be beneficial but is considered to be outside the scope of this thesis. The motion of the ego vehicle is assumed to follow a straight path. This means that speeds and positions of detected vehicles will only need to be compensated in one dimension, that is the in the direction of travel of the ego vehicle. In the case where the ego vehicle is turning, the model equations would need to account for the fictitious forces accompanied by the rotating reference frame. Adding the possibility of the ego vehicle performing a turning manoeuvre will not change the setup of the CA system, only the model equations of the other vehicle. For the sake of simplicity, it is not included in this thesis.

2.4.1 Coordinate frame

Special care must be taken when modelling the other car. The ego vehicle will measure relative to it's own global position, effectively making it a moving reference point. This means that even though an object it perceives is stationary, if the reference point is moving, that object *appears* to be moving. With that in mind, it is reasonable to assume that the ego vehicles sensor position is the origin point of our moving coordinate frame. The frame is Cartesian with the normal convention of the negative and positive axes. The y-axis is considered as the direction where the ego vehicle is heading and the angle convention is illustrated in Figure 2.4. Note, that the coordinate frame for the prediction models of the ego and other vehicle does not move with the predicted position of the ego vehicle. This means that all predicted positions are made relative to the ego vehicles *current* position, see Figure 2.5.

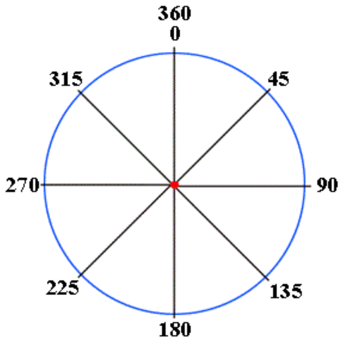


Figure 2.4: Illustrates the angle convention of the models, with the y-axis of the ego vehicle always lying on the 0° direction and the other vehicle defined according to that current frame.

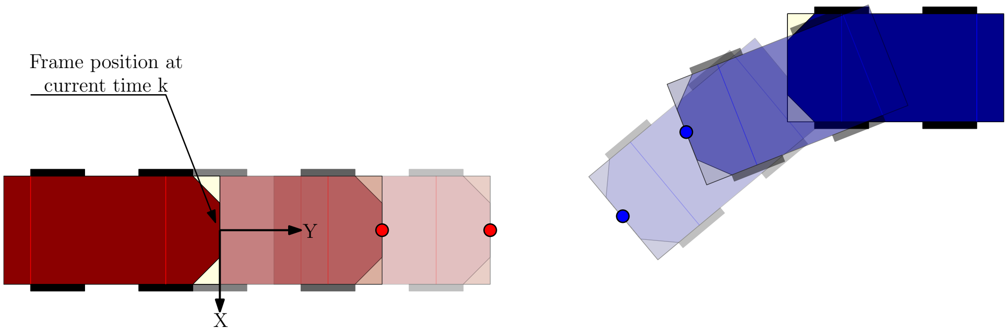


Figure 2.5: Reference frame is positioned at the current position of the ego vehicle and all prediction, represented with coloured circles, are made w.r.t that frame.

2.4.2 Sensors

Since the cars are interacting with an unknown environment, it is necessary to introduce how the ego vehicle senses it. Figure 2.6 illustrates how the ego vehicle measures the relative position of the other vehicle by range distance R and the azimuth angle ϕ which are common among frontal sensors. Along with the range and azimuth angle, the speed, heading and length of the other vehicle are measured. The ego vehicle is equipped with inertial sensors that measure its own speed, acceleration and yaw rate.

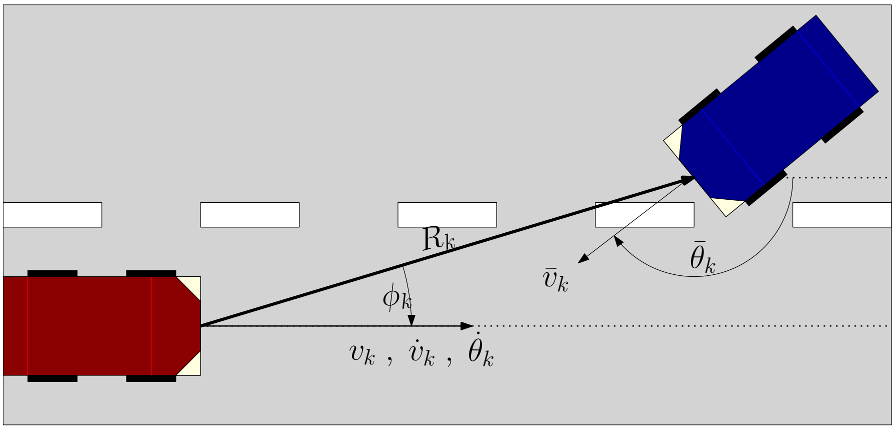


Figure 2.6: Illustration of the assumed measurement convention. $(\bar{.})$ signify that the measured parameter applies for the other vehicle.

2.4.3 Ego Vehicle

The model used to represent ego vehicle is generally called the unicycle model, given by:

$$\begin{bmatrix} \dot{a}(t) \\ \dot{V}(t) \\ \ddot{\theta}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a(t) \\ V(t) \\ \dot{\theta}(t) \\ \theta(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} J(t) \\ \dot{\omega}(t) \end{bmatrix}, \tag{2.1}$$

where a, v and θ are acceleration, speed and heading respectively. The input $J(t)$ is the change in acceleration, often referred to as jerk, and $\dot{\omega}(t)$ the change in yaw rate. The discrete version of the model is given in Equation (2.2) with a step size of ΔT

$$\begin{bmatrix} a_{k+1} \\ V_{k+1} \\ \theta_{k+1} \\ \dot{\theta}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \Delta T & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_k \\ V_k \\ \theta_k \\ \dot{\theta}_k \end{bmatrix} + \begin{bmatrix} \Delta T & 0 \\ \frac{\Delta T^2}{2} & 0 \\ 0 & \frac{\Delta T^2}{2} \\ 0 & \Delta T \end{bmatrix} \begin{bmatrix} J_k \\ \dot{\omega}_k \end{bmatrix} \tag{2.2}$$

where k denotes the time step. Now let $\chi_k = \begin{bmatrix} a_k & v_k & \theta_k & \dot{\theta}_k \end{bmatrix}^T$ and $u_k = \begin{bmatrix} J_k & \dot{\omega}_k \end{bmatrix}^T$ then Equation (2.2) can be rewritten as

$$\chi_{k+1} = f(\chi_k, u_k). \tag{2.3}$$

Let γ_k represent the measured states at time step k, then the measurement model can be written as

$$\gamma_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \chi_k + \begin{bmatrix} \sigma_a \\ \sigma_V \\ \sigma_\theta \\ \sigma_{\dot{\theta}} \end{bmatrix} = H\chi_k + W, \tag{2.4}$$

where σ_a,σ_V,σ_θ,$\sigma_{\dot{\theta}}$ denote measurement noise of the acceleration, speed, heading and yaw rate respectively. As mentioned before, the ego vehicle is the centre of the reference system and therefore no positional change is modelled.

2.4.4 Other vehicle

It is necessary to keep in mind that the reference frame is moving w.r.t the ego vehicle. As such, this *fictitious* movement needs to be captured in the kinematic equations along with the actual movement of the other vehicle. This was done in such a way, that the model separates the effect of the ego vehicles speed from the other vehicles movement, that is, *relative speed* is not being estimated, but the actual speed of the other vehicle. Now, consider Figure 2.7 where it is necessary to model each movement separately and

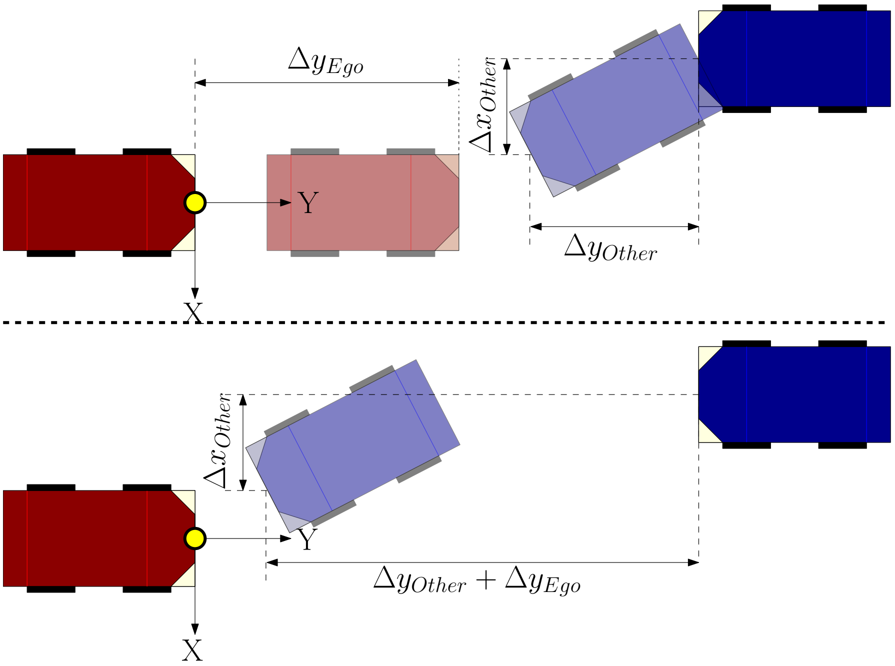


Figure 2.7: Difference between a moving reference frame (above) and an unmoving one (below). The sensor will always perceive the total movement of both vehicles, as such it is considered to have a moving reference frame. The sensor perceives that the other vehicle has moved the total distance of both vehicles.

later combine them to describe the complete perceived movement. Equation (2.5) defines how the position of the ego vehicle changes.

$$\begin{aligned} x_{k+1} &= sin(\theta_k + \dot{\theta}_k \Delta T + \dot{\omega}_k \frac{\Delta T^2}{2})(V_k + \frac{a_k \Delta T}{2} + J_k \frac{\Delta T^2}{3})\Delta T, \\ y_{k+1} &= cos(\theta_k + \dot{\theta}_k \Delta T + \dot{\omega}_k \frac{\Delta T^2}{2})(V_k + \frac{a_k \Delta T}{2} + J_k \frac{\Delta T^2}{3})\Delta T. \end{aligned} \tag{2.5}$$

Without loss of generality, the ego vehicle is assumed to be heading straight. This simplifies Equation (2.5) to

$$\begin{aligned} x_{k+1} &= 0 \\ y_{k+1} &= (V_k + \frac{a_k \Delta T}{2} + J_k \frac{\Delta T^2}{3})\Delta T. \end{aligned} \tag{2.6}$$

Now, let $(\bar{.})$ signify that the state variable applies for the other vehicle, then similarly the movement is described as

$$\begin{aligned} \tilde{x}_{k+1} &= sin(\bar{\theta}_k + \dot{\bar{\theta}}_k \Delta T + \dot{\bar{\omega}}_k \frac{\Delta T^2}{2})(\bar{V}_k + \frac{\bar{a}_k \Delta T}{2} + \bar{J}_k \frac{\Delta T^2}{3})\Delta T, \\ \tilde{y}_{k+1} &= cos(\bar{\theta}_k + \dot{\bar{\theta}}_k \Delta T + \dot{\bar{\omega}}_k \frac{\Delta T^2}{2})(\bar{V}_k + \frac{\bar{a}_k \Delta T}{2} + \bar{J}_k \frac{\Delta T^2}{3})\Delta T. \end{aligned} \tag{2.7}$$

This gives the total perceived change in position of the other vehicle shown in Equation (2.8)

$$\begin{aligned} \bar{x}^f{}_{k+1} &= \bar{x}^f{}_k + \tilde{x}_{k+1} - x_{k+1} \\ \bar{y}^f{}_{k+1} &= \bar{x}^f{}_k + \tilde{y}_{k+1} - y_{k+1} \end{aligned} \tag{2.8}$$

where the superscript f denotes the *Front bumper.* Now it is of interest to estimate the position of the other vehicle's rear bumper. Assume that L_k represents the measured length of the other vehicle, then the position of the rear bumper is expressed as

$$\begin{aligned} \bar{x}^r_{k+1} &= \bar{x}^f{}_{k+1} - sin(\bar{\theta}_k + \dot{\bar{\theta}}_k \Delta T + \dot{\bar{\omega}}_k \frac{\Delta T^2}{2})(\bar{L}_k), \\ \bar{y}^r_{k+1} &= \bar{y}^f{}_{k+1} - cos(\bar{\theta}_k + \dot{\bar{\theta}}_k \Delta T + \dot{\bar{\omega}}_k \frac{\Delta T^2}{2})(\bar{L}_k), \end{aligned} \tag{2.9}$$

denoted with the superscript r. These two positional states combined with the following states

$$\begin{bmatrix} \bar{a}_{k+1} \\ \bar{V}_{k+1} \\ \dot{\bar{\theta}}_{k+1} \\ \bar{\theta}_{k+1} \\ a_{k+1} \\ V_{k+1} \\ \bar{L}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta T & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta T & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta T & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{a}_k \\ \bar{V}_k \\ \dot{\bar{\theta}}_k \\ \bar{\theta}_k \\ a_k \\ V_k \\ \bar{L}_k \end{bmatrix} + \begin{bmatrix} \Delta T & 0 & 0 \\ \frac{\Delta T^2}{2} & 0 & 0 \\ 0 & \Delta T & 0 \\ 0 & \frac{\Delta T^2}{2} & 0 \\ 0 & 0 & \Delta T \\ 0 & 0 & \frac{\Delta T^2}{2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{J}_k \\ \bar{\omega}_k \\ J_k \end{bmatrix}, \tag{2.10}$$

give the total model of the other vehicle. As before, let $\bar{u}_k = \begin{bmatrix} \bar{J}_k & \bar{\omega}_k & J_k \end{bmatrix}^T$ and $\bar{\chi}_k = \begin{bmatrix} \bar{a}_k & \bar{v}_k & \bar{\theta}_k & \dot{\bar{\theta}}_k & a_k & v_k & \bar{L}_k & \bar{x}^f_k & \bar{y}^f_k & \bar{x}^r_k & \bar{y}^r_k \end{bmatrix}^T$, then equations (2.10), (2.9) and (2.8) can be combined as

$$\bar{\chi}_{k+1} = \bar{f}(\bar{\chi}_k, \bar{u}_k). \tag{2.11}$$

Now consider again Figure 2.6. From the positional states x^f_k and y^f_k the range and azimuth angle can be calculated from the states as

$$\begin{aligned} R_k &= \sqrt{(\bar{x}^f{}_k)^2 + \bar{y}^f{}_k)^2}, \\ \phi_k &= atan(\frac{\bar{x}^f{}_k}{\bar{y}^f{}_k}). \end{aligned} \tag{2.12}$$

This leads to the following non-linear measurement equation

$$\bar{\gamma}_k = \begin{bmatrix} \bar{V}_k \\ \bar{\theta}_k \\ R_k \\ \phi_k \\ \bar{L}_k \\ \bar{a}_k \\ \bar{V}_k \end{bmatrix} + \begin{bmatrix} \bar{\sigma}_V \\ \bar{\sigma}_\theta \\ \bar{\sigma}_R \\ \bar{\sigma}_\phi \\ \bar{\sigma}_L \\ \bar{\sigma}_a \\ \bar{\sigma}_V \end{bmatrix} = \bar{H}(\bar{\chi}_k) + \bar{W}, \tag{2.13}$$

where $\bar{\gamma}_k$ are the measured states and $\bar{H}$ is the measurement equation. The $\bar{\sigma}_V$, $\bar{\sigma}_\theta$, $\bar{\sigma}_R$, $\bar{\sigma}_\phi$ and $\bar{\sigma}_L$ denote the measurement noise in the speed, heading, range, azimuth and length of the other vehicle respectively and σ_a and σ_V are the previously defined measurement noise of the ego vehicle.

2.4.5 Prediction models

The prediction models are different from the previously defined models, as there is no moving reference frame needed. The current states of the vehicles are used to estimate the future positions of the vehicles in a global reference frame, with the origin set at the current position of the ego vehicle.

2.4.5.1 Ego vehicle

The prediction model for the ego vehicle is similar to the one defined for filtering, with the added positional change. Let $(\hat{.})$ signify that the state in question is an estimate of said state. Then the prediction model is defined as,

$$\begin{aligned} \begin{bmatrix} \hat{a}_j \\ \hat{V}_j \\ \hat{\theta}_j \\ \hat{\dot{\theta}}_j \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ j\Delta T & 1 & 0 & 0 \\ 0 & 0 & 1 & j\Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} a_k \\ V_k \\ \theta_k \\ \dot{\theta}_k \end{bmatrix} + \begin{bmatrix} j\Delta T & 0 \\ \frac{(j\Delta T)^2}{2} & 0 \\ 0 & \frac{(j\Delta T)^2}{2} \\ 0 & j\Delta T \end{bmatrix}, \begin{bmatrix} J_k \\ \dot{\omega}_k \end{bmatrix} \\ \hat{x}_j &= x_k + sin(\theta_k + \dot{\theta}_k j\Delta T + \dot{\omega}_k \frac{(j\Delta T)^2}{2})(V_k + \frac{a_k j\Delta T}{2} + J_k \frac{(j\Delta T)^2}{3}) j\Delta T \\ \hat{y}_j &= y_k + cos(\theta_k + \dot{\theta}_k j\Delta T + \dot{\omega}_k \frac{(j\Delta T)^2}{2})(V_k + \frac{a_k j\Delta T}{2} + J_k \frac{(j\Delta T)^2}{3}) j\Delta T, \end{aligned} \tag{2.14}$$

where j denotes the state at j steps from the current time k.

Now, since it is of interest to know the future position of the ego vehicle in this global frame, it is necessary to introduce the positional change. The state vector χ_k is augmented with the current position, which as before, is the origin point. Let $(.)'$ specify a variable that is augmented with zeros, then the state vector is expressed as

$$\chi'_k = \begin{bmatrix} \chi_k & 0 & 0 \end{bmatrix}^T, \tag{2.15}$$

and the model can be defined compactly as

$$\hat{\chi}_j = \hat{f}(\chi'_k, u_k, j). \tag{2.16}$$

2.4.5.2 Other vehicle

The prediction model for the other vehicle is defined as follows,

$$\begin{bmatrix} \hat{\bar{a}}_j \\ \hat{\bar{V}}_j \\ \hat{\dot{\bar{\theta}}}_j \\ \hat{\bar{\theta}}_j \\ \hat{\bar{L}}_j \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ j\Delta T & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & j\Delta T & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{a}_k \\ \bar{V}_k \\ \dot{\bar{\theta}}_k \\ \bar{\theta}_k \\ \bar{L}_k \end{bmatrix} + \begin{bmatrix} j\Delta T & 0 \\ \frac{(j\Delta)T^2}{2} & 0 \\ 0 & j\Delta T \\ 0 & \frac{(j\Delta T)^2}{2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{J}_k \\ \dot{\bar{\omega}}_k \end{bmatrix} \tag{2.17}$$

where the change of the front and rear bumper are now

$$\begin{aligned} \hat{\bar{x}}_j^f &= \bar{x}_k^f + sin(\bar{\theta}_k + \dot{\bar{\theta}}_k j\Delta T + \dot{\bar{\omega}}_k \frac{(j\Delta T)^2}{2})(\bar{V}_k + \frac{\bar{a}_k j\Delta T}{2} + \bar{J}_k \frac{(j\Delta T)^2}{3})j\Delta T, \\ \hat{\bar{y}}_j^f &= \bar{y}_k^f + cos(\bar{\theta}_k + \dot{\bar{\theta}}_k j\Delta T + \dot{\bar{\omega}}_k \frac{(j\Delta T)^2}{2})(\bar{V}_k + \frac{\bar{a}_k j\Delta T}{2} + \bar{J}_k \frac{(j\Delta T)^2}{3})j\Delta T, \\ \hat{\bar{x}}_j^r &= \bar{x}_k^r + sin(\bar{\theta}_k + \dot{\bar{\theta}}_k j\Delta T + \dot{\bar{\omega}}_k \frac{(j\Delta T)^2}{2})((\bar{V}_k + \frac{\bar{a}_k j\Delta T}{2} + \bar{J}_k \frac{(j\Delta T)^2}{3})j\Delta T - \bar{L}_k), \\ \hat{\bar{y}}_j^r &= \bar{y}_k^r + cos(\bar{\theta}_k + \dot{\bar{\theta}}_k j\Delta T + \dot{\bar{\omega}}_k \frac{(j\Delta T)^2}{2})((\bar{V}_k + \frac{\bar{a}_k j\Delta T}{2} + \bar{J}_k \frac{(j\Delta T)^2}{3})j\Delta T - \bar{L}_k), \end{aligned} \tag{2.18}$$

and as before, j denotes the state at j steps from the current step k. The prediction model is then compactly defined as,

$$\hat{\bar{\chi}}_j = \hat{\bar{f}}(\bar{\chi}_k, \bar{u}_k, j). \tag{2.19}$$

2.4.6 State Constraints

To improve the accuracy of the models, state and input constraints has been introduced.

Speed and acceleration

As previously stated, the vehicles are considered not capable of backward manoeuvres, thus the minimum speed will be 0 km/h while the maximum speed is the maximum speed limit of an urban area, or 50 km/h in this project. The speed is strictly related to acceleration, that is, the available acceleration depends on the current speed of the vehicle. A model for this is given in [4] which is shown in Equation (2.20).

$$a_{available}(v(t)) = \frac{{}^{k}\!/_{v(t)} + \alpha_f}{2} + \frac{{}^{k}\!/_{v(t)} - \alpha_f}{2} \tag{2.20}$$

where α_f is maximum acceleration w.r.t to road friction and k is a parameter that describes engine power. Where these values are given in the paper as $\alpha_f = 9.1{}^{m}\!/_{s^2}$ and $k = 66.6{}^{m^2}\!/_{s^3}$. Equation (2.20) is valid in the span,

$$-\alpha_f \leq a_{available}(v(t)) \leq \alpha_f. \tag{2.21}$$

However, the maximum deceleration is often not the same as the maximum acceleration, which leads to the following equation

$$-a_{max} \leq a_{available}(v(t)) \leq \alpha_f. \tag{2.22}$$

where the maximum deceleration, $-a_{max}$, should be evaluated specifically for each vehicle, since it important value for use in braking manoeuvres.

Yaw

Without knowing the road geometry, it will be difficult to constrain the heading between a specific angle interval. Therefore, when the road geometry is unknown, the oncoming vehicle is simply predicted to lie within [$-\infty$,∞] which is the case in this project. If information about the road geometry is available, the allowed yaw rate can be constrained depending on the surrounding environment. Consider Figure 2.8, clearly, the knowledge of road geometry could benefit for example in the case of an avoidance manoeuvre. Knowing that there is no turn available to the other vehicle, the allowable future yaw can be constrained such that the other vehicle will stay in it's own lane with some certainty.

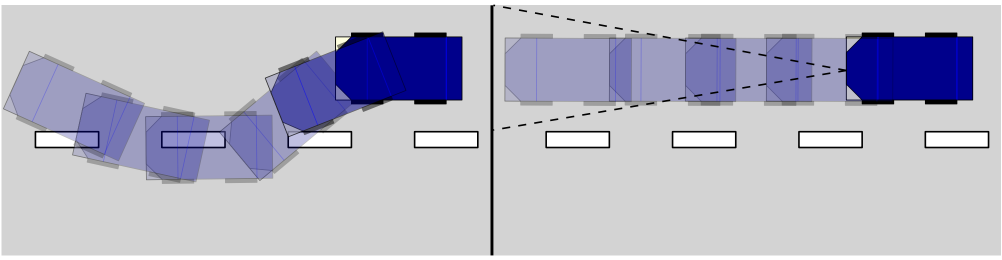


Figure 2.8: Illustrating possible effect of constraining the yaw/yaw rate of predictions (right) as opposed to no constraints (left).

Yaw rate

From [4], an expression for the maximum yaw rate is constrained with

$$\dot{\theta}(t) = \begin{cases} {}^{v(t)sin(\omega\theta_{max})}\!/_{L}, & \text{if } v \leq v_{lat} \\ {}^{\alpha_f c}\!/_{v(t)} & \text{otherwise} \end{cases} \tag{2.23}$$

where v_{lat} is the breakpoint velocity where the longitudinal acceleration switches from maximum steering angle to tire-to-road friction is set at $6.76{}^{m}\!/_{s^2}$ and ω denotes either $[-1{,}1]$, depending if it's a left or right turn. In this work, the yaw rate constraint will be considered to lie within the values of a left and right turn.

In the same way for the yaw, knowing the road geometry would provide better constraints for the yaw rate. Assume the function K(d(t)) gives the curvature of the future path of a vehicle when driven a distance d(t), then the yaw rate needed at that time with speed V(t) is given by

$$\dot{\theta}(t)_C = K(d(t))V(t). \tag{2.24}$$

This equation can be used to constrain the yaw rate at time t. An example of such a constraint could perhaps be

$$\Big[(1-p)\dot{\theta}(t)_C \ , \ (1+p)\dot{\theta}(t)_C\Big] \ , \tag{2.25}$$

where the constant p is used to control how tight the constraint are set.

Position

The positional constraints are only possible with accurate map information and local position on the map. If the local position is known with a certain accuracy then the position of the oncoming vehicle on the map is also known. This allows the position predictions to be limited to the roads available, thus further narrowing the predictions. This constraint will not be implemented in this work, but it's worth noting that such knowledge would greatly benefit such a system.

2.5 Assumptions

To conclude the description of the problem and to offer an overview of the conditions of the project, a list of assumption, that were stated in this chapter and will be valid during the rest of the thesis, is organized below.

- Two vehicle are involved in each scenario, the ego vehicle and the other vehicle.
- The ego vehicle is the vehicle equipped with the CA system.
- The other vehicle is manually driven with no active CA system capabilities.

- For simplicity and without loss of generality, the ego vehicle is assumed going straight and will not perform a backing manoeuvre.
- The sensor is an advanced frontal sensor or combination of such complemented with an internal measurement unit for the ego states.
- Measurements are provided by a generic sensor, such as a radar, are speed, heading, position and length of the other vehicle and speed, yaw rate and acceleration for the ego vehicle.
- Vehicles are operating within the linear region and within the urban speed limit of 50 $\frac{km}{h}$. This means that little or no slip of the tires.
- Perfect conditions, meaning that the sensor will have clear sight and the road conditions are good.
- Only frontal collisions of the ego vehicle are considered.

3

Path Prediction

The nature of the collision type in question is such, that often when the other vehicle has entered the ego vehicles lane, the time until the collision happens is very short . In many cases it is not enough to start braking when the other vehicle has already entered your lane but it is necessary to start braking *before* it has entered the lane. Being able to predict a collision and the duration of how long the vehicle will occupy the collision area will greatly increase the performance of any collision avoidance (CA) system.

3.1 Problem description

Given a noisy set of measurements the goal is to perform a path prediction. Obviously, a number of filtering techniques area available to perform the noise reduction and combining that with the constant turn rate and acceleration (CTRA) model should allow for adequate predictions. But the goal is to capture *all* possible future paths of a vehicle, where the most likely one is the CTRA.

Consider Figure 3.1 which illustrates how using only the CTRA model to predict the future would essentially not capture all possible paths of the vehicle. Here the ellipsoid demonstrate how a predicted probability density would look like using the different procedures.

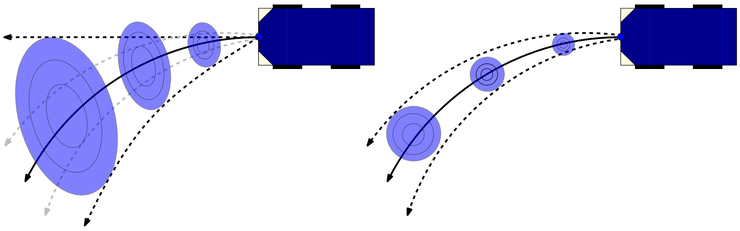


Figure 3.1: Comparison between using a CTRA model only (right) versus applying the probability of the driver changing the input (left). Here the ellipsoids represents the probability density of the predicted positions.

Notice how the predicted area of the future position gets bigger when the driver is assumed to have a defined probability of changing the turn rate and acceleration as opposed to the CTRA model only which is much smaller. The goal is therefore to provide a probability map of where the vehicles can be positioned at a given time step which can be used to evaluate a possible collision.

3.1.1 Related work

There have been a number of interesting work done in estimating the future path of vehicles. A. Eidehall and L. Petersson in [4] propose an elegant way of predicting vehicle movements based on visibility and obstacles in the environment using Monte Carlo sampling. Similarly [5] uses Markov Chains to make a probabilistic danger map of moving obstacles. Both contributions inspired the method used in this thesis.

A very similar procedure is done in [6] where the Extended Kalman filter (EKF) is used for map based path predictions.

3.1.2 Thesis contribution

As previously mentioned, a driver typically uses a constant turn rate and acceleration during manoeuvres. But to capture the possible movements of the vehicle, it is necessary to model the possible change in acceleration and turn rate. The major contribution of this chapter is the procedure of using a defined probability of a driver to change the turn rate and acceleration to predict future position of the other vehicle. This is done by using an Unscented Kalman filter (UKF) and is visualized in Figure 3.2.

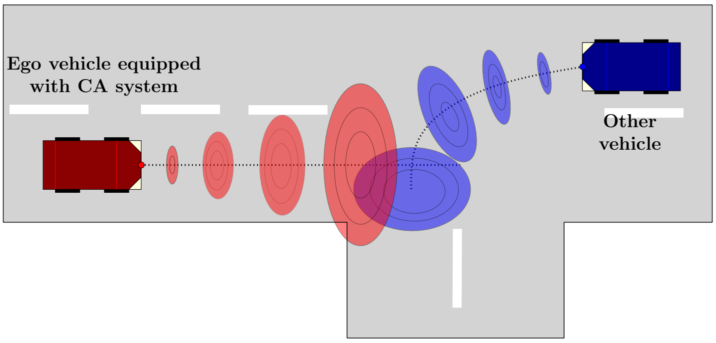


Figure 3.2: Illustration of how the predictions will look like. The probability densities (blue and red ellipsoids) represent at what points the front bumpers of the vehicles can be positioned at at each consecutive time step.

3.2 A Kalman filter-based solution

The Kalman filter is a powerful estimation tool capable of estimating present and future states, essentially filtering out noise in an optimal way and providing a good estimate on the future state of a system. The performance of the filter is of course limited to the accuracy of the model used and the characteristics of said model. In the coming sections, a brief overview of the Kalman filter procedures is provided. For more details, the reader is referred to [7]. Note that in this section, for the sake of brevity, the x and y now represent the state vector and measured states respectively.

3.2.1 Kalman filter

The linear Kalman filter is applicable to a system on the following form

$$\begin{aligned} \dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t) + W(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) + V(t) \end{aligned} \tag{3.1}$$

Where **A**,**B**,**C**,**D** are the state, input, output and feed through matrices respectively, x and u are the states and inputs specified in vector form. W and V are uncorrelated Gaussian noise vectors represented as $W \sim N(0,Q)$ and $V \sim N(0,R)$ where, respectively, Q and R are the covariance matrices of the process and measurement noise. By discretizing the matrices, Equation (3.1) can be written as

$$\begin{aligned} \dot{x}_k &= \mathbf{A}_d x_{k-1} + \mathbf{B}_d u_{k-1} + W_{k-1}, \\ y_k &= \mathbf{C}_d x_k + \mathbf{D}_d u_k + V_k, \end{aligned} \tag{3.2}$$

where the subscript d denotes a discretized version of the state space matrices. In the subsequent sections, note that $\mathbf{D}_d = 0$.

3.2.1.1 A priori estimate

Since the Kalman filter is a recursive estimator, it uses the knowledge of past measurement to estimate the expected state at time step k, given past measurement. Let the notation $(\hat{.})$ represent the estimated state vector, $(.)_{[k|k-1]}$ be the estimated states at time k given the filtered state vector at time $k-1$ and $(.)_k$ denote the time step of the different variables. Along with this state estimate, an *a priori* estimate of the covariance is made given the same information, similarly denoted by $P_{[k|k-1]}$,

$$\begin{aligned} \hat{x}_{[k|k-1]} &= \mathbf{A}_d \hat{x}_{[k-1|k-1]} + \mathbf{B}_d u_{k-1} \\ P_{[k|k-1]} &= \mathbb{E}[x_[k|k-1]|y_{1:k-1}] = \mathbf{A}_d P_{[k-1|k-1]} \mathbf{A}_d^T + Q_k, \end{aligned} \tag{3.3}$$

where knowledge of the process noise, contained in the covariance matrix Q, is used in the estimation. To avoid confusion with coming section, the use of these equations will be referred as *forecasting.*

3.2.1.2 A posteriori estimate

When a new measurement is received, the Kalman filter uses the information from the *a priori* estimate to filter out the noise of the measurement. Given the measurement z_k the goal is to give an *a posteriori* estimate of the actual state and the covariance of that estimated state. The first step is to calculate the residual error of the prediction,

$$e_k = \mathbf{C}_d \hat{x}_{[k|k-1]} - z_k, \tag{3.4}$$

which is then used to calculate the covariance in the measurement space. Given the information of the measurement variance in the covariance matrix R_k the covariance matrix S,

$$S_k = \mathbf{C}_d P_{[k|k-1]} C_d^T + R_k, \tag{3.5}$$

describes the expected variance of the states in the measurement space. This information is then used to calculate the Kalman gain. The Kalman gain is used to calculate the filtered mean of the actual state along with the estimated covariance matrix $P_{k|k}$,

$$\begin{aligned} K_k &= P_{[k|k-1[} \mathbf{C}_d^T S_k^{-1}, \\ \hat{x}_{[k|k]} &= \hat{x}_{[k|k-1]} + K_k e_k, \\ P_{k|k} &= (I - K_k \mathbf{C}_d) P_{[k|k-1]}. \end{aligned} \tag{3.6}$$

The equations described in this section will be referred as the *update* step.

3.2.1.3 Restrictions

It is uncommon that a system can be described fully with linear equations and therefore needs to be linearised. One type of Kalman filter that handles this kind of system is the EKF. There the state space matrices A_d, B_d, C_d, D_d will be linearised around the previous estimated state $\hat{x}_{[k-1|k-1]}$ at each time step.

Now, since the objective of this thesis is to predict the future trajectory of the oncoming vehicle, it is considered of high importance that the process noise is propagated as accurately as possible through the system. According to [8], the usage of EKF to evaluate the possible states of a highly non-linear system is outperformed by the UKF[9]. In [10], the performance increase on a similar vehicle model via using UKF rather than EKF is investigated and significant increase is detected.

3.2.2 Unscented Kalman filter

The main idea behind the UKF is that, by use of the unscented transform (UT), *"it is easier to approximate a probability distribution than it is to approximate an arbitrary non-linear function or transformation"* [11]. In this framework a non-additive UKF is used. The unicycle model used is subjected to a non-additive process noise and as such it is necessary to use a non-additive UKF. In [12], the steps to this method is described in more detail.

The model in this project is subject to a non-additive process noise and therefore, the steps described in the coming section are defined with that in mind. Before the filtering steps are defined, it is necessary to introduce the UT.

3.2.2.1 The Unscented Transform

As previously mentioned, when using the Kalman filter, the update equations need to be linear or linearised at each time step. The UT is a way around that problem by applying the non-linear update equation to a finite set of points, to estimate the transformed probability distribution. These finite set of points are referred to as *sigma points.*

Let $(.)^{(i)}$ reprisent the i^{th} column in a matrix. Given the state vector x with n states and it's covariance matrix P the sigma points are calculated as,

$$\begin{aligned} \rho &= \sqrt{P}, \\ \mathbf{X}^{(0)} &= x, \\ \mathbf{X}^{(i)} &= x + \rho^{(i)}\sqrt{n+\lambda}\ , \\ \mathbf{X}^{(i+n)} &= x - \rho^{(i)}\sqrt{n+\lambda}\ ,\ i = 1,..,n, \end{aligned} \tag{3.7}$$

where $\lambda = \alpha(n+\kappa) - n$ and α and κ are tunable parameters that determine the spread of the mean and are usually set to a small positive value (e.g 1e-3). The $\sqrt{P}$ is efficiently calculated by using Cholensky decomposition [13]. Before propagating the sigma points through the function, the weight coefficients, which are used for calculations of the mean and covariance, are needed. These weight coefficients are defined as,

$$\begin{aligned} \omega_0 &= \frac{\lambda}{n+\lambda}\ , \\ \Omega_0 &= \frac{\lambda}{n+\lambda} + (\kappa - \alpha^2 + \beta)\ , \\ \omega_i = \Omega_i &= \frac{1}{2(n+\lambda)}\ ,\ i = 1,...,2n, \end{aligned} \tag{3.8}$$

where ω and Ω denote the weighting coefficients for the mean and variance respectively. Now let $\mathbf{X}_f^{(i)}$ represent the propagated sigma point from column i through function f such that $\mathbf{X}_f^{(i)} = f(\mathbf{X}^{(i)})$ and $(\hat{.})$ denotes the estimation of the corresponding variable. Then the transformed mean and covariance are calculated as,

$$\begin{aligned}
\hat{x} &= \mathbb{E}[\mathbf{X}_f^{(i)}] = \sum_{i=0}^{2n} \omega_i \mathbf{X}_f^{(i)}, \\
\hat{P} &= \mathbb{E}[(\mathbf{X}_f^{(i)} - \hat{x})(\mathbf{X}_f^{(i)} - \hat{x})^T] = \sum_{i=0}^{2n} \Omega_i (\mathbf{X}_f^{(i)} - \hat{x})(\mathbf{X}_f^{(i)} - \hat{x})^T, \\
C &= \mathbb{E}[(\mathbf{X}_f^{(i)} - m)(\mathbf{X}^{(i)} - x)^T] = \sum_{i=0}^{2n} \Omega_i (\mathbf{X}_f^{(i)} - \hat{x})(\mathbf{X}^{(i)} - x)^T,
\end{aligned} \tag{3.9}$$

where $\hat{x}$, $\hat{P}$, C are the transformed mean, covariance and cross covariance respectively. For clarity, the steps described in this section are concluded with the following equation,

$$\begin{bmatrix} \hat{x}, & \hat{P}, & C \end{bmatrix} = Unscented(x{,}P{,}f{,}Q). \tag{3.10}$$

3.2.2.2 A priori estimate

As stated before, the model is subjected to non-additive process noise. To handle this, the covariance matrix $P_{[k-1|k-1]}$ is augmented with the process noise and the estimated mean state vector $\hat{x}_{[k-1|k-1]}$ is augmented with zeros, before the UT is performed. Let $(.)'$ denote the augmented version of the corresponding variable, such that,

$$\begin{aligned}
P'_{[k-1|k-1]} &= \begin{bmatrix} P_{[k-1|k-1]} & 0 \\ 0 & Q \end{bmatrix}, \\
\hat{x}'_{[k-1|k-1]} &= \begin{bmatrix} \hat{x}_{[k-1|k-1]} \\ 0 \end{bmatrix},
\end{aligned} \tag{3.11}$$

where n is the number of states and m is the number of process noise applied, the size of the matrices are described as,

$$\begin{aligned}
P_{[k-1|k-1]} \quad &\in \mathbb{R}^{n \times n}, \\
Q \quad &\in \mathbb{R}^{m \times m}, \\
P'_{[k-1|k-1]} \quad &\in \mathbb{R}^{(n+m) \times (n+m)}, \\
\hat{x}'_{[k-1|k-1]} \quad &\in \mathbb{R}^{1 \times (n+m)}.
\end{aligned} \tag{3.12}$$

These augmented state and covariance matrices are then transformed with the UT. Using the non-linear update function f gives,

$$\begin{bmatrix} \hat{x}_{[k|k-1]}, & \hat{P}_{[k|k-1]} \end{bmatrix} = Unscented(\hat{x}'_{[k-1|k-1]}{,}P'_{[k-1|k-1]}{,}f{,}Q). \tag{3.13}$$

The steps taken from equations (3.11) through (3.13) are concluded in the following equation,

$$\begin{bmatrix} \hat{x}_{[k|k-1]}, & \hat{P}_{[k|k-1]} \end{bmatrix} = Forcast(P_{[k-1|k-1]},\tilde{x}_{[k-1|k-1]},f,Q). \tag{3.14}$$

3.2.2.3 A posteriori estimate

To get the *a posteriori* estimate, the UT is again used to perform the estimation, but this time into the measurement space. Using the *a priori* estimates $\hat{x}_{[k|k-1]}$ and $P_{[k|k-1]}$, with a measurement equation H, gives,

$$\begin{bmatrix} y_k, & S_k, & C_k \end{bmatrix} = Unscented(\hat{x}_{[k|k-1]},P_{[k|k-1]},H,0). \tag{3.15}$$

The residual error is then calculated as before,

$$e_k = y_k - z_k. \tag{3.16}$$

and the Kalman gain, filtered state mean and covariance matrix are,

$$\begin{aligned} L_k &= S_k + R, \\ K_k &= C_k L_k^{-1}, \\ \hat{x}_{[k|k]} &= \hat{x}_{[k|k-1]} + K_k e_k, \\ P_{[k|k]} &= P_{[k|k-1]} - K_k L_k K_k^T, \end{aligned} \tag{3.17}$$

where R is the measurement noise covariance matrix. As before, the steps performed in equations (3.15) through (3.17) are concluded in the following equation,

$$\begin{bmatrix} \hat{x}_{[k|k]}, & P_{[k|k]} \end{bmatrix} = Update(P_{[k|k-1]},\hat{x}_{[k|k-1]},H,R). \tag{3.18}$$

3.2.2.4 Unscented Kalman filter constraints

There are many methods available to perform input and state constraints, where the most popular are investigated in [14]. One of the simplest methods of state constraints is by projecting the sigma points, generated from Equation (3.7), that lie beyond the bounds onto the boundary. These constrained sigma points are then propagated through the function and constrained again.

Assume that the vectors a and b contain the boundary of all states such that the state vector x should fulfil $a \leq x \leq b$, where the constraints at each moment in time are calculated as explained in Section 2.4.6. The procedure is illustrated in Algorithm 3.1 where n is the number of states and $\mathbf{X}$ are the sigma points generated from the UT.

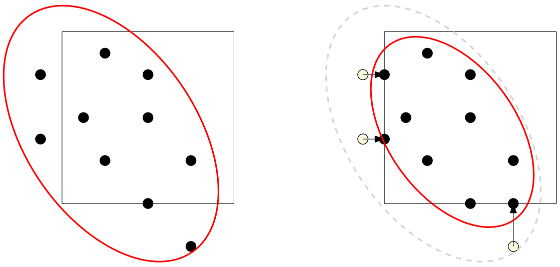


Figure 3.3: Illustration of how the sigma points lying outside of the constraints (left figure) are moved onto the boundary. The red circle denotes the distribution calculated before (left) and after (right) the constraining. Notice how the distribution becomes more restricted after constraining is done

Algorithm 3.1: Constrained Sigma Points

```
Input  : X (Sigma point matrix) and $\hat{x}$ (Current state mean)
Output: X
Initialize
$[a,b]$ = Constraints based on the current state mean vector $\hat{x}$
for $j < 1 : n$ do
    Check for constraint violation
    $Lc = a(j) \geq \mathbf{X}(j,1 : 2n + 1)$
    $Uc = b(j) \leq \mathbf{X}(j,1 : 2n + 1)$
    Projecting violations onto boundary
    $\mathbf{X}(j,Lc) = a(j)$
    $\mathbf{X}(j,Uc) = b(j)$
end
```

The constraints are only employed during the path prediction, as it is of importance to have the predicted states within the physical limits of the vehicle. The constrained UT is explained in Algorithm 3.2, which will be referred to as the Constrained Unscented Transform (CUT).

Algorithm 3.2: Constrained UT procedure,, referred to as CUT

```
Input  : $x$ , $P$ , $f$
Output: $\hat{x}$ , $\hat{P}$ , $C$
Generate sigma points using Equation (3.7)
Run Algorithm 3.1
Propagate through desired function
Run Algorithm 3.1
Use Equation (3.9)
```

3.2.2.5 Filtering procedure

To make the coming section compact, the following algorithm is introduced to represent a filtering procedure, namely

Algorithm 3.3: Filtering procedure, referenced as **Filter**

```
  Input  : $z_k$ , $\hat{x}_{[k-1|k-1]}$, $P_{[k-1|k-1]}$, $f$ , $H$, $R$ ,$Q$
  Output: $\hat{x}_{[k|k]}$ , $P_{[k|k]}$
  $[\hat{x}_{[k|k-1]}, P_{[k|k-1]}] = Forcast(P_{[k-1|k-1]}, \hat{x}_{[k-1|k-1]}, f, Q)$
  $[\hat{x}_{[k|k]}, P_{[k|k]}] = Update(\hat{P}_{[k|k-1]}, \hat{x}_{[k|k-1]}, H, R)$
```

3.3 Prediction procedure

To conclude the different sections, the filtering and prediction procedure are combined into one complete algorithm. Recalling from Section 2.4 where χ, γ represented the state vector and measured states, $(\bar{.})$ denotes that the parameter in question relates to the other vehicle, $(\hat{.})$ is used to denote a predicted state. The derived models, for the ego and other vehicle respectively were,

Filtering

$$\begin{aligned} \chi_{k+1} &= f(\chi_k, u_k), & \bar{\chi}_{k+1} &= \bar{f}(\bar{\chi}_k, \bar{u}_k), \\ \gamma_k &= H\chi_k + W, & \bar{\gamma}_k &= \bar{H}(\bar{\chi}_k) + \bar{W}, \end{aligned}$$

Prediction

$$\hat{\chi}_j = \hat{f}(\tilde{\chi}_k, u_k, j)., \qquad \hat{\bar{\chi}}_j = \hat{\bar{f}}(\bar{\chi}_k, \bar{u}_k, j),$$

and the measurement noise covariance matrices are,

$$\begin{aligned} R &= diag(W), \\ \bar{R} &= diag(\bar{W}). \end{aligned} \tag{3.19}$$

Since the ego vehicle filter model does not include the positional change, and is in fact augmented with zeros as explained in the previous chapter, the filtered state covariance matrix of the ego vehicle, $P_{[k|k]}$ needs to be augmented as well. The Kalman filter procedure requires that the covariance matrices should be positive semi-definite. Consequently the covariance matrix can not be augmented with zero variance, which should actually be the case, as the position of the ego vehicle is defined as the origin point. The augmented covariance matrix is therefore,

$$P'_{[k|k]} = \begin{bmatrix} P_{k|k} & 0 & 0 \\ 0 & \sigma_x^2 & 0 \\ 0 & 0 & \sigma_y^2 \end{bmatrix}, \tag{3.20}$$

where the variance of the position is defined to be a small value, that is,

$$\sigma_x = 10^{-3} \quad \sigma_y = 10^{-3} \ . \tag{3.21}$$

Then the total filtering procedure is described in Algorithm 3.4 where Q, Q_o and Q_p are the process noise matrices that describe the inputs of the vehicle models.

Algorithm 3.4: Complete filtering and prediction algorithm

```
Input  : Ego sensor data: $z$, Other sensor data: $\bar{z}$, Detected
Output: $\hat{P}_j$ , $\hat{\chi}_j$ , $\hat{\bar{P}}_j$ , $\hat{\bar{\chi}}_j$
Initialize: $P_{[1|0]} = \mathbb{I}, \quad \chi_{[1|0]} = z, \quad \bar{P}_{[1|0]} = \mathbb{I}, \quad \bar{\chi}_{[1|0]} = z, \quad k = 1$
while True do
    if Detected = True & Detected_prev = False then
        Initialize: $\bar{P} = \mathbb{I}, \quad \bar{\chi}_{[k-1|k-1]} = \bar{z}$
    if Detected = True then
        Perform filtering on ego states, using Algorithm 3.3
        $[P_{[k|k]} \ , \chi_{[k|k]}] = \text{Filter}(z \ , \chi_{[k-1|k-1]}, P_{[k-1|k-1]}, f \ , H, R \ ,Q)$
        Perform filtering on other states, using Algorithm 3.3
        $[\bar{P}_{[k|k]} \ , \bar{\chi}_{[k|k]}] = \text{Filter}(\bar{z} \ , \bar{\chi}_{[k-1|k-1]}, \bar{P}_{[k-1|k-1]}, \bar{f} \ , \bar{H}, \bar{R} \ ,\bar{Q})$
        for $j = 1 : t_{pred}$ do
            Forecast step performed using CUT
            $[\hat{P}_j \ , \hat{\chi}_j] = \text{Forecast}(\tilde{P}_{[k|k]} \ , \hat{\bar{x}}_{[k|k]}, f_p \ ,Q_p)$
            $[\hat{\bar{P}}_j \ , \hat{\bar{\chi}}_j] = \text{Forecast}(\bar{P}_{[k|k]} \ , \bar{\chi}_{[k|k]}, \bar{f}_p \ ,Q_p)$
        end
    else
        Perform filtering on ego states, using Algorithm 3.3
        $[P_{[k|k]} \ , \chi_{[k|k]}] = \text{Filter}(z \ , \chi_{[k-1|k-1]}, P_{[k-1|k-1]}, f \ , H, R \ ,Q)$
    end
    Detected_prev = Detected
    k = k + 1
end
```

This procedure provides state predictions from which information about the predicted positions of each vehicle can be acquired, similar to Figure 3.2. However, it does not evaluate the probability of a collision given these paths and therefore a collision detection is needed, which is introduced in the next chapter.

4

Collision detection

The detection of collision resulting from conflicting paths is a key part in this proposed collision avoidance (CA) system. The Unscented Kalman filter (UKF) provides the information needed to make an estimate of the collision, but careful consideration is needed so that the number of false alarms is kept to a minimum. This is done by checking how probable it is, that the vehicles are positioned at the same area at the same time by using their predicted positions in Cartesian coordinate frame.

4.1 Problem description

The path prediction provides information about the estimated future positions of the two vehicles over the prediction horizon. Given these estimated positions, which are described as a Bivariate Normal Distribution (BVD), it is necessary to develop an efficient way to evaluate if those two independent probability densities indicate a possible collision. Upon detecting a collision, three parameters are estimated. Consider Figure 4.1 where an overview of a frontal collision is shown. The parameters of interest, are the time to collision (TTC), collision time interval (ΔTTC) and distance to collision (DTC), denoted as t_c, Δt_c and d_c respectively. These parameters are later used to determine the correct collision avoidance timing. In this thesis, the only collision scenario considered, is where the other vehicle takes a left turn across the ego vehicle's path and as such, the collision definition is defined accordingly. For other types of scenarios the collision definition might need to be revised.

4.1.1 Related work

A lot of research has been done on collision detection methods. Kim et al. [15] assumes a Vehicle-To-Vehicle (V2V) communication, where vehicles transmit their future paths and can warn the driver if the paths intersect. In a similar way, Wang et al. [16] predicts the future paths of vehicle and uses V2V to transmit the information. From there, the

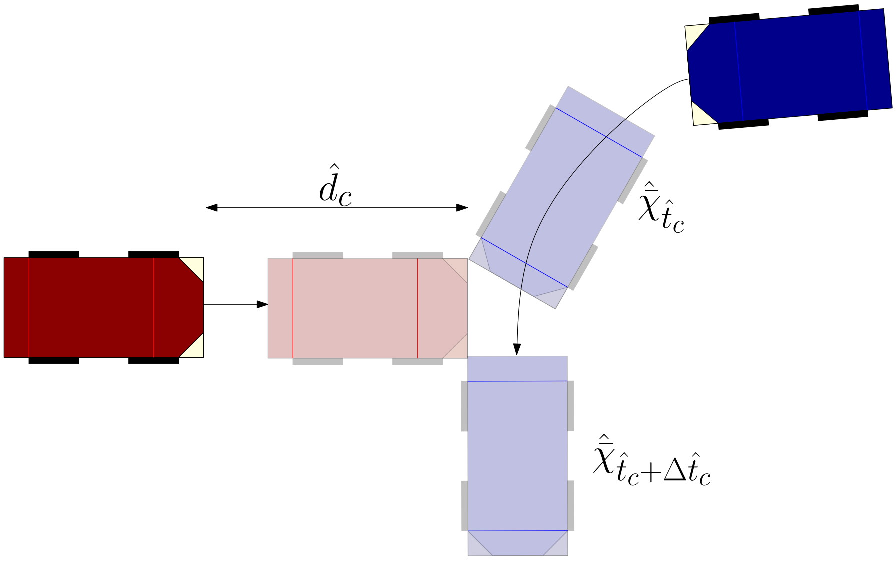

$\hat{d}_c$
$\hat{\bar{\chi}}_{\hat{t}_c}$
$\hat{\bar{\chi}}_{\hat{t}_c+\Delta\hat{t}_c}$

Figure 4.1: A scenario illustrating the important parameters for the proposed collision avoidance systems. The distance to collision, $\hat{d}_c$, time to collision, $\hat{t}_c$ and collision time interval, $\Delta\hat{t}_c$. To be able to successfully avoid a collision, the ego vehicle needs information about these three parameters to know when it is necessary to start braking.

intersection points are defined and the time to collision is calculated based on current speed. They propose using circles or boxes as polygons to calculate the collision time. In both works, the probability of the collision is not defined. Weidl et al. [17] uses map knowledge and V2V communication to identify conflicting paths and evaluates the probability from these conflicts. [18] proposes using polygons to represent the vehicles, and check for their potential intersection which could be considered a collision. The calculations of the intersecting polygons are done using lookup tables and are only done when the polygons have a possibility of intersecting to save computation power.

4.1.2 Thesis contribution

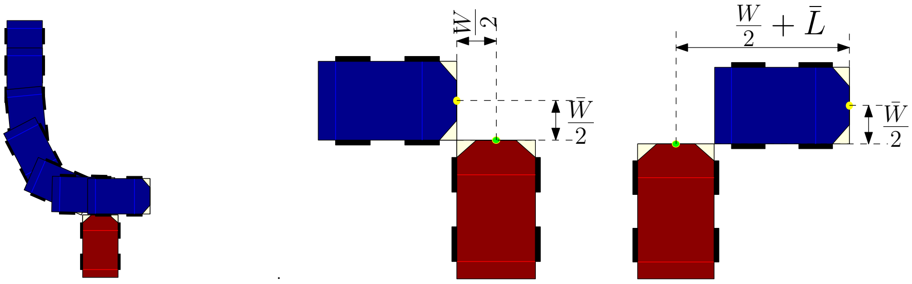
The main contribution of this thesis is an efficient way of evaluating a possible collision without the need of introducing polygons to represent vehicles and check for intersection, or similarly using lookup tables. This is done by defining what vector combination can lead to a collision and evaluating the probability of such a vector to exist.

4.2 Collision definition

To be able to evaluate the probability of a collision, the space where it can occur needs to be defined. As stated in Section 2.2, only frontal collisions when the front of ego vehicle collides with the side of the other vehicle are considered. There are essentially two areas that are of importance. One that defines where a collision will occur and another which defines when the collision area is unoccupied. For computation purposes, the collision areas are defined by convex squares.

4.2.1 Collision area

To detect a frontal collision, it is assumed that it is possible to define a vector space, where it is likely to happen. By identifying the extreme points of the areas, for example the front corner of both vehicles colliding and the front corner of the ego vehicle colliding with the back corner of the other vehicle, the bounds can be identified. Figure 4.2 illustrates this concept, where the other vehicle performs a left turn across path (LTAP) and the two extreme collision points are identified. Given these collision definitions above it is possible to define the collision space, shown in Figure 4.3. This collision box does not capture all types of turns, and therefore a second collision box, which is meant to capture an aggressive turn scenario, is defined. This is illustrated in Figure 4.4.



(a) A vehicle making a typical LTAP

(b) The first point of this manoeuvre that would lead to a frontal collision.

(c) The last point of the manoeuvre where frontal collision can happen.

Figure 4.2: Illustration of how the definition of a collision can be defined by a vector space, by first identifying the extreme points of the a particular collision. These points are used to mathematically express this area.

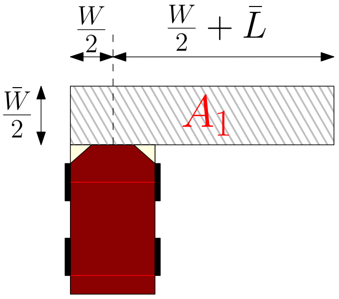


Figure 4.3: Illustration of how the collision box is defined based on the extreme points from Figure 4.2.

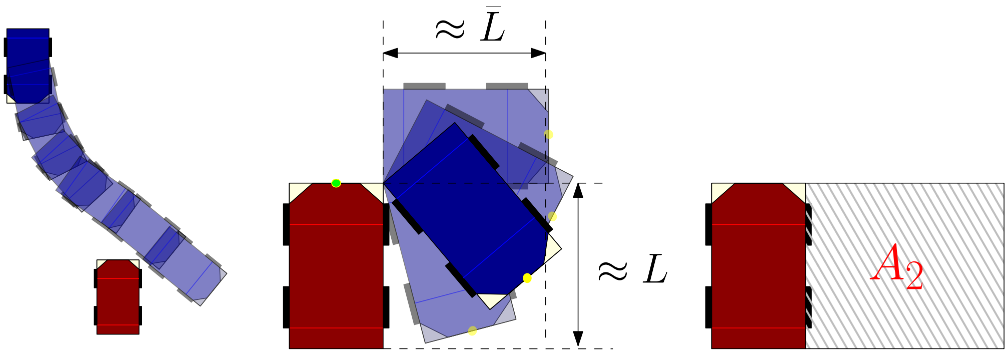


Figure 4.4: Same procedure as done through figures 4.2 and 4.3.

Considering the position of the ego vehicle's front bumper as the origin point, the two convex vector spaces, A_1 and A_2, are defined as,

$$\begin{aligned} \begin{bmatrix} -\frac{W}{2} \\ 0 \end{bmatrix} &\leq A_1 \leq \begin{bmatrix} \frac{W}{2} + \bar{L} \\ \frac{\bar{W}}{2} \end{bmatrix}, \\ \begin{bmatrix} \frac{W}{2} \\ -\bar{L} \end{bmatrix} &\leq A_2 \leq \begin{bmatrix} \frac{W}{2} + \bar{L} \\ 0 \end{bmatrix}, \end{aligned} \tag{4.1}$$

containing all possible vectors that will result in a frontal collision. The parameters L and W denote the length and size of the ego vehicle while $\bar{L}$ and $\bar{W}$ are the length and width of the other vehicle. Note, that although the size of the other vehicle is assumed to be known, it is possible, without loss of generality, to redefine the parameters of these collision boxes based on real time measurement of the other vehicle's size.

4.2.1.1 Conservativeness

The collision spaces defined above are not completely accurate. There are scenarios where they will detect a frontal collision in cases where there should not be one. Consider Figure 4.5 where the other vehicle crashes into the side of the ego vehicle. Clearly, this is not a frontal collision, as the system is supposed to detect, but is none the less agreeing

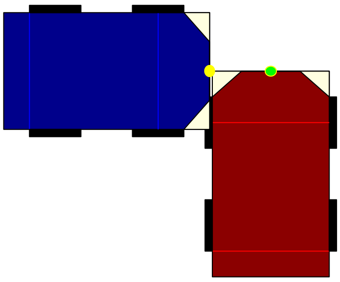


Figure 4.5: Illustration of a possible „false" detection using collision box A_1.

with collision space A_1. But in this case a braking intervention is a plausible collision avoidance solution and as such, this collision box is still valid. Some conservativeness is also introduced in A_2 as shown in Figure 4.6. If one considers that drivers have a comfort zone, that is, the distance to a passing vehicle that a driver is comfortable with, it is possible to argue that this conservativeness includes this plausible comfort zone.

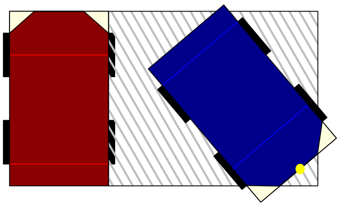


Figure 4.6: Illustration of conservativity of collision box A_2.

4.2.2 Collision free space

As mentioned before, it is not only necessary to evaluate if a collision will occur, it is also crucial to estimate the *duration* of the collision. This is done by checking when the back bumper of the other vehicle, lies inside an area, which does not lead to a collision, as shown in Figure 4.7. Note, that this area is not representing the same situation as area A_2. Here, the position point in question is the back bumper of the other vehicle which the ego vehicle cannot crash into when it has reached this area. In a similar way as before, area A_3 is defined as,

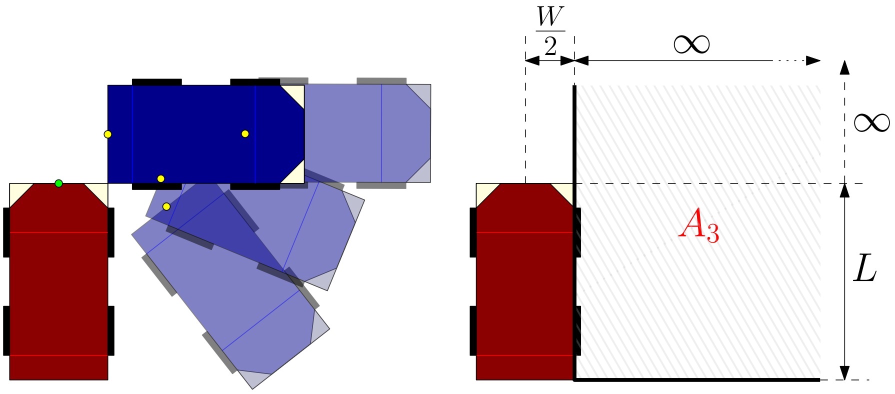


Figure 4.7: When the back bumper of the other vehicle is inside vector space A_3, the *collision area* is unoccupied.

$$\begin{bmatrix} \frac{W}{2} \\ -L \end{bmatrix} \leq A_3 \leq \begin{bmatrix} \infty \\ \infty \end{bmatrix}. \tag{4.2}$$

4.3 Deterministic detection

The variable used for detection of a collision is the vector between the two vehicles. For two Cartesian points N and M, the vector from N to M is calculated as,

$$\overrightarrow{NM} = [M_x - N_x, M_y - N_y]. \tag{4.3}$$

Where the subscripts x and y denote the Cartesian coordinates of points N and M. Now let $\mathcal{E}_j$ and $\mathcal{F}_j$ denote the position of the front bumper of the ego vehicle and other vehicle respectively, at the predicted time step j. The vector between the two, $\overrightarrow{\mathcal{EF}}_j$, is calculated using the previous equation. Then the existence of a collision can be verified by checking the following condition,

$$P_{Collision}(\overrightarrow{\mathcal{EF}}_j) = \begin{cases} 1, & \text{if } \overrightarrow{\mathcal{EF}}_j \in A_1 \cup A_2, \\ 0, & \text{otherwise,} \end{cases} \tag{4.4}$$

where a collision is detected at time j if $P_{Collision}(\overrightarrow{\mathcal{EF}}_j)$ is 1. Similarly, by calculating the vector $\overrightarrow{\mathcal{ER}}_j$ where $\mathcal{R}$ now denotes the position of the rear bumper of the other vehicle, it is possible to identify when the collision area is free by verifying

$$P_{Free}(\overrightarrow{\mathcal{ER}}_j) = \begin{cases} 1, & \text{if } \overrightarrow{\mathcal{ER}}_j \in A_3, \\ 0, & \text{otherwise.} \end{cases} \tag{4.5}$$

Now assume that a collision is detected at the future time step a, then, and only then, is it necessary to identify when that collision area is free at some time b by using Equation (4.5) and obviously, $b > a$. Note, these equations are only valid for the deterministic case, where the exact future position of both vehicles are known. Since the positions are only an estimation, the probability that the vector $\overrightarrow{\mathcal{EF}}_j$ belongs to some vector space needs to be calculated. A solution to this problem will be given in the next section.

4.4 Probabilistic detection

In this section, a method for collision detection is presented. However, it is first necessary to introduce the tools needed for the calculations.

4.4.1 Bivariate normal distribution

A BVD is a special case of a multivariate distribution with dimension of two. Consider a random Cartesian point, whose position is described as,

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim N\left(\overbrace{\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}}^{\boldsymbol{\mu}}, \overbrace{\begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}}^{\boldsymbol{\Sigma}}\right),$$

where μ_1 and μ_2 denote the mean of x and y respectively. Furthermore σ_1 and σ_2 denote the variance of said position and σ_{12} denote the cross covariance, between these two variables. To calculate the probability density, at some point $\rho = [x_1,y_1]$, the following equation applies,

$$\mathcal{D}(\rho) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} e^{(Q(\rho)/2)}, \tag{4.6}$$

where

$$Q(v) = (v - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (v - \boldsymbol{\mu}), \tag{4.7}$$

and the variance and cross covariance full fills the following condition,

$$\sigma_i^2 > 0 \text{ where } i \in [1,2], \tag{4.8}$$

$$|\sigma_{12}| < \sigma_1 \sigma_2. \tag{4.9}$$

Now the goal is to find the vector between two BVD's, which, according to Equation (4.3), results in a linear combination of the two distributions.

4.4.2 Linear combinations of bivariate distributions

Consider the two bivariate, normally distributed random variables X_1 and X_2 given by $N \sim (\mu_1,\Sigma_1)$ and $N \sim (\mu_2,\Sigma_2)$ respectively. The linear combination of these two distribution is given by

$$aX_1 + bX_2 \sim N(a\mu_1 + b\mu_2 \ , |a|\Sigma_1 + |b|\Sigma_2 + 2ab\Sigma_{12}), \tag{4.10}$$

where a and b can be any combinations of numbers and Σ_{12} denotes the cross correlation between X_1 and X_2.

Now, given the estimated positions of the ego vehicle and other vehicle, namely the bivariate normally distributed variables $\mathcal{E} \sim (\mu_{\mathcal{E}},\Sigma_{\mathcal{E}})$, $\mathcal{F} \sim (\mu_{\mathcal{F}},\Sigma_{\mathcal{F}})$ and $\mathcal{R} \sim (\mu_{\mathcal{R}},\Sigma_{\mathcal{R}})$, then equations (4.3), (4.6) and (4.10) can be used to define the following distributions,

$$\begin{aligned} \mathcal{P}_{\mathcal{F}_j}(x,y) &\sim N(a\mu_{\mathcal{F}_j} - b\mu_{\mathcal{E}_j} \ ,\Sigma_{\mathcal{F}_j} + \Sigma_{\mathcal{E}_j}), \\ \mathcal{P}_{\mathcal{R}_j}(x,y) &\sim N(a\mu_{\mathcal{R}_j} - b\mu_{\mathcal{E}_j} \ ,\Sigma_{\mathcal{R}_j} + \Sigma_{\mathcal{E}_j}), \end{aligned} \tag{4.11}$$

where clearly $\Sigma_{12} = \mathbf{0}$, as the positions of the two vehicles positions are uncorrelated.

4.4.3 Joint Cumulative Distribution

The Joint Cumulative Distribution (JCD) equations are given in [19] providing an efficient way to compute the probability density over the area of interest. Using the JCD function, Equation (4.4) can be transformed as,

$$\mathcal{P}_{Collision_j} = \int_{A_1} \mathcal{P}_{\mathcal{F}_j}(x,y) \mathrm{d}A + \int_{A_2} \mathcal{P}_{\mathcal{F}_j}(x,y) \mathrm{d}A, \tag{4.12}$$

where the bounds of the integrals are defined by the collision spaces derived in Section 4.2. And in the same way, to check the probability of the collision area being free, the following equation is used,

$$\mathcal{P}_{Free_j} = \int_{A_3} \mathcal{P}_{\mathcal{O}_j}(x,y)\mathrm{d}A. \tag{4.13}$$

With these equations, it is possible to estimate the parameters $\hat{t}_c$, $\Delta\hat{t}_c$ and $\hat{d}_c$, which will be described in the next section.

4.5 Collision detection procedure

By combining the predicted positions, calculated from the UKF, with equations (4.12) and (4.13) from the previous section, it is possible to check for a collision, and upon detection, calculate the characteristics of that collision. To estimate the parameters of the potential collision, the position predictions of the two vehicles from Section 3.3 are needed, as well as the predicted speed and distance travelled of the ego vehicle.

4.5.1 Predicted positions

Consider Algorithm 3.4 where, at each filter update, the algorithm returns the predicted state vectors of both vehicles, along with the corresponding covariance. Given the following covariance matrix of the other vehicle and its predicted state mean from prediction step j,

$$\hat{\hat{P}}_j = \begin{bmatrix} & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \ddots & & & & \\ \vdots & & \sigma^2_{x^f_j} & \sigma_{x^f_j,y^f_j} & \sigma_{x^f_j,x^r_j} & \sigma_{x^f_j,y^r_j} \\ \vdots & & \sigma_{x^f_j,y^f_j} & \sigma^2_{y^f_j} & \sigma_{y^f_j,x^r_j} & \sigma_{y^f_j,y^r_j} \\ \vdots & & \sigma_{x^f_j,x^r_j} & \sigma_{y^f_j,x^r_j} & \sigma^2_{x^r_j} & \sigma_{x^B_j,y^r_j} \\ \vdots & & \sigma_{x^f_j,y^r_j} & \sigma_{y^f_j,y^r_j} & \sigma_{x^r_j,y^r_j} & \sigma^2_{y^r_j} \end{bmatrix}, \quad \hat{\hat{\chi}}_j = \begin{bmatrix} \vdots \\ \vdots \\ x^f_j \\ y^f_j \\ x^r_j \\ y^r_j \end{bmatrix},$$

where the rest of the states are omitted for clarity. From here the defined probability density of the front and rear bumper, represented with the green and yellow cells respectively, are fetched to define the predicted position of the other vehicle, namely,

$$\mathcal{F}_j \sim N\left(\begin{bmatrix} x^f_j \\ y^f_j \end{bmatrix}, \begin{bmatrix} \sigma^2_{x^f_j} & \sigma_{x^f_j,y^f_j} \\ \sigma_{x^f_j,y^f_j} & \sigma^2_{y^f_j} \end{bmatrix}\right),$$

$$\mathcal{R}_j \sim N\left(\begin{bmatrix} x^r_j \\ y^r_j \end{bmatrix}, \begin{bmatrix} \sigma^2_{x^r_j} & \sigma_{x^r_j,y^r_j} \\ \sigma_{x^r_j,y^r_j} & \sigma^2_{y^r_j} \end{bmatrix}\right).$$

In a similar manner, the predicted position of the ego vehicle is defined as

$$\mathcal{E}_j \sim N\left(\begin{bmatrix} x_j \\ y_j \end{bmatrix}, \begin{bmatrix} \sigma_{x_j}^2 & \sigma_{x_j,y_j} \\ \sigma_{x_j,y_j} & \sigma_{y_j}^2 \end{bmatrix}\right).$$

These distribution provide the means to estimate if a collision is plausible, and if so, how long the danger will last.

4.5.1.1 Time of collision

By inputting the distributions from the previous section, into equations 4.12 and 4.13, the TTC and **Δ**TTC can be estimated. By inputting the estimates, $\mathcal{F}_j$, $\mathcal{R}_j$ and $\mathcal{E}_j$ into said equations, gives the following two vectors,

$$\begin{aligned} \boldsymbol{\mathcal{P}}_{Collision} &= \left[\mathcal{P}_{Collision_1} \quad ,\dots, \quad \mathcal{P}_{Collision_N}\right], \\ \boldsymbol{\mathcal{P}}_{Free} &= \left[\mathcal{P}_{Free_1} \quad ,\dots, \quad \mathcal{P}_{Free_N}\right], \end{aligned} \tag{4.14}$$

where N is the number of steps in the prediction horizon t_{pred}. Figure 4.8 shows an example of a probability evolution for the two vectors, when plotted with respect to prediction horizon. When $\boldsymbol{\mathcal{P}}_{Collision}$ violates the threshold, $\mathcal{T}_1$, a collision is detected at time $\hat{t}_c$ where the maximum of $\boldsymbol{\mathcal{P}}_{Collision}$ lies. Indeed it is unnecessary to calculate $\boldsymbol{\mathcal{P}}_{Free}$ unless $\boldsymbol{\mathcal{P}}_{Collision}$ breaks the threshold. Upon such an event, the $\Delta\hat{t}_c$ is the time difference between the two peaks.

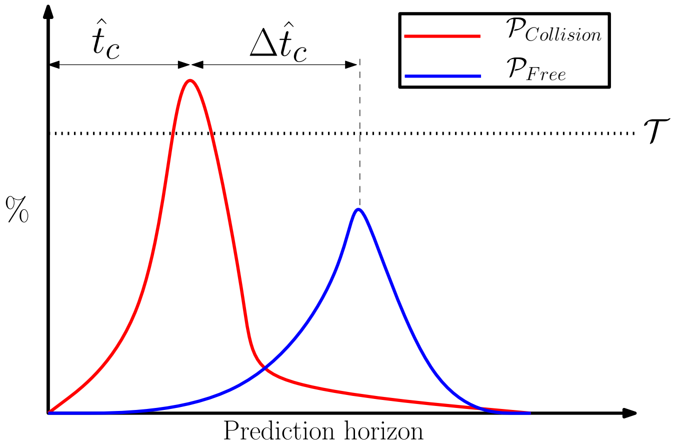


Figure 4.8: Example of a predicted collision, where the detection threshold is set at $\mathcal{T}_1$.

4.5.1.2 Speed and distance travelled

The predicted speed of the ego vehicle, v_j at future time step j, is fetched in a similar manner as in Section 4.5.1, namely,

$$v_j \sim N(\mu_{v_j}, \sigma_{v_j}^2),$$

where μ_{v_j} and $\sigma^2_{v_j}$ denote the predicted mean and variance of the speed, at future time j.

To calculate distance travelled $\hat{d}_j$ and the variance $\sigma_{\hat{d}_j}$, the forecast procedure from the linear Kalman filter is used. The first step is to augment the state vector and covariance matrices, as they do not include a state for distance travelled. Let $(\tilde{\cdot})$ denote an augmented version of said variable, then,

$$\begin{aligned} \hat{\tilde{\chi}}_j &= \begin{bmatrix} \hat{\chi}_k & , & \hat{d}_k \end{bmatrix}^T, \\ \hat{\tilde{P}}_j &= \begin{bmatrix} \hat{P}_k & 0 \\ 0 & \sigma^2_{\hat{d}_k} \end{bmatrix}, \end{aligned} \tag{4.15}$$

where $\hat{d}_k$ and $\sigma^2_{\hat{d}_k}$ are initialized as zero and $\hat{P}_k$ and $\hat{\chi}_k$ are the current estimated covariance and states of the ego vehicle. Using the following update function,

$$F_{up} = \begin{bmatrix} \frac{(j\Delta T)^2}{2} & j\Delta T & 0 & 0 & 1 \end{bmatrix}, \tag{4.16}$$

$\hat{d}_{j+1}$ is defined as,

$$\hat{d}_{j+1} \sim \mathbb{N}(F_{up}\hat{\tilde{\chi}}_j \, , \sigma^2_{d_{j+1}}), \tag{4.17}$$

and $\sigma^2_{\hat{d}_{j+1}}$ is calculated as,

$$\sigma^2_{\hat{d}_{j+1}} = F_{up}^T \hat{\tilde{\chi}}_j F_{up}. \tag{4.18}$$

These steps are indeed the same as forecasting step in a linear Kalman filter done iteratively. The distance state was excluded from the original prediction model as the filter could become unstable when the distance state needed to be reset in conjunction with employing the Constrained Unscented Transform (CUT).

4.5.2 Detection algorithm

To clarify further the procedure, assume that $\mathcal{F}_j$, $\mathcal{R}_j$, $\mathcal{E}_j$ and $\hat{d}_j$ are vectors that contain the information defined above over the whole prediction horizon t_{pred}, including the current time, such that $j = [0,1,\ldots,t_{pred}]$, then the procedure can be described with the following algorithm

Algorithm 4.1: Collision detection procedure.

```
Input  : $v$ , $\hat{d}_j$ , $\mathcal{E}_j$ , $\mathcal{F}_j$ , $\mathcal{R}_j$
Output: $\hat{t}_c$, $\hat{d}_{c,t_r+1}$, $\hat{v}_{t_r+1}$, $\Delta\hat{t}_c$

for $j < 1 : t_{pred}$ do
    Run $\mathcal{E}_j$ and $\mathcal{F}_j$ through Equation 4.12
    Collect collision probability vector, $\boldsymbol{\mathcal{P}}_{Collision}$

    Run $\mathcal{E}_j$ and $\mathcal{R}_j$ through Equation 4.12
    Collect collision area free probability vector, $\boldsymbol{\mathcal{P}}_{Free}$
end
Go through the collected data
if $any(\boldsymbol{\mathcal{P}}_{Collision}) > \mathcal{T}_1$ then
    Identify at what integer the maximums lie
    $j_{Coll} = find(max(\boldsymbol{\mathcal{P}}_{Collision}))$
    $j_{Free} = find(max(\boldsymbol{\mathcal{P}}_{Free}))$
    if $j_{Free} \leq j_{Coll}$ then
        $j_{Free} = j_{Coll} + 1$

    Define outputs
    $\hat{t}_c = j_{Coll}\Delta T$
    $\Delta\hat{t}_c = (j_{Free} - j_{Coll})\Delta T$
    $\hat{d}_{c,tr+1} \sim N(d_{t_r+1} - d_{j_{coll}}, \sigma^2_{d_{t_r+1}} + \sigma^2_{d_{j_{coll}}})$
    $\hat{v}_{t_r+1} \sim N(v_{t_r+1}, \sigma^2_{v_{t_r+1}})$
else
    $Danger = False$
end
```

Here the $\hat{d}_{c,t_r+1}$ contains the predicted distance to collision at the future time step $t_r + 1$ and v_{t_r+1} is similarly the predicted speed. The t_r denotes the reaction time of the ego vehicle. The reason that the output of the collision parameters are given w.r.t the time step $t_r + 1$ is motivated in Chapter 5.

$\mathcal{T}_1$ is a threshold and a tunable parameter. It should generally be set to a high value to avoid triggering a detection at non-threatening situations. Notice that, if the j_{Free} is less or equal to the j_{Coll}, it is changed such that it is at least one time step larger. This is necessary since sometimes, during early predictions, the j_{Free} is unavailable. Upon detecting a danger, given the information $\hat{t}_c$, $\Delta\hat{t}_c$, $\hat{d}_{c,t_r+1}$ and v_{t_r+1}, it is clear that a threat is present. It is however not clear if it is necessary to perform immediate braking or if it is possible to intervene at a later point in time. Assessing if, and when, an intervention is needed is evaluated by the collision avoidance presented in the next chapter.

5

Collision avoidance

A collision avoidance system can be described in two major parts, assessing the current threat and making a decision according to that threat.

Many different methods have been developed in attempt of assessing a threat. Since it is important to avoid or mitigate an upcoming collision, one of the biggest difficulties, when assessing the threat, is to be over conservative. This causes the system to react when the driver does not deem the situation as threatening. If one hopes to gain acceptance, of automotive consumers, this type of behaviour is not tolerable. A formal Threat Assessment (TA) algorithm, that triggers a braking intervention at the last possible instant, will be proposed in this chapter.

Decision Making (DM) algorithms are often used to decide what type of intervention should be performed in the presence of a threat. Since only one type of intervention is active in the system, the DM algorithm is trivial and will not be discussed to any further extent.

5.1 Problem description

The only interventions type included in this thesis is braking. This simplifies the problem of collision avoidance, but does not make it trivial. The threat assessment still needs to trigger the intervention at the correct time. In order to minimize the number of false interventions, the goal is to start braking as late as possible. The last possible time instance that an intervention can be triggered to avoid the upcoming collision is called the *"point of no return"*. If passed this point, then the collision is considered inevitable. Using the definition of the parameters describing the time to collision, t_c, the collision time interval, Δt_c, and distance to collision, d_c, initially introduced in Section 4.1, the objective of the collision avoidance (CA) system can be formulated as follows: The CA system should trigger a braking intervention as late as possible, but still be able to prevent the ego vehicle from travelling d_c before $t_c + \Delta t_c$.

The reference behaviour for a given collision is illustrated in Figure 5.1 as an example. It is clear, that if the intervention was triggered later, shown with a red circle, the ego vehicle would have crossed the black dashed line before the last red dashed line. Meaning that if the ego vehicle would have delayed the intervention, it would not have been able to avoid travelling $\hat{d}_c$ before $\hat{t}_c + \Delta\hat{t}_c$. The approach of intervening as late as possible is commonly used when designing minimally invasive collision avoidance systems. Some of the related systems will be discussed further to give the reader a better background to the problem.

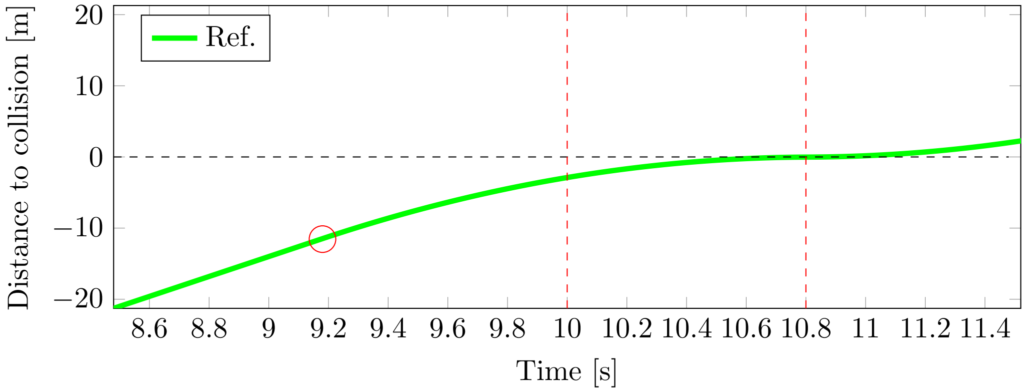


Figure 5.1: The green line shows the ego vehicles advancement towards the collision area that is indicated by the dashed lines. The distance to the black dashed line indicates the estimated distance to the collision, d_c. The interval between the two dashed red lines describes the estimated time span of the collision, t_c to $t_c + \Delta t_c$. The red circle marks the point where the braking intervention was initiated. The intervention was triggered 0.82 s before the collision at a speed of 14 m/s and $d_c = -11.48m$

5.1.1 Related work

As mentioned in previous sections, a lot of threat assessment algorithms, used in active safety applications, is based on the approach of intervening just before the point of no return. The point of no return is usually found by assuming a vehicle model and then calculating the input needed to avoid the collision. In these cases, the point of no return is usually defined as the point where all of the included vehicles are unable to avoid the collision by using any feasible control inputs. Two similar algorithm based on this approach are presented in [20] and [21].

Another way of calculating the point of no return is by using reachability tools. Reachability tools can be used to decide if the current state of the system is contained in a set from where it will evolve to a given target set for all feasible external inputs over a specific time horizon. In [22], this formal method has been used to determine the point of no return for a vehicle. In this case however the information was not used to trigger any intervention of the ego vehicle, but instead as a negotiation parameter for cooperative driving through intersections.

5.1.2 Thesis contributions

In this thesis, the point of no return is calculated by using formal reachability tools. This approach is inspired by the work presented in [22]. The contribution of the thesis is a formal TA algorithm, that can be implemented in a real time system, dealing with avoidance of frontal collision in scenarios including intersecting traffic. The proposed algorithm guarantees safety by providing tools to avoid the area of the predicted collision over a certain time span, while integrating the uncertainties connected to the distance to collision. This approach minimizes the amount of false interventions, by intervening as late as possible, but always before the point of no return, assuming the knowledge of position and time of the collision is sufficient.

5.2 A formal threat assessment

The threat assessment is based on formal reachability tools, which can be used to derive the robust controllable sets, introduced in [23]. These sets can be used to answer the question *"Will the current state of the system evolve to a specific target set for all admissible disturbances?"*. The answer provides the information needed to determine if the ego vehicle has passed the point of no return or not.

Similar evaluation has been presented in [22]. This article presents the concept of attraction sets. The attraction sets is a form of robust controllable sets that are derived by describing the vehicle as an autonomous system, where the inputs to the system is seen as disturbances. The sets are derived discretely, by back propagation, and the Nth step spans the states, from where the system will be unable to avoid, belonging to a predefined target set in N time step for all admissible inputs.

A brief introduction to the concept of reachability tools and the theory behind the robust controllable sets are presented in the sequel.

5.2.1 Preliminaries

This section is aimed to briefly develop the concept of reachability tools and robust controllable sets. For further details, the reader can refer to [23].

Robust controllable sets

In this thesis, when working with robust controllable sets, the type of systems considered are autonomous systems with additive disturbances. Hence only the theory related to that type of systems will be covered in this section. The robust controllable sets, for an autonomous system, states that if the systems is at an arbitrary state x_k at time step N and x_k belongs to the N^{th} robust controllable set, the system will evolve to a predefined target set, $\boldsymbol{\mathcal{S}}$, in N time steps for all possible disturbances. The robust controllable sets are calculated by back propagating the arbitrary system G, from the predefined target set. The target set is a polyhedron and the system being back propagated can be written in general form as,

$$x_{k+1} = G(x_k) + w_k = Ax + w. \tag{5.1}$$

The system is subject to state constraints, $\boldsymbol{\mathcal{X}}$, and the following disturbances, $\boldsymbol{\mathcal{W}}$,

$$x \in \boldsymbol{\mathcal{X}}, w \in \boldsymbol{\mathcal{W}}, \forall k \geq 0. \tag{5.2}$$

The sets $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{W}}$ are polyhedron. The set containing all backward reachable states in one time step is called the one-step robust controllable set and is defined as,

$$Pre(\boldsymbol{\mathcal{S}},\boldsymbol{\mathcal{W}}) \triangleq \{x \in \mathbb{R}^n \ : \ G_a(x,w) \in \boldsymbol{\mathcal{S}}, \, \forall w \in \boldsymbol{\mathcal{W}}\}. \tag{5.3}$$

If the current state belongs to this set, then it is guaranteed that the system will evolve into the the target set, $\boldsymbol{\mathcal{S}}$, in one time step for all admissible disturbances. The target set can be defined using the $\mathcal{H}$-representation of a polyhedron,

$$\boldsymbol{\mathcal{S}} = \{s : Hx < h\}. \tag{5.4}$$

Using equation (5.1) and (5.4), the set $Pre(\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{W}})$ can be rewritten as,

$$\begin{aligned} Pre(\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{W}}) &= \{x : \, G_a(x,w) \leq h, \forall \in \boldsymbol{\mathcal{W}}\} = \\ &= \{x : \, HAx < h - Hw, \forall w \in \boldsymbol{\mathcal{W}}\}. \end{aligned} \tag{5.5}$$

This set can be represented as the following polyhedron,

$$Pre(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}}) = \left\{ x \in \mathbb{R}^n : HAx \leq \tilde{h} \right\}, \tag{5.6}$$

where,

$$\tilde{h}_i = \min_{w \in \boldsymbol{\mathcal{W}}} (h_i - H_i w). \tag{5.7}$$

By iterating (5.3), the N-step robust controllable set, $\boldsymbol{\mathcal{K}}_N(\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{W}})$, given a target set, $\boldsymbol{\mathcal{O}} \in \boldsymbol{\mathcal{X}}$, can be defined as

$$\boldsymbol{\mathcal{K}}_l(\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{W}}) \triangleq Pre(\boldsymbol{\mathcal{K}}_{l-1}(\boldsymbol{\mathcal{O}}, \, \boldsymbol{\mathcal{W}}),\boldsymbol{\mathcal{W}}) \cap \boldsymbol{\mathcal{X}}, \, \boldsymbol{\mathcal{K}}_0(\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{W}}) = \boldsymbol{\mathcal{O}}, \, l \in \{1,...,N\}. \tag{5.8}$$

This definition states that, if the current state, x, belongs to $\boldsymbol{\mathcal{K}}_l(\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{W}})$ then it guarantees that the system will evolve to the target set, $\boldsymbol{\mathcal{O}}$, in l steps for all possible disturbances. This is true for all l between 1 and N. The last set in this series is commonly referred to as the N-step robust controllable set and it is calculated by iterating equations (5.5) to (5.7).

5.2.2 Derivation of the attraction sets

The attraction sets is a special case of robust controllable sets, introduced in Section 5.2.1, derived for a non-autonomous system by interpreting the inputs to the system as a disturbance instead. This transforms the non-autonomous system to an autonomous system. The model introduced for the ego vehicle in Section 2.4.3 as Equation (2.2) has been rewritten, to exclude redundant states and steering input, to simplify the calculations. The input u_k has also been replace by the disturbance w_k. This results in the following system,

$$\chi_{k+1} = \underbrace{\begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}}_{\text{A}} \begin{bmatrix} d_k \\ v_k \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{\Delta T^2}{2} \\ \Delta T \end{bmatrix}}_{\text{B}} w_k \tag{5.9}$$

where ΔT is the time step used in the prediction procedure in Algorithm 3.4.

The attraction sets are then derived in the same manner as the robust controllable sets, by back propagation from a target set. The target set, $\boldsymbol{\mathcal{S}}$, will from here on be referred to as the collision set, $\boldsymbol{\mathcal{C}}$, and can in this particular case be defined as,

$$\boldsymbol{\mathcal{C}} = \left\{ \begin{matrix} d_k| & 0 & \leq & d_k & \leq & d_{max} \\ v_k| & 0 & \leq & v_k & \leq & v_{max} \end{matrix} \right\}, \forall k \geq 0, \tag{5.10}$$

where v_{max} is the assumed maximum allowed speed of the system and $d_{max} = t_{pred} v_{max}$, where t_{pred} is a full prediction horizon, as defined in Chapter 3. In accordance with the reference behaviour illustrated in Figure 5.1, this collision set defines the collision to occur at $d_k = 0$. The definition of d_{max}, is the maximum distance the ego vehicle can travel over a full prediction horizon. This is done to incorporate the criteria, stated in Section 5.1, that deceleration should be the only valid intervention to avoid the collision set.

While deriving the attraction sets, the system will be subject to the following state constraints,

$$\chi_k \in \boldsymbol{\mathcal{X}} = \left\{ \begin{matrix} d_k| & -d_{max} & \leq & d_k & \leq & d_{max} \\ v_k| & 0 & \leq & v_k & \leq & v_{max} \end{matrix} \right\}, \forall k \geq 0, \tag{5.11}$$

The additive disturbances belong to the following set,

$$w_k \in \boldsymbol{\mathcal{W}} = \left\{ w_k| \quad -a_{max} \quad \leq \quad w_k \quad \leq \quad 0 \right\}, \forall k \geq 0. \tag{5.12}$$

where $-a_{max}$ represents the maximum available deceleration. The upper limit is set to zero since the only valid intervention type is deceleration.

From this system description the sets can be iteratively defined, by repeating the steps stated in Section 5.2.1 using the collision set defined in Equation (5.10). The attraction sets is then computed using a Multi Parametric Toolbox (MPT) [24] in MATLAB [25].

5.2.3 Formal formulation of the problem

Since the collision will be avoided by braking, the only way to guarantee that the ego vehicle will not be in a collision, is to ensure that it reaches the area where the collision would have occurred after the other vehicle has passed. This was formulated in Section 5.1 as: The CA system should trigger a braking intervention as late as possible, but still be able to prevent the ego vehicle from travelling d_c before $t_c + \Delta t_c$.

Assume that the trajectories of both vehicles, involved in a collision, is known and that they are driving with constant speed. It would, in this case, be easy to determine when the two vehicles were on trajectories leading to a collision. Recall that the Nth step attraction set is defined as, the set that spans the states from where the system will be unable to avoid belonging to a predefined target set, in N time step for all admissible inputs. This definition can be used to formulate a condition for when an intervention should be triggered. If it is known that the ego vehicle, in one time step from the current time k, will enter the attraction set $t_c + \Delta t_c$ time steps away, $\mathcal{A}_{t_c+\Delta t_c}$, from the collision set, $\boldsymbol{\mathcal{C}}$, an intervention should be triggered to prevent this. The condition can be formulated as follows,

$$\chi_{k+1} \in \mathcal{A}_{t_c+\Delta t_c}. \tag{5.13}$$

The assumption about known trajectories and constant speeds made in the previous paragraph is in most cases not valid. It is hence necessary to estimate the time and duration of the collision, which is performed by Algorithm 4.1. Also, if the vehicles are not assumed to travel with a constant velocity, it becomes necessary to account for driver acceleration. Since Equation (5.11) does not permit acceleration, it is possible for the ego vehicle that did not belong to $\mathcal{A}_{\hat{t}_c+\Delta\hat{t}_c}$ to accelerate in to a subsequent set, see Figure 5.2. Therefore, to ensure a correctly triggered intervention, one would need to check if the ego vehicle will belong to any of the intermediate steps between $\mathcal{A}_{\hat{t}_c}$ and $\mathcal{A}_{\hat{t}_c+\Delta\hat{t}_c}$. The condition can then be formulates as,

$$\chi_{k+1} \in \mathcal{A}_{\hat{t}_c+l}, \forall l = 0, ..., \Delta\hat{t}_c.. \tag{5.14}$$

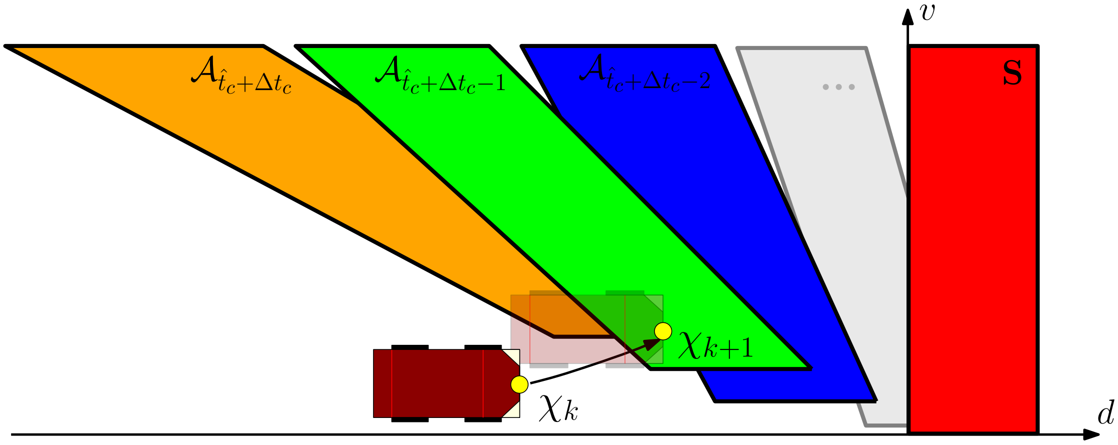

$\mathcal{A}_{\hat{t}_c+\Delta t_c}$
$\mathcal{A}_{\hat{t}_c+\Delta t_c-1}$
$\mathcal{A}_{\hat{t}_c+\Delta t_c-2}$
...
v
S
χ_{k+1}
χ_k
d

Figure 5.2: An illustration of attraction sets where the driver is not a part of $\mathcal{A}_{\hat{t}_c+\Delta\hat{t}_c}$, but due to acceleration ends up in a subsequent attraction set.

In both equations (5.13) and (5.14), the ego vehicle is assumed to be able to avoid entering an attraction set in just one time step. This is of course not true for most vehicles considering this would require an immediate deceleration. Hence, to make Equation (5.14) applicable for a real vehicle, one would need to predict the state of the vehicle at the reaction time step plus an additional time step, $t_r + 1$. The reaction time, is the time it takes for a vehicle to start decelerating and the additional time step to execute the deceleration. The prediction of the ego vehicles state at time $t_r + 1$ in to the future is provided by Algorithm 4.1. The condition that needs to be fulfilled to trigger an intervention can be written as,

$$\hat{\chi}_{(t_r+1)} \in \mathcal{A}_{\hat{t}_c+l-(t_r+1)}, \forall l = 0, ..., \Delta\hat{t}_c, \tag{5.15}$$

where $\hat{\chi}_{(t_r+1)}$ is the estimated state vector of the ego vehicle in $(t_r + 1)$ time steps.

5.2.4 Integration of the uncertainties

The condition defined in Equation (5.15) states when the ego vehicle should trigger an intervention. To find out if this condition is true, a membership test is performed, which checks if the predicted state of the ego vehicle belongs to a certain attraction set. However, if there exists an uncertainty on the states of the ego vehicle, then this method might result in an unwanted intervention.

Using robust controllable sets to analyse the controllability of a system is a powerful tool. All the uncertainties in the system could be incorporated as disturbances to the system by increasing the size of $\boldsymbol{\mathcal{W}}$, described in Equation (5.12). Increasing the size of $\boldsymbol{\mathcal{W}}$ will cause the robust controllable sets to decrease in size, as illustrated in Figure 5.3, with each iterative step of Equation (5.8).

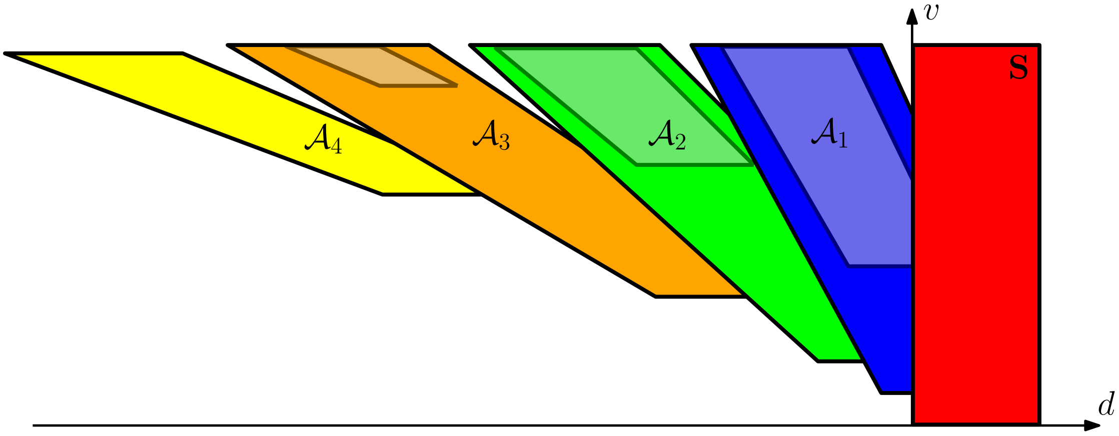


Figure 5.3: If the disturbance set, $\boldsymbol{\mathcal{W}}$, is increased in size the robust controllable set will physically decrease more in size with each step of back propagation.

Since the sets, in most applications, needs to be derived offline, an a priori estimate of the disturbances is needed. In this case, accurately estimating the uncertainty in the predicted distance to collision, $\hat{d}_{c,t_r+1}$, is very hard. It would likely result in a too conservative estimate causing the system to trigger the interventions too late. In the procedure described in Section 4.5.1.2, an online estimate of the uncertainty in the $\hat{d}_{c,t_r+1}$ is performed. This section will discuss how to stochastically incorporate this estimate in to the decision of triggering an intervention.

Recall from Algorithm (4.1) that $\hat{d}_{c,t_r+1}$ was defined as,

$$\hat{d}_{c,tr+1} \sim \mathbb{N}(d_{c,tr+1}, \sigma^2_{d_{c,tr+1}}), \tag{5.16}$$

where $d_{c,tr+1} = d_{t_r+1} - d_{t_c}$ and $\sigma^2_{d_{c,tr+1}} = \sigma^2_{d_{tc}} + \sigma^2_{d_{t_r+1}}$. This makes it possible to define the probability of belonging to a certain set as,

$$\mathcal{P}_{Belong} = \int_{\mathcal{A}_{\hat{t}_c+k-(t_r+1)}} \hat{d}_{c,t_r+1} \mathrm{d}x. \tag{5.17}$$

The solution to the integral can be approximated numerically using equations (5.18) to (5.20). N points are uniformly distributed between $-3\sigma_{d_{c,tr+1}}$ to $3\sigma_{d_{c,tr+1}}$,

$$\vec{D} = [a, a+\epsilon, a+2\epsilon, ..., a+N\epsilon], \tag{5.18}$$

where $a = -3\sigma_{d_{c,tr+1}}$ and $\epsilon = \frac{6\sigma_{d_{c,tr+1}}}{N-1}$. A series of membership tests are then performed to define which of the points belong to $\mathcal{A}_{\hat{t}_c+k-(t_r+1)}$,

$$(\vec{B})_i = \begin{cases} 1, & \text{if } (\vec{D})_i \in \mathcal{A}_{\hat{t}_c+k-(t_r+1)}, \\ 0, & \text{otherwise.} \end{cases} \quad \text{for i } = 1, ..., N \tag{5.19}$$

The probability of belonging to $\mathcal{A}_{\hat{t}_c+k-(t_r+1)}$ can now be approximated as,

$$\mathcal{P}_{Belong} = \vec{B}\vec{Z}, \tag{5.20}$$

where $\vec{Z}$ is a predefined vector containing the probability of randomly picking a value from the standard normal distribution that lies within a certain interval. To calculate this vector for the standard normal distribution $N+1$ uniformly distributed points are generated between -3σ to 3σ,

$$\vec{z} = [-3\sigma, -3\sigma+\gamma, -3\sigma+2\gamma, ..., -3\sigma+N\gamma], \tag{5.21}$$

where $\gamma = \frac{6\sigma}{N}$. By stepping trough $\vec{z}$ and using the Cumulative Distribution Function (CDF), $\phi(z)$, for a normal distribution, the probability of picking a value in each interval can be computed,

$$(\vec{Z})_i = \phi((z)_{i+1}) - \phi((z)_i) \text{ for i } = 1, \quad ..., N, \tag{5.22}$$

For more information about the CDF for a normal distribution, see [19].

The probability, $\mathcal{P}_{Belong}$, can then be compared to threshold, $\mathcal{T}_2$, to determine if the ego vehicle is predicted to belong to $\mathcal{A}_{\hat{t}_c+k-(t_r+1)}$ in (t_r+1) time steps. $\mathcal{T}_2$ controls how certain the CA system needs to be of the timing of the intervention before making the intervention. A greater value will demand higher certainty and in most cases generate a later intervention. The procedures described in equations (5.18) to (5.22) is further clarified in Figure 5.4.

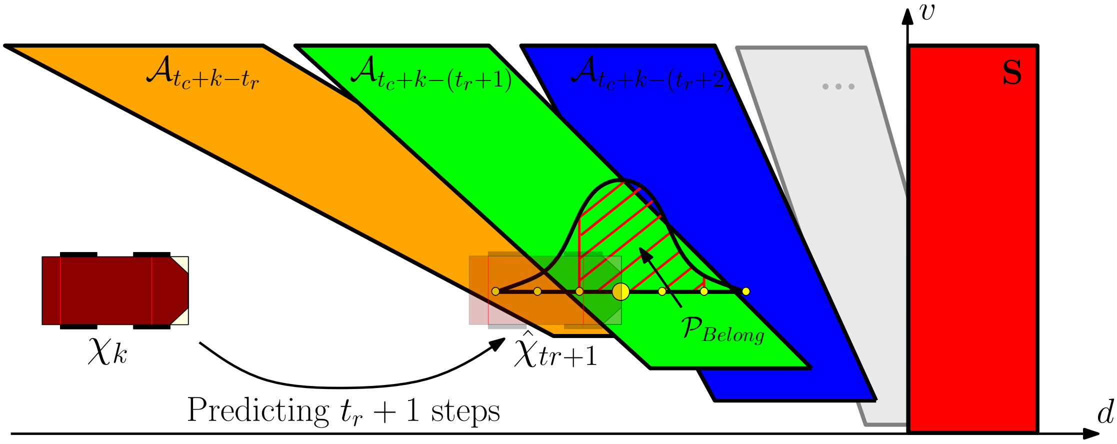

$\mathcal{A}_{t_c+k-t_r}$
$\mathcal{A}_{t_c+k-(t_r+1)}$
$\mathcal{A}_{t_c+k-(t_r+2)}$
...
S
v
d
$\mathcal{X}k$
$\hat{\mathcal{X}}tr+1$
$\mathcal{P}_{Belong}$
Predicting $t_r + 1$ steps

Figure 5.4: The ego vehicle is at predicted $t_r + 1$ steps and then a number of uniformly distributed points are generated around the predicted distance travelled and a membership test is performed to conclude which points are inside $\mathcal{A}_{\hat{t}_c+k-(t_r+1)}$. Using this knowledge the probability, $\mathcal{P}_{Belong}$, of the state of the ego vehicle belonging to $\mathcal{A}_{\hat{t}_c+k-(t_r+1)}$ in (t_r+1)s is calculated. If the probability is higher than a certain threshold, $\mathcal{T}_2$, the intervention is triggered.

5.3 Proposed algorithm

This section will define the proposed algorithm and explain how it is connected to the rest of the CA system. The algorithm is provided with the predicted estimated distance to collision, $\hat{d}_{c,t_r+1}$, and velocity, $\hat{v}_{t_r+1}$, of the ego vehicle and the current estimate of the time to collision, $\hat{t}_c$, collision time interval, $\Delta\hat{t}_c$, from Algorithm 4.1. Additionally, the proposed algorithm will also need to be initialized with the time step of the predictions algorithm, ΔT, the reaction time t_r and the attraction sets.

When all inputs are defined and everything is initialized the algorithm loops the loop variable l from 0 to $\Delta\hat{t}_c$ to check if the input $\hat{d}_{c,t_r+1}$ and $\hat{v}_{t_r+1}$ is contained in any of the sets $\mathcal{A}_{\hat{t}_c+l-(t_r+1)}$. The containment is done according to the procedure described in Section 5.2.4 and is called from the algorithm by the function $contains(\hat{d}_{c,t_r+1},\hat{v}_{t_r+1},\mathcal{A}_{\hat{t}_c})$. This function returns a probability which is then compared to the threshold, $\mathcal{T}_2$. If the probability is bigger than the threshold, then a brake signal is set to high, and the vehicle starts braking, otherwise it is set to low. In both cases Algorithm 3.4 is executed.

Algorithm 5.1: Threat assessment algorithm

```
Input  : $\hat{d}_{c,t_r+1}$, $\hat{v}_{t_r+1}$, $\hat{t}_c$ and $\Delta\hat{t}_c$
Output: $Brake$

Initialize
$\Delta T$ =Time step of the predictions algorithm;
$t_r$ =Vehicle reaction time;
$\mathcal{A}$ =All attractions sets;
Containment check
for $l = 0 : \Delta T : \Delta\hat{t}_c$ do
    $Probability = contains(\hat{d}_{c,t_r+1}, \hat{v}_{t_r+1}, \mathcal{A}_{\hat{t}_c+l-(t_r+1)})$
    if $Probability > \mathcal{T}_2$ then
        $Brake = 1$;
    else
        $Brake = 0$;
    end
end

Continue
Run Algorithm 3.4
```

6

Results & Analysis

The main goal of this thesis has been developing a system that is capable of avoiding frontal collision in the case of intersecting traffic. The proposed system has been developed as three sub-systems Path Prediction, Collision Detection and Collision Avoidance presented in chapter 3, 4 and 5 respectivly. The first part of this chapter will focus on presenting the results from the evaluation of these three sub-systems. The second part will cover the overall system performance in the test scenario presented in Section 2.2. The third part of the chapter presents a functionality test of the complete system running on a real time computer in a vehicle, reacting to inputs from Vehicle-To-Vehicle (V2V) communication. However before presenting any results the preliminaries will cover the definition of important system parameters.

6.1 Preliminaries

This section will focus on defining the system parameters needed for the collision avoidance (CA) system. The test vehicle used, when deriving the parameters, was a Volvo S60 T6 (2010). The presented parameters are applicable only for that specific vehicle.

6.1.1 System parameters

The following system parameters are used by the system

- The thresholds, $\mathcal{T}_1$ and $\mathcal{T}_2$, described in chapters 4 and 5 respectively
- Time step, ΔT used through out the proposed system.
- Prediction horizon, t_{pred}, the maximum prediction time used.
- The standard deviation of the jerk and yaw acceleration, σ_a and σ_θ, for use in the process noise matrix, introduced in Chapter 3.

- The maximum speed, v_{max} is set to 50 $\frac{km}{h}$ in accordance with the assumption made in Section 2.5. It is the maximum speed that vehicles is assumed to have in the system. It is used when deriving the attraction sets in Section 5.2.2.

- The maximum deceleration, $-a_{max}$, for use when constraining the vehicle model in Section 2.4.6 and when deriving the attraction sets in Section 5.2.2.

- Reaction time of the vehicle, t_r, used when assessing the threat in Section 5.1.

Threshold

The two different parameters, $\mathcal{T}_1$ and $\mathcal{T}_2$ are tunable parameters that control how sensitive the CA system is desired to be. The first one controls at what probability level the system will deem as a possible collision. To avoid having the collision detection too sensitive, which would lead to false detections, this threshold should generally be set to a high value. The $\mathcal{T}_1$ controls the intervention sensitivity of the collision avoidance system. Since the distance to collision, $\hat{d}_c$, is only an estimation, $\mathcal{T}_2$ controls how confident the system should be about the need of an intervention before performing one. These parameters were set as 90% and 50%, respectively.

Time step

One of the goals was for the system to be able to run in real time. This means that between each update of the system, it should be able to complete all calculations regarding path prediction, collision detection and collision avoidance. With some testing it was concluded that this limit was around 0.08 s before the system could not complete all steps in time. It is however worth noting, that the code was not optimized in any way, and this sampling time could very well be decreased drastically.

For simplicity, the same time step was used in all parts of the system and in the derivation of the attraction sets.

Prediction horizon

The prediction horizon is the length of time for which both vehicles states are predicted forward into the future. Essentially, this prediction horizon can be as long as the system designer desires. However, a very long prediction horizon only adds more computational load, and is useless as the variance of the predictions increases drastically with time, making it impossible to draw any conclusions from the predictions that lie very far from the current time. During testing and simulations, it was evident that the detection time of a collision was never above 1.52 s and therefore became the chosen value for t_{pred}. The performance of the different prediction horizon is investigated in Section 6.2.1 of this chapter.

Process noise values

To allow the Unscented Kalman filter (UKF) predictions to capture all possible paths of the vehicles, the process noise needs to capture the range of inputs a typical driver could perform. The process matrices used are

$$Q_p = \begin{bmatrix} 0.35^2 & 0 \\ 0 & 0.0135^2 \end{bmatrix}, \quad Q = \begin{bmatrix} 4^2 & 0 \\ 0 & 0.3^2 \end{bmatrix}, \quad \bar{Q} = \begin{bmatrix} Q & 0 \\ 0 & 4^2 \end{bmatrix},$$

where the Q and $\bar{Q}$ are used for filtering of the ego and other vehicle, respectively and Q_p for predictions. The motivation for the values used in the Q_p can be found in Appendix B.1.

The reason why the Q and $\bar{Q}$ differ from Q_p is related to the response time of the filter. To be able to track and filter the states when an aggressive manoeuvre is performed, like emergency braking, the process noise needs to be set high so that the filter can respond quickly to that change. However, since it is assumed that both vehicles are operating within the linear region, the process noise assumed during prediction is kept at a much lower level. This allows the system to estimate the states of the vehicles when they perform an aggressive manoeuvre, but when predicting, it will assume that the current state of the system will not change as aggressively.

Maximum deceleration and reaction time

In sections 2.4.6 and 5.1 a maximum deceleration and reaction time of the vehicle was introduced. These two parameters were set according to the results of the experiment described in Appendix B.2. From these results, the parameters were set as,

$$\begin{aligned} a_{max} &= 8.5 {}^{m}\!/\!{}_{s^2}, \\ t_r &= 0.24s = 3\Delta T. \end{aligned}$$

6.2 System evaluation

This section is split into two parts. Firstly, the accuracy of the path prediction is investigated and secondly, the performance of the CA system.

6.2.1 Prediction accuracy

A simulated left turn across path (LTAP) was generated in Matlab and fed through the prediction algorithm. Figure 6.1 shows how a predicted position compares to the actual one at that predicted time. Note that the movement of the other vehicle are shown as the sensor of the ego vehicle perceives them.

In order to investigate the robustness of the system with respect to noise, the same scenario was simulated with a higher noise level. The results are shown in Figure 6.2, where a similar behaviour is observed. In Figure 6.2 it is evident, that with added noise, the predictions become more uncertain, but the filter is still able to give an adequate

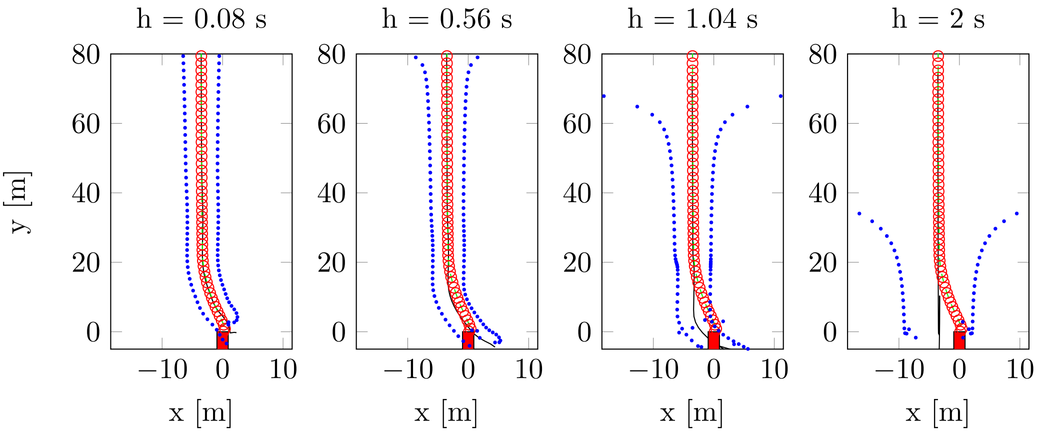


Figure 6.1: Results from the predictions of filter at different prediction times, h with low noise values of $\sigma_v = 0.01\frac{m}{s^2}$, $\sigma_R = 0.01m$, $\sigma_\theta = 0.05\,^\circ$, $\sigma_\phi = 0.01\,^\circ$ compared to the path. Here the dotted line illustrates the 95% confidence interval and the red circles are measured values. The actual path is green and the predicted one is black. The ego vehicle is shown in the bottom of all figures

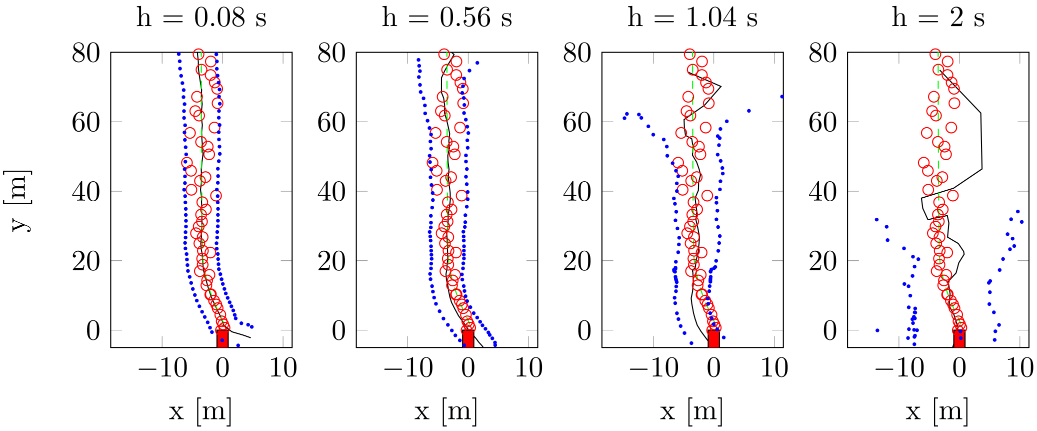


Figure 6.2: Similar configuration as in Figure 6.1 using the noise values; $\sigma_v = 0.5\frac{m}{s^2}$, $\sigma_R = 0.5m$, $\sigma_\theta = 2\,^\circ$, $\sigma_\phi = 1.375\,^\circ$.

predictions up to at least 1 s into the future. Notice that during this turning manoeuvre, the predicted position is determined to be closer to the ego vehicle than it actually will be, as is evident by the predicted and actual collision point. This will introduce some conservativeness when estimating the time to collision, $\hat{t}_c$.

6.2.1.1 Prediction conservativeness

As is evident from figures 6.1 and 6.2, the prediction algorithm can estimate that a turn takes place closer to the ego vehicle then it actually does. The reason for this behaviour is since constant turn rate and acceleration (CTRA) is assumed, the mean of the predictions, and therefore the highest probability density, will lie on the CTRA path. But if a turn is not made with a constant yaw rate, then the predictions will obviously not be accurate. The difference is illustrated in Figure 6.3. There really is no way around this behaviour known to the authors, since, in an effort to keep the system applicable to general situations, it could not be assumed that a vehicle would perform a turn with an increasing/decreasing turn rate, without knowing the road geometry. It is however worth noting, that although the most likely path in these types of scenarios is not representing the actual path, the actual path will still be captured within the variance of the predictions.

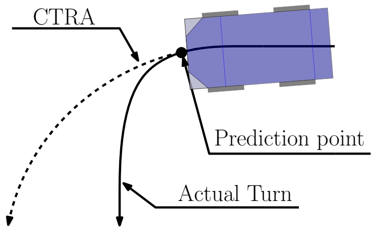


Figure 6.3: Showing how the *mean* predicted path (dotted) looks like compared to the actual (solid) when a prediction is performed from the indicated prediction point.

6.2.2 Collision detection

To validate the collision detection method, that is, calculating the probability of a collision, given the estimated positions of the ego and other vehicles' front bumper, was done as follows. Firstly, by using the derived collision detection method and secondly, by counting the number of times a collision would have happened given the two random positions. Consider the two Bivariate Normal Distribution (BVD) variables describing the position of the two vehicles, $\mathcal{E}$ and $\mathcal{F}$ as

$$\mathcal{E} \sim \mathbb{N}\left(\begin{bmatrix} 0 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \right),$$

$$\mathcal{F} \sim \mathbb{N}\left(\begin{bmatrix} -1 \\ 8 \end{bmatrix}, \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix} \right).$$

Now, let us introduce an arbitrary collision area, A, defined in Equation 6.1. The goal, as before, is to calculate the probability that the other vehicle's front bumper resides in this box relative to front bumper of the ego vehicle. Keep in mind that both the position of the ego vehicle and other vehicle are non-deterministic. An illustration of this area is shown in Figure 6.4. As before, this collision area is defined as,

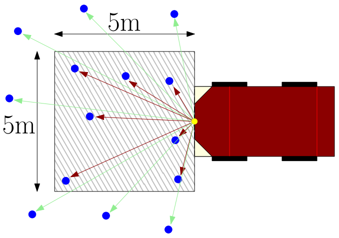
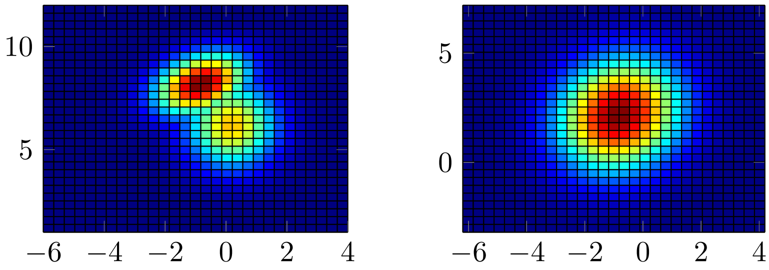


Figure 6.4: Illustrating the area of interest in this simulation. Blue dots denote the possible random vectors between the ego vehicle and the other vehicle. The green arrow represent a safe vector while the red one represent a vector that falls within the area. Not that the origin point is also stochastic, but is visually shown as deterministic for clarity.

$$\begin{bmatrix} -2.5m \\ 0m \end{bmatrix} \leq A \leq \begin{bmatrix} 2.5m \\ 5m \end{bmatrix}, \tag{6.1}$$

where the middle of the front bumper of the ego vehicle is the origin. Using this collision space, and the steps defined in Chapter 4, the probability distributions in Figure 6.5(a) are transformed into the one shown in Figure 6.5(b). To investigate the validity of the calculations, a simulation was performed, where a random point was chosen from the two distributions $\mathcal{E}$ and $\mathcal{F}$ and then counted how often the relative position was contained in A. The value was then compared to the calculated one from Equation (4.13). From Figure 6.6, it is clear, that as the number of random samples taken increase, the expected value of how often these two random points fall within A approaches the theoretical one.



(a) Probability densities of the ego and other vehicles positions.

(b) Probability density after calculation of the vector between the two probabilities.

Figure 6.5: Illustration of how the probability density changes from two randomly distributed points to a corresponding random vector.

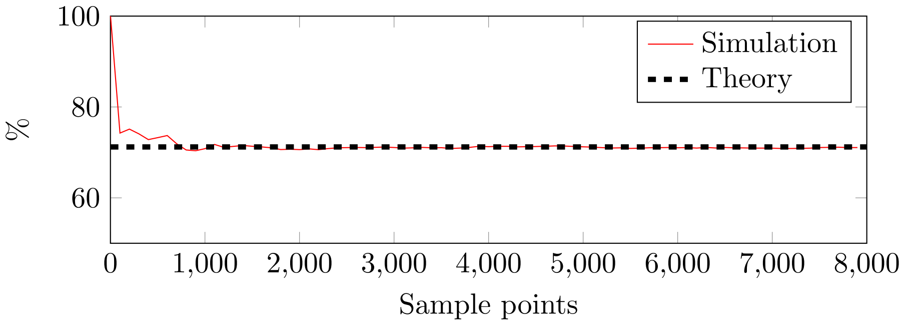


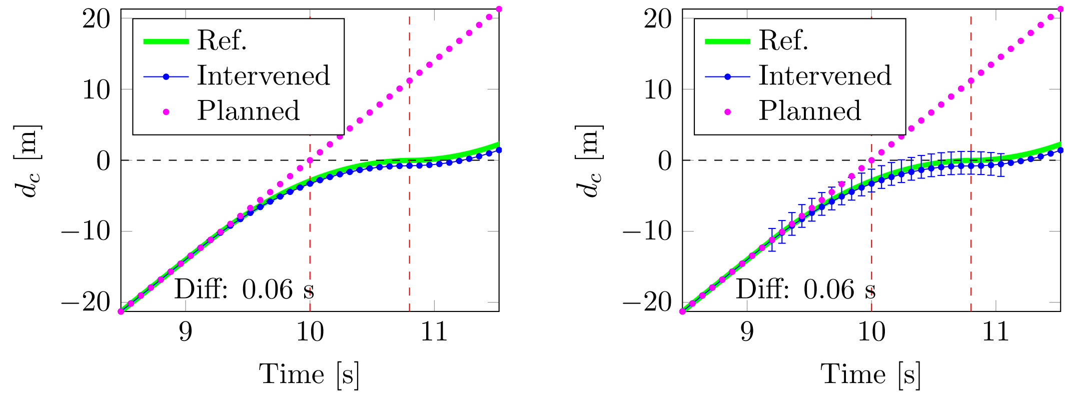
Figure 6.6: A comparison of how often the two randomly selected points from $\mathcal{E}$ and $\mathcal{F}$ land within A versus the theoretical value.

6.2.3 Attraction sets

The algorithm proposed in Section 5.3 will, in this section, be tested to evaluate if the desired behaviour described in Section 5.1 has been achieved. The tests were performed assuming perfect knowledge of the other and the ego vehicle's planned path, leading to perfect estimates of t_c and Δt_c. In the test that will be studied $t_c = 10$ and $\Delta t_c = 0.8$. An uncertainty of the distance to collision, $\hat{d}_c$, will in some of the tests be introduced to evaluate how this affect the behaviour of the collision avoidance. The objective of the tests is to investigate how close the intervention timing, derived using the proposed algorithm, is to the desired intervention timing.

Figure 6.7(a) shows a comparsion of the intervention behaviour using the proposed solution and the reference using no noise and zero reaction time. The intervention is triggered 0.06 s before the reference intervention.

Figure 6.7(b) shows the same thing as (a), but with an uncertainty of $\hat{d}_c$ and reaction time of the braking system, t_r. σ_{d_c} is set to 1.6 m and t_r is equal to 0.24 s. The intervention is triggered 0.06 s before the reference intervention.

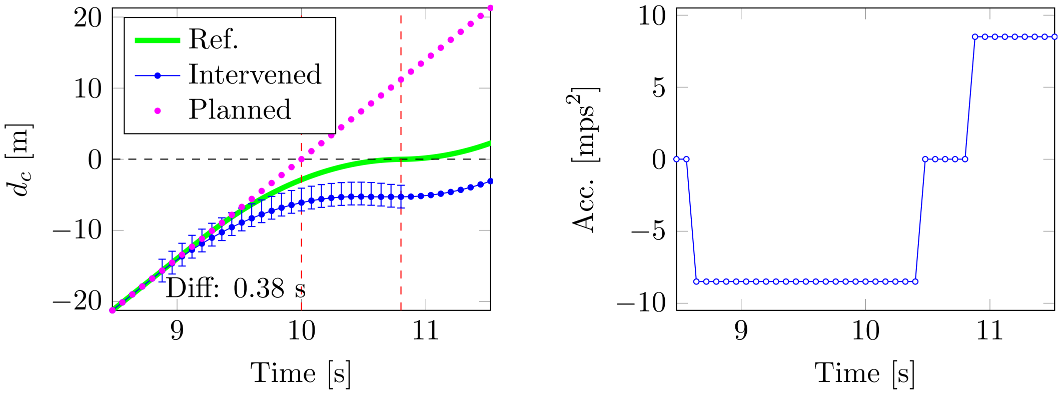


(a) The intervention behaviour of the proposed solution is very close to the reference behaviour. The intervention is triggered 0.06 s before the desired intervention.

(b) The intervention behaviour of the proposed solution is very close to the reference behaviour even though an uncertainty of the distance to collision has been introduced. The intervention is triggered 0.06 s before the desired intervention.

Figure 6.7: The braking timing is compared between the reference path and the intervened path. The planned path shows where the driver was intending to drive before the intervention, the intervened path shows the intervention behaviour of the proposed solution and the reference is the desired behaviour intervention behaviour. The black dashed line indicates when the distance to collision goes to zero and the red dashed line indicates the collision time interval. The solid vertical lines seen in Figure 6.7(b) along the Intervened path illustrates the predicted mean of the distance to collision and the variance of the uncertainty around this mean.

Both of the tests show that the CA system triggers the braking 0.06 s before the reference path. The reason for the early intervention is likely due to the time step of the CA system. The threshold $\mathcal{T}_2$, discussed in Section 6.1.1, is set to 50% in both tests. If this threshold is set lower and there exists an uncertainty in the distance to collision the intervention is triggered earlier. In Figure 6.8(a) the behaviour of the CA system is shown in the same scenario as Figure 6.7(b), but with $\mathcal{T}_2$ set to 0%. The intervention is triggered 0.38 s before the reference intervention. This is earlier than in Figure 6.7(b) and it can be seen that the path of the ego vehicle is further away from the collision due to the lower threshold.



(a) Compared to Figure 6.7(b) the proposed solution trigger the intervention earlier. This is caused by the $\mathcal{T}_2$ being equal to 0%. The intervention is triggered 0.38 s before the reference intervention.

(b) The acceleration profile applied during the avoidance manoeuvre performed in Figure 6.8(a).

Figure 6.8: The behaviour of the collision avoidance system when $\mathcal{T}_2$ is set to 0%.

In both figures 6.7 and 6.8 full braking is applied as soon as the intervention is triggered, as can be seen in Figure 6.8(b). In Figure 6.7(a) it is evident that the early intervention, due to the time step and the full braking, generates a path that is always lagging behind the referenced path. This could be solved by allowing a smaller time step or by applying less brake over the first time step. Smaller time step would introduce a higher computational load, using lower acceleration over the first time step is hence prefered. The needed acceleration over the first time step is calculated using the following equation.

$$a_{needed} = \frac{\frac{V_2 - V_1}{D_2 - D_1}(\hat{d}_{c,t_r+1} + \hat{v}_{t_r+1}\Delta T) - \hat{v}_{t_r+1} + V_2 - \frac{V_2 - V_1}{D_2 - D_1} D_2}{\Delta T(1 - (\Delta T/2)\frac{V_2 - V_1}{D_2 - D_1}} \tag{6.2}$$

where D_1 and V_1 represent the lower left corner of the attraction set being evaluated and D_2 and V_2 represents the upper left corner. In Figure 6.9 the points are marked with a circle in the respective corner of the illustrated set $\mathcal{A}_3$. $\hat{d}_{c,t_r+1}$ and $\hat{v}_{t_r+1}$ represent the distance to collision and speed in the $(t_r + 1)^{th}$ step away from the ego vehicles current position.

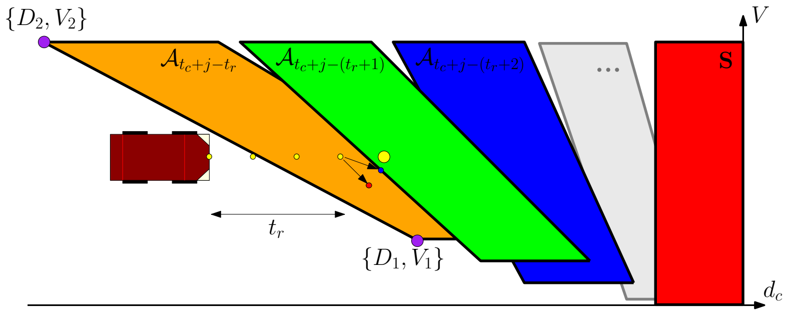
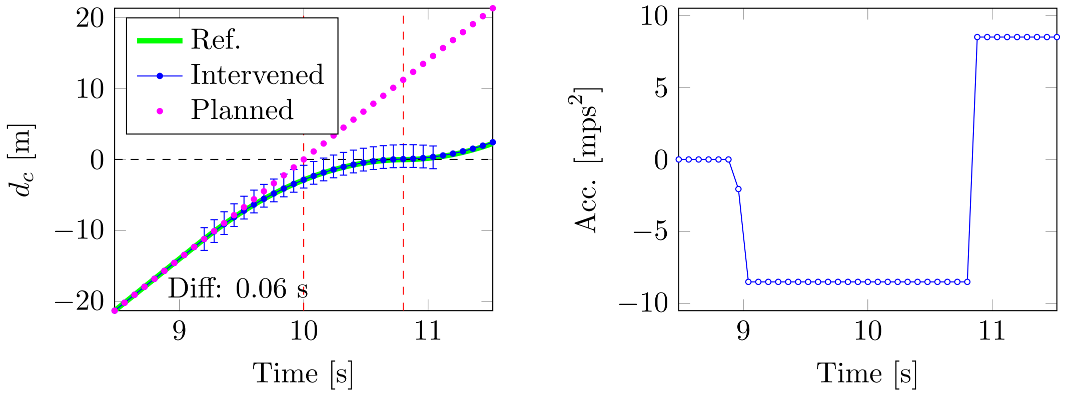

$\{D_2, V_2\}$
$\mathcal{A}_{t_c+j-t_r}$
$\mathcal{A}_{t_c+j-(t_r+1)}$
$\mathcal{A}_{t_c+j-(t_r+2)}$
...
S
V
t_r
$\{D_1, V_1\}$
d_c

Figure 6.9: The ego vehicle is predicted $t_r + 1$ steps and land inside $\mathcal{A}_3$. The green dot represents an intervention using only the needed amount of acceleration while the red dot represents applying maximum deceleration.

By using Equation (6.2) and performing the same test as illustrated previously in Figure 6.7 the result shown in Figure 6.10 is obtained.



(a) The intervention is still triggered 0.06 s before the desired intervention. However due to the lower deceleration, calculated from Equation (6.2), over the first time step of the intervention, the intervention behaviour is closer to reference compared to intervention path presented in Figure 6.7(b).

(b) The acceleration profile applied during the avoidance manoeuvre performed in Figure 6.10(a).

Figure 6.10: Evaluation of the behaviour of the collision avoidance system when applying only the needed amount of acceleration over the first time step of braking.

In Figure 6.10(a) it is evident that the behaviour of the CA system using Equation (6.2) generates a path closer to the reference path than the result presented in Figure 6.7. This approach will however not be used in the current implementation of the system. It is considered too hard to guarantee that the wanted acceleration behaviour is properly applied during the ramp up time of the braking system. Addtionally the conservativeness introduced by applying full brakes immediately can be considered small.

6.3 System performance

This section will be focusing on evaluating the performance of the complete CA system. Performance tests will be done at both low and high speeds.

6.3.1 Performance at lower speeds

The performance experiment, for low speed scenarios, was performed by implementing the system in Matlab [25] and feed it with real data. The data collection, described in Appendix A.2, was performed with a longitudinal offset due to safety reasons, and as such, the timing of the ego vehicle in the data was skewed in a way that a collision would occur, illustrated in Figure 6.11. Upon a detection of a collision, the program simulates a braking intervention with the ramp up time t_r.

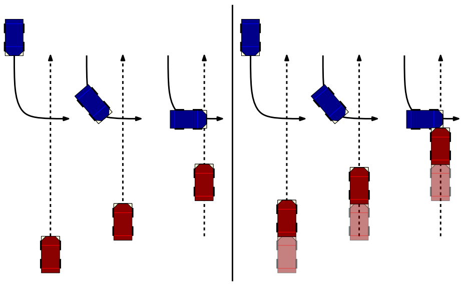


Figure 6.11: Shows the difference between before (left) and after (right) skewing the timing of the ego vehicle

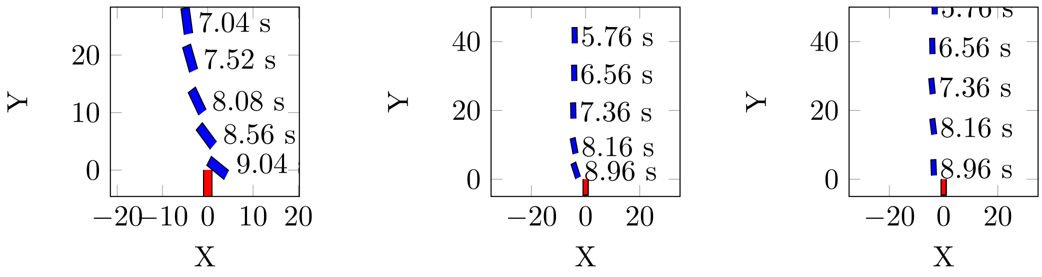
There are essentially three types of scenarios that are investigated.

- A LTAP scenario where the ego vehicle is travelling straight in his own lane.
- An avoidance manoeuvre scenario, where the other vehicle is avoiding an unidentified object in his lane, by steering towards the ego vehicles lane and then returns back in to his own lane.
- An abandoned LTAP scenario, where the driver of the other vehicle initiates a LTAP, but then realizes he will not make the turn in time and stops before entering the ego vehicles lane.

Each of these scenarios are tested both with and without added noise. Figure 6.12 shows details about each scenario. Here a collision happens in the LTAP scenario at 9.04 s and the other vehicle occupies the ego vehicles lane until 9.20 s. In the other two scenarios, no collisions happen.

6.3.1.1 Left turn across path

The experimental setup for the data used in this section can be found under *Scenario 1* in Appendix A.2 and is illustrated in Figure 6.12(a).



(a) LTAP. **(b)** Abandon turn. **(c)** Avoidance

Figure 6.12: Time evolution of each scenario. The movement of the other vehicle (blue) are shown in the reference frame of the frontal sensor of the ego vehicle (red)

Low noise simulation

All results in this section were simulated with the original measured noise of $\sigma_v = 0.1\frac{m}{s}$, $\sigma_R = 0.01m$, $\sigma_\theta = 1\,^\circ$, $\sigma_\phi = 0.1\,^\circ$. Shown in Figure 6.13, the collision is detected at 8.24 s and the collision avoidance is initiated at 8.40 s. This allows the brakes to be initiated at 8.64 s which happens to be very close to the optimal braking time of 8.67 s.

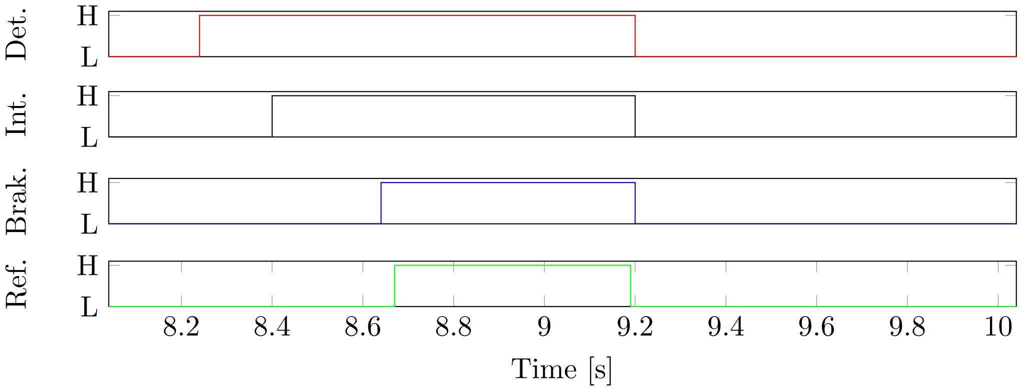


Figure 6.13: Illustration of the intervention timing of when each part of the CA system goes high. From top to bottom, threat detected, intervention performed, brakes initiate and *optimal* braking timing.

In Figure 6.14 it shows that the $\hat{t}_{c,t_r+1}$ and $\hat{d}_{c,t_r+1}$ are very accurate while the predicted collision time interval, $\Delta\hat{t}_{c,t_r+1}$, is overestimated by 0.24s. It is clear that the vehicle does not avoid entering the collision area, after the estimated time $\hat{t}_{c,t_r+1}+\Delta\hat{t}_{c,t_r+1}$, as one would expect, but that is due to the fact that the system detects that the danger has passed and the vehicle can continue driving.

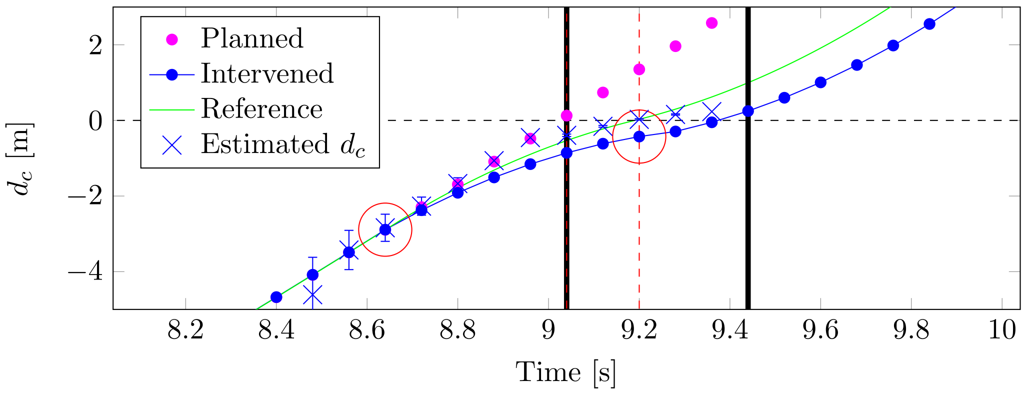


Figure 6.14: The planned path is indicating where the driver was planning on driving, the intervened path is the path proposed to avoid the collision and the reference is the desired behaviour of the collision avoidance system. The black dashed line indicates when the distance to collision goes to zero, the red dashed line indicates the actual collision time interval and the solid black lines indicates the estimated one. The blue crosses illustrate the predicted mean of the distance to collision and the variance of the uncertainty around this mean is illustrated with a blue bar. The circles indicate when brakes were initiated and released. The distance to collision before the collision is measured in negative values.

Increased noise simulation

All results in this section were simulated with measurement noise of $\sigma_v = 0.5\frac{m}{s}$, $\sigma_R = 0.5m$, $\sigma_\theta = 2\,°$, $\sigma_\phi = 1.375\,°$. Clearly, by increasing the noise of the measurements,

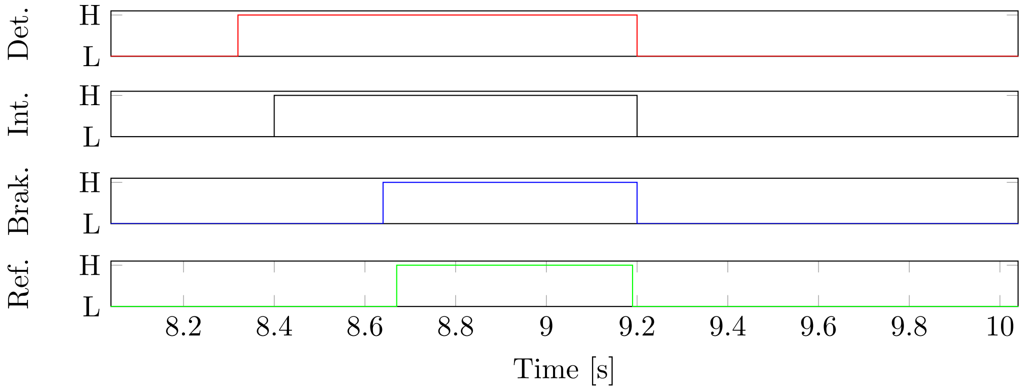


Figure 6.15: Illustration of the intervention timing of when each part of the CA system goes high. From top to bottom, threat detected, intervention performed, brakes initiate and *optimal* braking timing.

the time of detection is increased, giving the system less time to react. But still, the attraction sets deem it possible to wait for 0.16 s before braking should be applied and the intervention happens at the same time as in the low noise case. Looking at Figure 6.16 it is clear that the $\hat{t}_{c,t_r+1}$ is overestimated by 0.08 s and $\Delta\hat{t}_{c,t_r+1}$ is more than double the

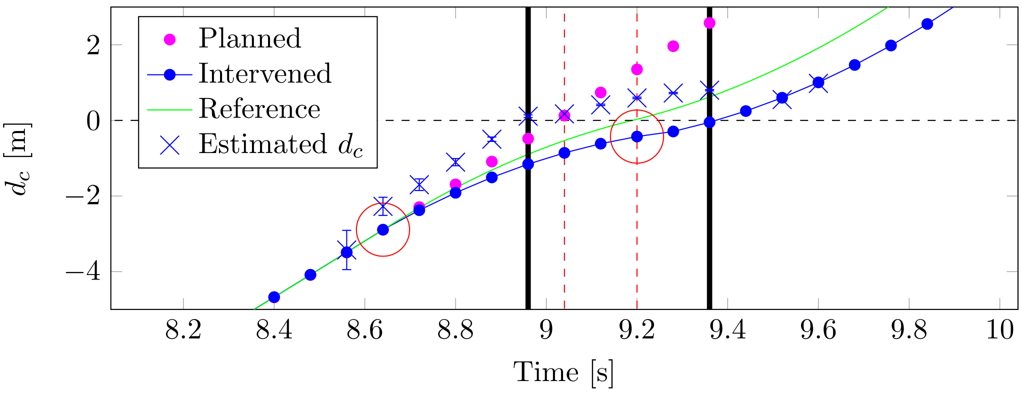


Figure 6.16: The planned path is indicating where the driver was planning on driving, the intervened path is the path proposed to avoid the collision and the reference is the desired behaviour of the collision avoidance system. The black dashed line indicates when the distance to collision goes to zero, the red dashed line indicates the actual collision time interval and the solid black lines indicates the estimated one. The blue crosses illustrate the predicted mean of the distance to collision and the variance of the uncertainty around this mean is illustrated with a blue bar. The circles indicate when brakes were initiated and released. The distance to collision before the collision is measured in negative values.

actual value. However, since the actual collision time interval is contained within these two and there was time to react, the collision is avoided. Nevertheless, these kinds of overestimation could result in overly conservative interventions.

6.3.1.2 Abandon Turn

The experimental setup for the data used in this section, can be found under *Scenario 2* in Appendix A.2 and is visualized in Figure 6.12(b). All results in this section were simulated with measurement noise of $\sigma_v = 0.5\frac{m}{s}$, $\sigma_R = 0.5m$, $\sigma_\theta = 2\,°$, $\sigma_\phi = 1.375\,°$. The low measurement noise scenarios are omitted as they are trivial considering the result when the noise is increased. The system did not initiate a false intervention since the threat level never reached the threshold. In fact, the probability of a collision was

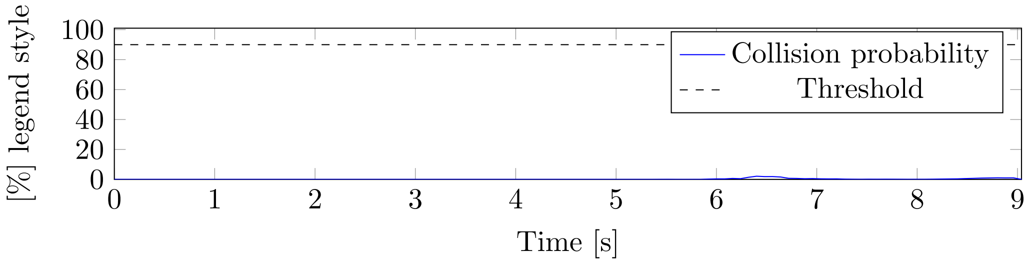


Figure 6.17: Time evolution of the calculated collision probability for the abandoning turn scenario.

very low throughout the simulation, which is desired of the system in a scenario such as this.

6.3.1.3 Avoidance manoeuvre

The experimental setup for the data used in this section, can be found under *Scenario 3* in Appendix A.2 and is visualized in Figure 6.12(c). Small skewing was done in an effort to make the avoidance manoeuvre be performed closer to the ego vehicle. All results in this section were simulated with measurement noise of $\sigma_v = 0.5\frac{m}{s}$, $\sigma_R = 0.5m$, $\sigma_\theta = 2\,°$, $\sigma_\phi = 1.375\,°$.

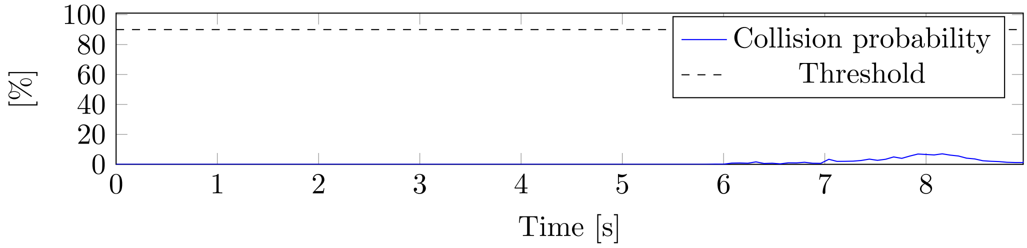


Figure 6.18: Time evolution of the calculated collision probability for the abandoning turn scenario.

As before, the system did not initiate a false intervention although a slight increase is detected around the time of the avoidance manoeuvre, it is not enough to trigger a detection.

6.3.2 Performance at higher speeds

The simulations used to test the performance of the CA system at higher speeds were performed in a software called PreScan[26]. The reason the tests at higher speed was not conducted with real data was due to safety reasons, connected to collecting the needed data. Three different scenarios were implemented in PreScan, with high and low measurement noise, and the results from these simulations will be individually presented in the following sections.

In an effort to have the data realistic, a not so perfect sensor is implemented. The current measurement models, used by the system, assumes that the measured position of the other vehicle is always at the middle of its front bumper. But realistically, using a radar for example, this can be quite challenging. Thus, in an effort to simulate this, the sensor used, measures the distance to the closest point of the other vehicle's body, this is visualized in Figure 6.19, which simulates a so called *floating* measurement point.

Since the model equation are not designed to tackle this particular problem, it is clear that some unexpected behaviour can occur. For example, from Figure 6.19 the system would assume that the middle of the other vehicles front bumper is located at the purple dot instead of the yellow one. This would make the system believe that the other vehicle is closer longitudinally than it actually is, but further away laterally. In

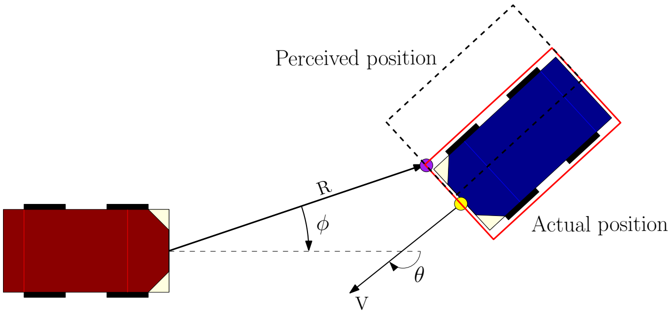
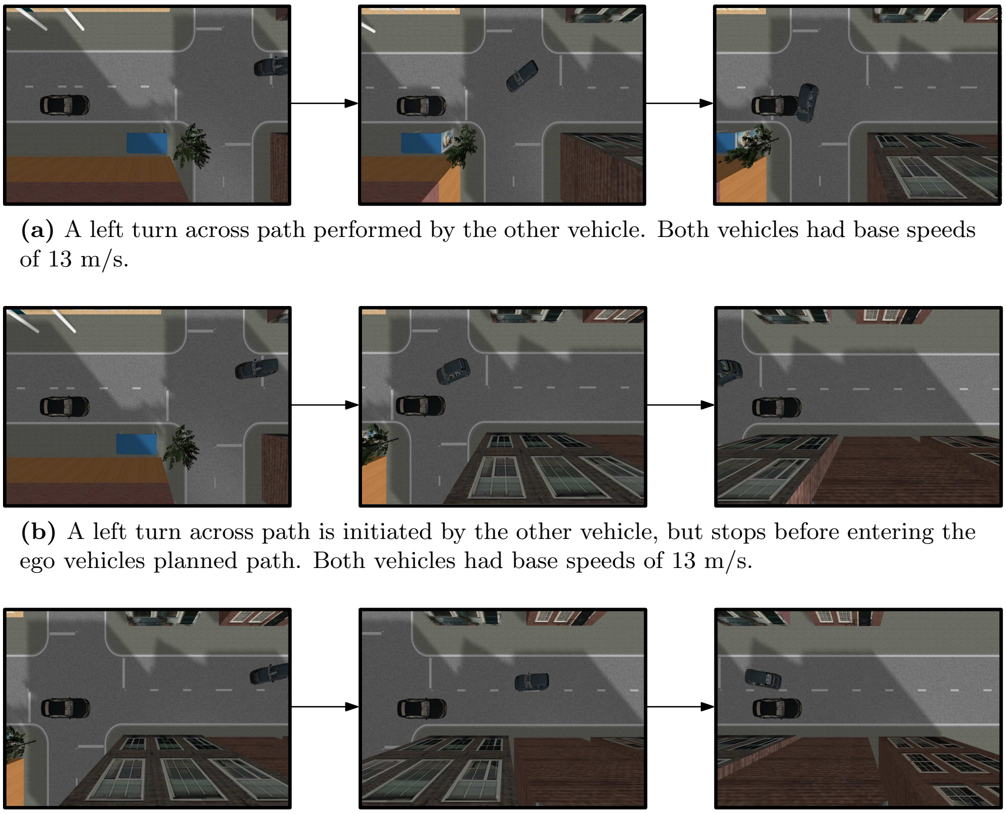


Figure 6.19: The sensor detects the closest point of the other vehicle's body. Since the system assumes that it always gets the middle point of the front bumper, it perceives the position of the vehicle at slightly different position.

the subsequent sections, the scenarios used are shown in Figure 6.20. For details about each scenario, the reader is referred to Appendix A.3.



(a) A left turn across path performed by the other vehicle. Both vehicles had base speeds of 13 m/s.

(b) A left turn across path is initiated by the other vehicle, but stops before entering the ego vehicles planned path. Both vehicles had base speeds of 13 m/s.

(c) The other vehicle avoids an obstacle in the opposing lane and then returns back to the centre of the lane. Both vehicles had base speeds of 13 m/s. This scenario has been used in the results presented in this section.

Figure 6.20: Illustration of the scenarios used in the coming sections. All scenarios are generated in PreScan.

6.3.2.1 Left turn across path

In the LTAP scenario, illustrated in Figure 6.20(a), the ego and other vehicle had a base speed of 13 m/s. The other vehicle adjusted its speed to make the turn and then it accelerated to the base speed again. The scenario was used in two different simulations. One with low noise and the other with higher noise values.

Low noise simulation

The results here are all simulated with a measurement noise of $\sigma_v = 0.05\frac{m}{s}$, $\sigma_R = 0.004m$, $\sigma_\theta = 0.7\,^\circ$, $\sigma_\phi = 0.2\,^\circ$.

As Figure 6.22 shows, the collision was effectively avoided, even though the time of when the collision should start is overestimated. The braking intervention is triggered and then abandoned due to that the estimated threat of a collision went to zero. The reason that the threat goes high and then returns to zero, might be connected to an

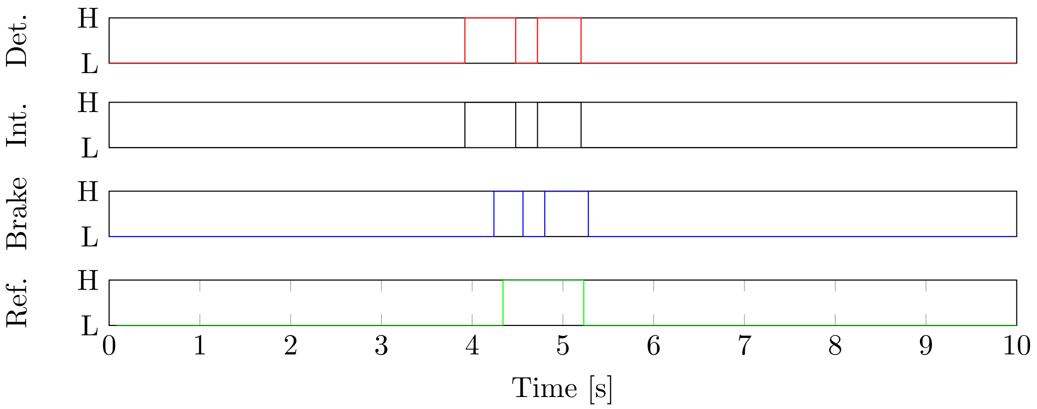


Figure 6.21: Illustration of the intervention timing of when each part of the CA system goes high. From top to bottom, threat detected, intervention performed, brakes initiate and *optimal* braking timing.

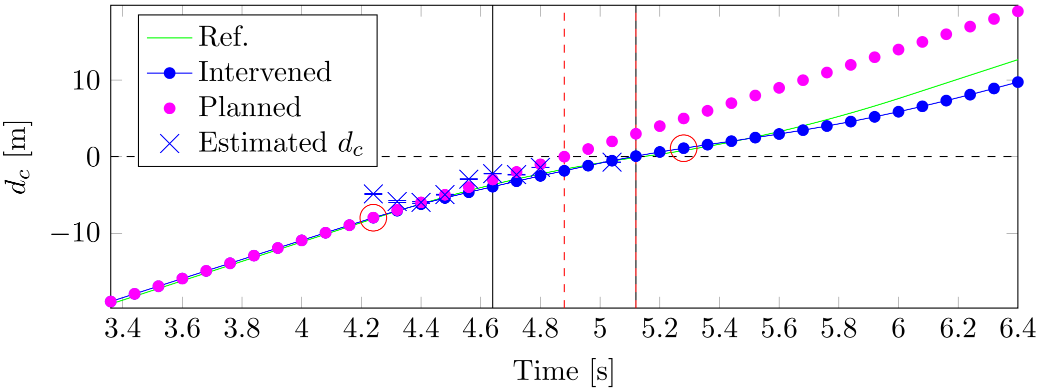


Figure 6.22: The planned path is indicating where the driver was planning on driving, the intervened path is the path proposed to avoid the collision and the reference is the desired behaviour of the collision avoidance system. The black dashed line indicates when the distance to collision goes to zero, the red dashed line indicates the actual collision time interval and the solid black lines indicates the estimated collision time interval. The blue crosses are illustrates the predicted mean of the distance to collision and the variance of the uncertainty around this mean is illustrated with a blue bar. The distance to collision before the collision is measured in negative values.

increase in yaw estimated by the filter, as shown in Figure 6.23. This increase in yaw makes the CA system believe that the other vehicle is engaging in a path that would lead to a collision earlier then the actual collision. This can be seen at time 4.21 s in Figure 6.22, where $\hat{d}_{c,t_r+1}$ is indicating that the collision would be closer than it really is. A plausible reason for the estimated increase in yaw is that the measurement point started to move along the front bumper towards the corner of the bounding box, as discussed in the introduction of this section. This causes the CA system to believe that the collision was closer than it really was and hence an intervention was triggered. This intervention was later abandoned, when measurement point had stopped moving and $\hat{d}_{c,t_r+1}$ was better, only to be triggered again due to the actual collision. This behaviour is of course not desirable and could be improved by modelling the movement of the measurement point.

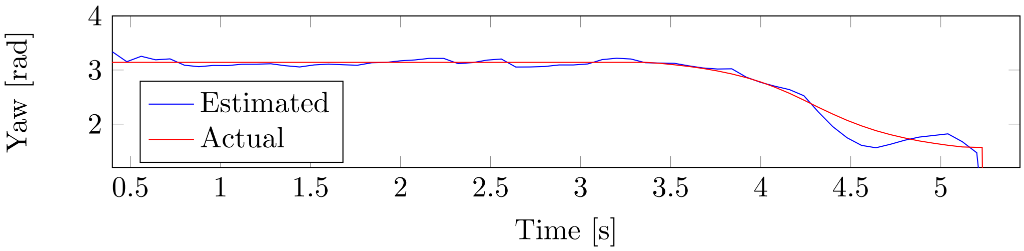


Figure 6.23: The measured and estimated yaw of the other vehicle. The estimated yaw start to increase at 4.5 s causing the collision avoidance to interrupt the intervention. This likely experienced due to the moving measurement point. After 5.2 s the other vehicle is out of sensor sight and the measurement goes low.

Added noise simulation

The results here are all simulated with measurement noise of $\sigma_v = 0.5\frac{m}{s}$, $\sigma_R = 0.5m$, $\sigma_\theta = 2\,°$, $\sigma_\phi = 1.4\,°$.

As can be seen in figures 6.24 and 6.25, the avoidance is performed in a successful manner even though noise is added, as described in the initial part of this section, to the measurement signals. Interestingly, the shortcomings of the model connected to the floating measurement point described in Section 6.3.2, is not as evident when the measurement noise is further increased. This is likely due to the fact that the filter puts less trust in the measurements when the noise is increased. In Figure 6.26 the actual and estimated yaw is compared.

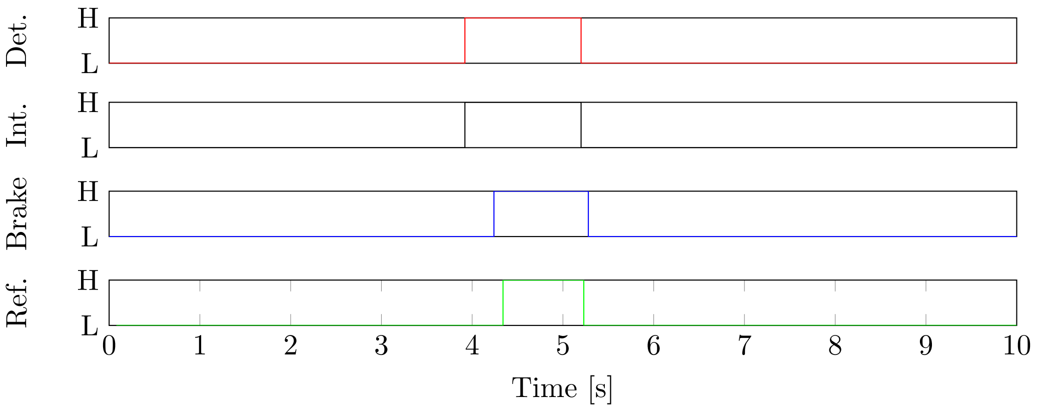


Figure 6.24: Illustration of the intervention timing of when each part of the CA system goes high. From top to bottom, threat detected, intervention performed, brakes initiate and *optimal* braking timing.

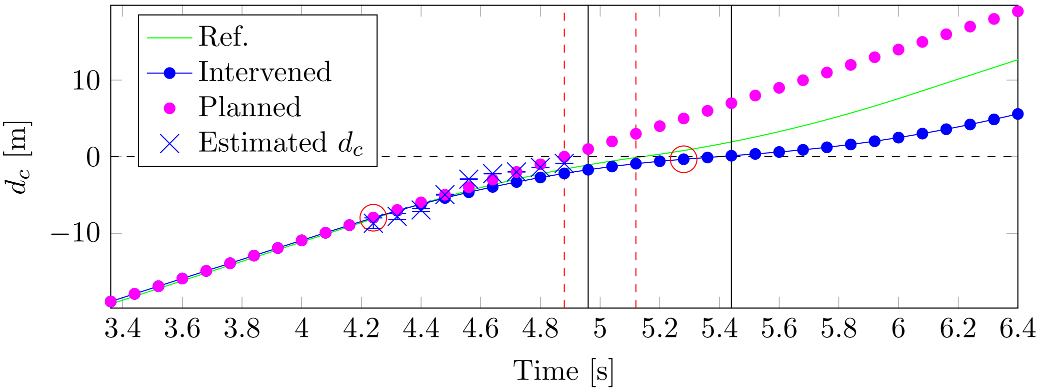


Figure 6.25: The planned path is indicating where the driver was planning on driving, the intervened path is the path proposed to avoid the collision and the reference is the desired behaviour of the collision avoidance system. The black dashed line indicates when the distance to collision goes to zero, the red dashed line indicates the actual collision time interval and the solid black lines indicates the estimated one. The blue crosses illustrate the predicted mean of the distance to collision and the variance of the uncertainty around this mean is illustrated with a blue bar. The circles indicate when brakes were initiated and released. The distance to collision before the collision is measured in negative values.

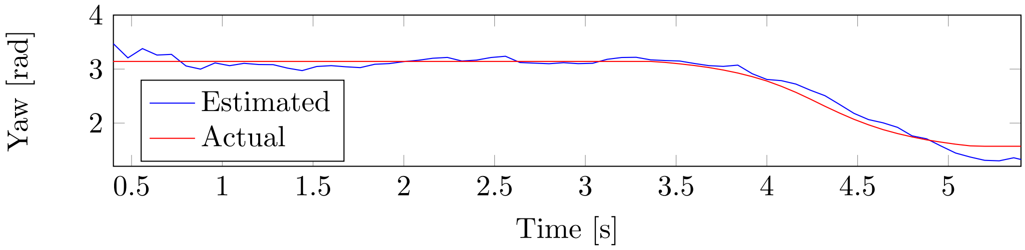


Figure 6.26: The measured and estimated yaw of the other vehicle. After 5.5 seconds the other vehicle is out of sensor sight and the measurement goes low.

6.3.2.2 Abandon turn

The abandon turn scenario illustrated in Figure 6.20(b) is meant to test how prone the proposed CA system is to false interventions.

This scenario did not trigger an intervention for either the low noise simulation or the simulation with higher noise. Hence the result will only be presented for the latter. Figure 6.27 shows how the probabilities evolved over time. The added measurement noise was, $\sigma_v = 0.5\frac{m}{s}$, $\sigma_R = 0.5m$, $\sigma_\theta = 2\,°$, $\sigma_\phi = 1.4\,°$.

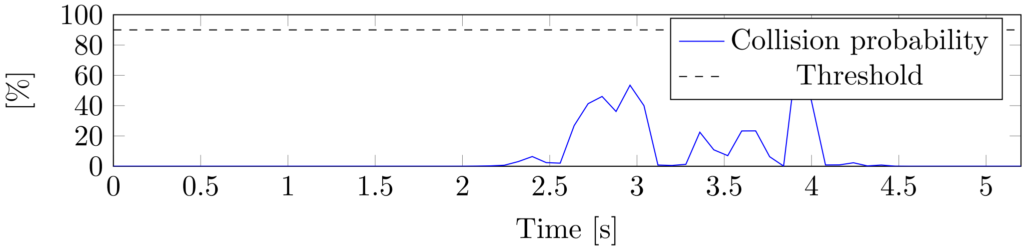


Figure 6.27: Time evolution of the calculated collision probability for the abandoning turn scenario.

The probability increased twice during the simulation. The first peak was detected when the other vehicle initiated the turn and went low again when it reached zero speed. The second peak is detected when the ego vehicle is closing in to the intersection where the other vehicle is waiting. At a certain point the other vehicle could, according to the distribution of the process noise describing the control inputs, start accelerating again and enter the ego vehicle's path at the same time instance as the ego vehicle. This is however a short time instance and that is why the sudden peak is observed. None of the peak are however high enough to trigger an intervention and hence the desired behaviour of the system is achieved.

6.3.2.3 Avoidance manoeuvre

The avoidance manoeuvre scenario, illustrated in Figure 6.20(c), is in the same way as the previous scenario meant to test how prone the proposed CA system is to false interventions. This scenario did not trigger an intervention for either the low noise simulation or the simulation with higher noise. Hence the result will only be presented for the later case and only include Figure 6.28 that shows how the probabilities evolved over time. The added measurement noise was, $\sigma_v = 0.5\frac{m}{s}$, $\sigma_R = 0.5m$, $\sigma_\theta = 2^\circ$, $\sigma_\phi = 1.4^\circ$.

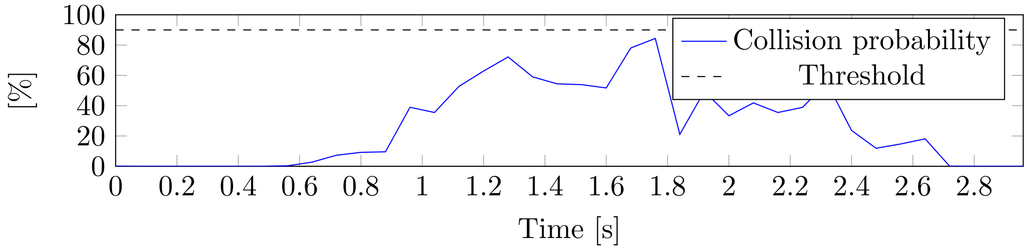


Figure 6.28: Time evolution of the calculated collision probability for the abandoning turn scenario.

The probabilities evolves as expected considering the driving behaviour applied by the other vehicle. Due to the threshold the intervention is however never triggered and desired behaviour is achieved.

6.3.2.4 Aggressive left turn across path

The aggressive LTAP is initiated earlier than the previous LTAP scenario, as can be seen in Figure 6.20(c).The scenario was used in two different simulations one with low noise and one with the same added noise as from previous sections.

Low noise simulation

The results here are all simulated with the original measured noise of $\sigma_v = 0.05\frac{m}{s}$, $\sigma_R = 0.004m$, $\sigma_\theta = 0.7^\circ$, $\sigma_\phi = 0.2^\circ$.

As can be seen in figures 6.29 and 6.30 the collision was avoided in an almost optimal manner. For the observant reader this might seem peculiar since the estimated time to collision and collision time interval do not correlate with the actual values. However, since both $\hat{t}_{c,t_r+1}$ and $\Delta\hat{t}_{c,t_r+1}$ are underestimated, the optimal intervention timing for the estimated and actual collision coincide.

To investigate this further the reference behaviour for both the estimated collision and the actual collision is calculated. The two reference intervention signals is shown in Figure 6.31, together with the brake signal presented in Figure 6.29. By comparing the two interventions signals with the brake signal, it is evident that both trigger signals would have generated a similar intervention time for the emergency braking. That the estimated and the actual collision in this case generate similar behaviour is only a coincident.

Nevertheless, it should be noted, that the reason the braking performed is so close to the actual reference signal, after the triggering, is not coincidental. This can be

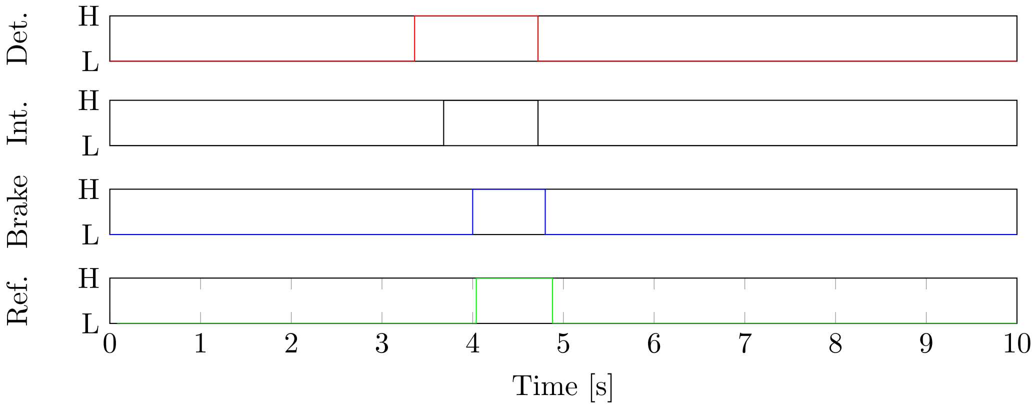


Figure 6.29: Illustration of the intervention timing of when each part of the CA system goes high. From top to bottom, threat detected, intervention performed, brakes initiate and *optimal* braking timing.

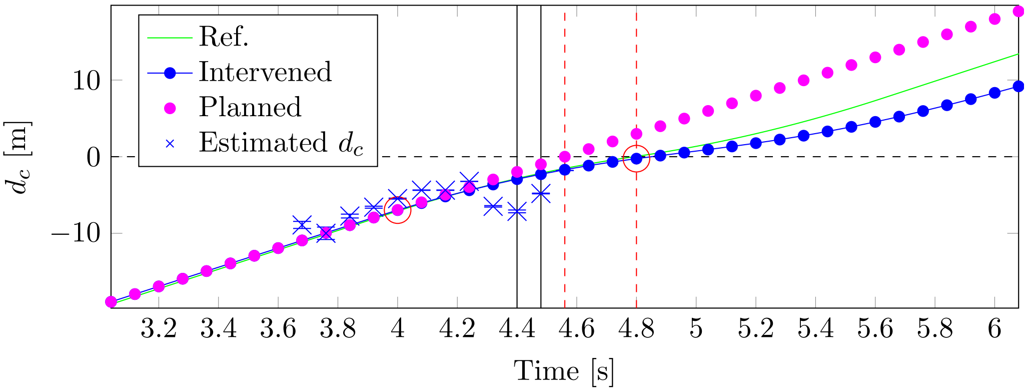


Figure 6.30: The planned path is indicating where the driver was planning on driving, the intervened path is the path proposed to avoid the collision and the reference is the desired behaviour of the collision avoidance system. The black dashed line indicates when the distance to collision goes to zero, the red dashed line indicates the actual collision time interval and the solid black lines indicates the estimated collision time interval at the time the intervention was triggered. The blue crosses are illustrates the predicted mean of the distance to collision and the variance of the uncertainty around this mean is illustrated with a blue bar. The distance to collision before the collision is measured in negative values.

shown by generating the same plot shown in Figure 6.30, but instead show $\hat{t}_{c,t_r+1}$ and $\Delta\hat{t}_{c,t_r+1}$ at a later time instance. This results in a plot, shown in Figure 6.32, that indicates that the estimated collision improves while the vehicle is braking and hence the performed braking behaviour is closer to the reference behaviour than the initially estimated collision would suggest.

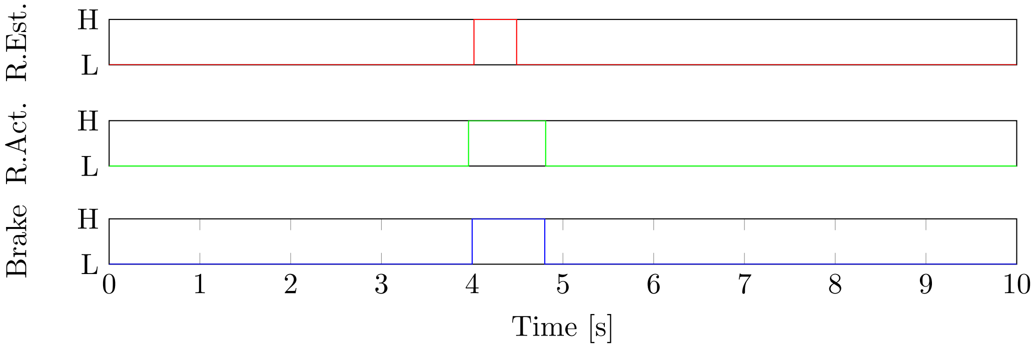


Figure 6.31: Intervention timing for different collision times. Top and middle figure show the optimal timing for the estimated and actual collision times respectively. The bottom one shows the performed intervention timing in the simulation.

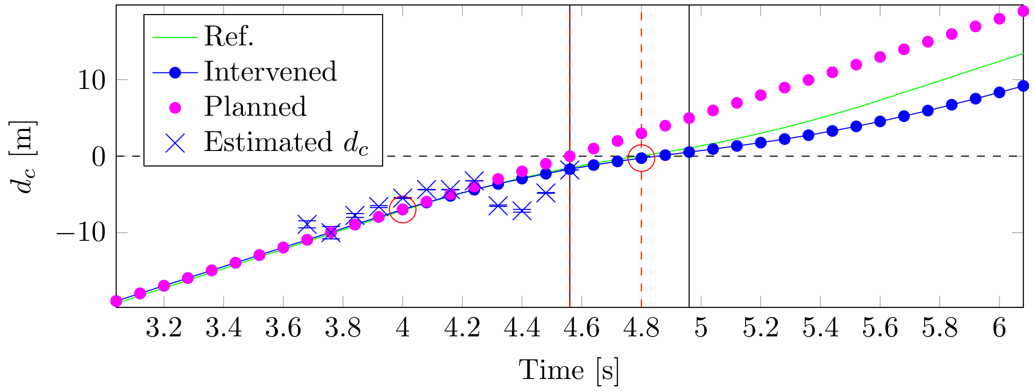


Figure 6.32: Shows the same plot as in Figure 6.30, but instead of showing the estimated collision at the intervention time it is shown at the 5^{th} time step after the intervention time.

In Figure 6.32 it can be seen that the estimated collision is closer to the actual one. $\Delta\hat{t}_{c,t_r+1}$ is still overestimated, but since the predictions will tell the CA system when to stop braking the intervention is still performed in a near optimal way.

Added noise simulation

The results here are all simulated with measurement noise of $\sigma_v = 0.5\frac{m}{s}$, $\sigma_R = 0.5m$, $\sigma_\theta = 2\,°$, $\sigma_\phi = 1.4\,°$.

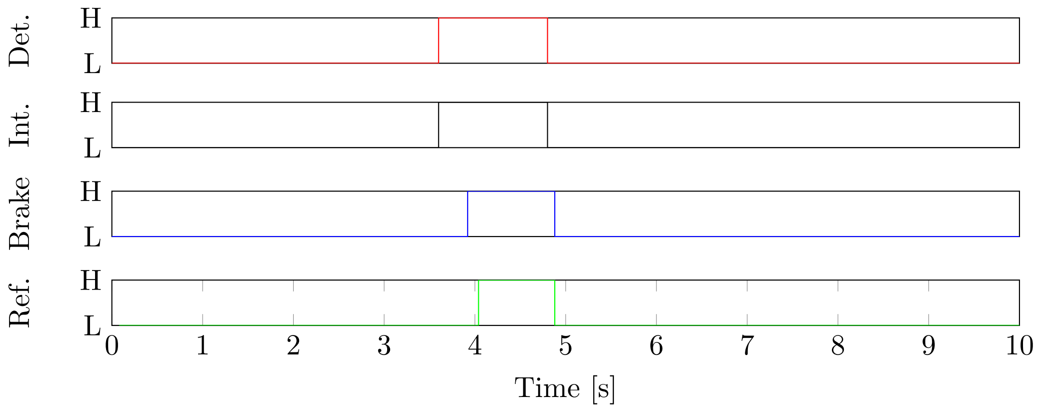


Figure 6.33: Illustration of the intervention timing of when each part of the CA system goes high. From top to bottom, threat detected, intervention performed, brakes initiate and *optimal* braking timing.

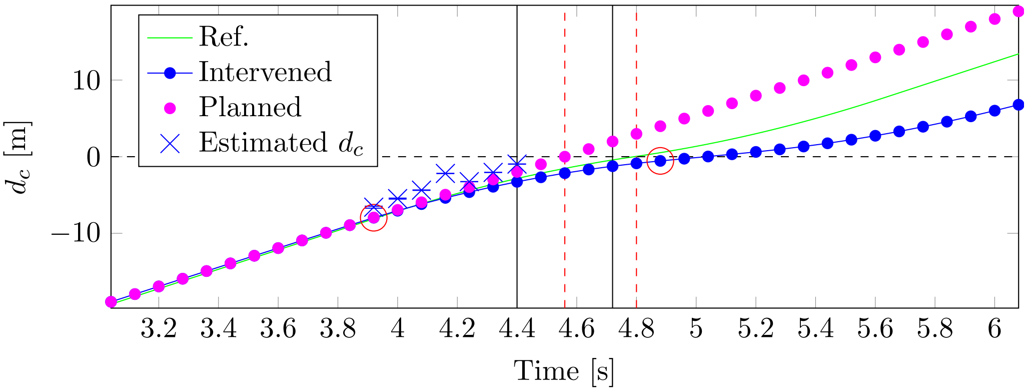


Figure 6.34: The planned path is indicating where the driver was planning on driving, the intervened path is the path proposed to avoid the collision and the reference is the desired behaviour of the collision avoidance system. The black dashed line indicates when the distance to collision goes to zero, the red dashed line indicates the actual collision time interval and the solid black lines indicates the estimated collision time interval. The blue crosses are illustrates the predicted mean of the distance to collision and the variance of the uncertainty around this mean is illustrated with a blue bar. The distance to collision before the collision is measured in negative values.

When the noise was added, a more conservative intervention was performed. The intervention was triggered 0.08 s earlier compared to when the system was subject to only low measurement noise. This is likely due to the higher uncertainties that are present. The same effect of underestimating the time to collision noted in the simulation with lower noise can be noted in this result as well. However it is not as evident as in the previous case. Overall the simulation generated a robust and effective avoidance manoeuvre.

6.4 Functionality test

A preliminary functionality test were performed. The functionality test is the first step to verify that the system is functioning properly when implemented in a vehicle. This is done before real testing of the system is started, in an effort to find eventual bugs and implementation errors in the system. The tests were done in two steps. First by replaying collected data on the CAN-system, connected to a test rig, including a real time vehicle computer running the complete CA system. The goal was to verify that the system compiled correctly and that all sensor signals were interpreted correctly. Secondly, to upload the compiled system to a vehicle computer in a real vehicle and feeding measurement data in real time, from the installed sensors, to the system. This was intended to verify basic functionality of the system.

In this section the results will be presented for the second part of the functionality test. The result of the first part has been left out since the results of the second part verifies that the test of the first part was successful. To be able to understand the result presented in this section a description of how the test was performed is necessary.

6.4.1 Test setup

A problem with testing CA systems in practice is that you need to trigger an intervention without posing any threat to the included vehicles and their drivers. This was solved in this case by setting a lateral offset on the other vehicle. In Figure 6.35 it is illustrated where the other vehicle, represented in blue, actually is and where the ego vehicle, shown in red, perceived it to be.

The equipment used in this experiment was the same used in the data collection described in Appendix A However due to the complications introduced when introducing an offset the laser scanner was not used, instead all measurements was done using DGPS and V2V communication.

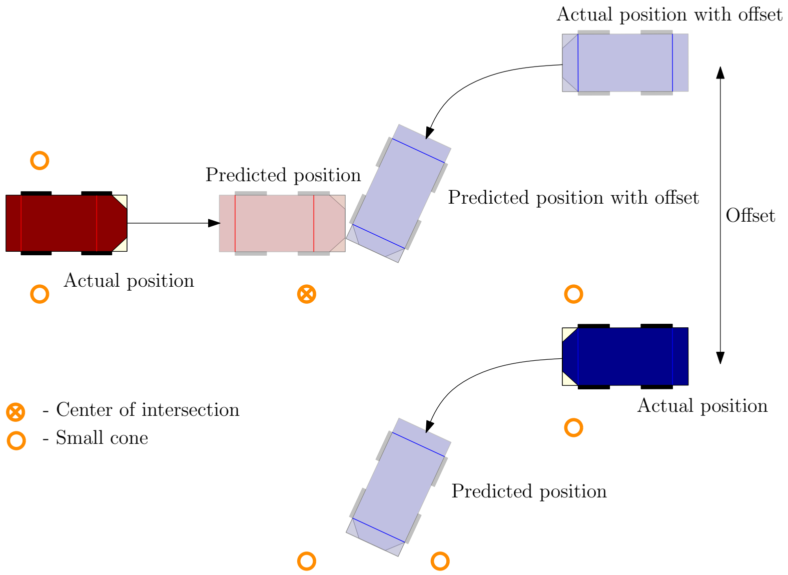


Figure 6.35: The scenario configuration used when performing the functionality tests. The other vehicle, blue, has a lateral offset making the ego vehicle, red, think that the other vehicle is actually turning in front of him and hence an intervention is triggered without posing any real threat to any of the involved vehicles.

6.4.2 Test result

As stated in the introduction to this section the tests that has been performed is preliminary functionality test. More tests were initially planned to be performed, but due to technical issues it was not possible in the time span of the thesis to perform further testing. The result presented below is hence not representative of the functionality of the final system, since no time was spent tuning the system before the preliminary test. The soul objective was to verify basic functionality. Due to this fact, the other vehicle was stationary and positioned in such a way that the other vehicle was perceived to be located inside of the ego vehicles lane due to the introduce offset.

The result of the test is presented in figures 6.36 and 6.37. All signals used to generate these plots are collected from the CAN-system of the ego vehicle. It should be noted that the intervention signal did not trigger an actual braking intervention during the test due to legal regulations. The intervention noted in Figure 6.37 was calculated afterwards to illustrate how the intervened path would have looked like if brakes would have been applied.

As can be seen in Figure 6.37, marked with a plus sign, the collision was detected at 10 time steps before the collision which translates to 0.2 s with the time step used in the vehicle computer. Due to the ramp up time the braking will then not be applied until 14.18 s. This is illustrated with a circle in Figure 6.37. Applying the brakes at this time instant makes it impossible to avoid the collision. To be able to avoid the collision the brakes would have needed to been applied at 13.6 s, as can be seen in Figure 6.36.

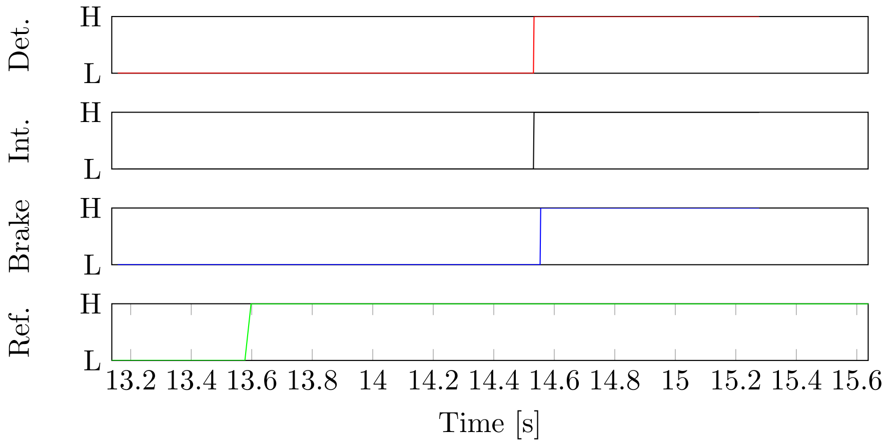


Figure 6.36: Illustration of the intervention timing of when each part of the CA system goes high. From top to bottom, threat detected, intervention performed, brakes initiate and *optimal* braking timing.

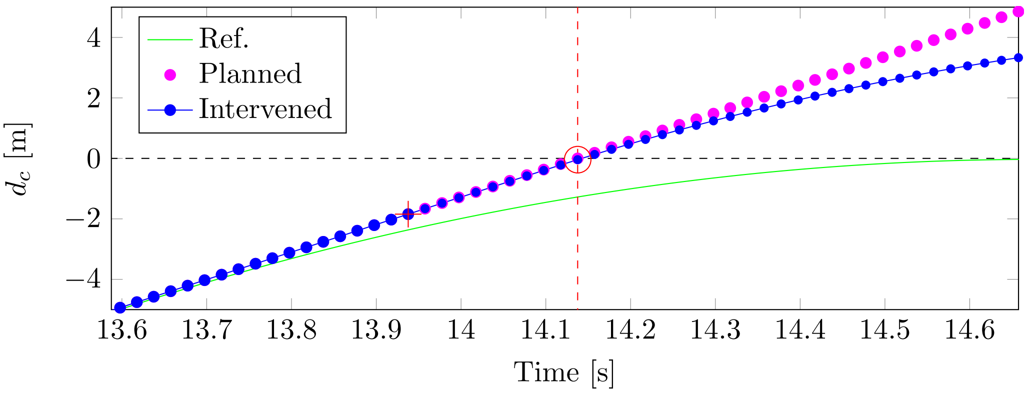


Figure 6.37: The planned path is indicating where the driver was planning on driving, the intervened path is the path proposed to avoid the collision and the reference is the desired behaviour of the collision avoidance system. The black dashed line indicates when the distance to collision goes to zero, the red dashed line indicates the actual time of the collision. The red cross indicates were the intervention was triggered and the red circle indicates were the braking was applied. The distance to collision before the collision is measured in negative values.

This is of course not a desired behaviour, but it was expected since the system had not been properly tuned. However no matter if the behaviour was desired or not, the result of the test was successful since the basic functionality of the implemented system could be verified.

7

Discussion & Concluding remarks

In Section 1.3, four general objectives of the thesis were stated,

1. Track and predict the paths of the surrounding traffic providing a probabilistic, time dependent occupancy map of the vehicles positions.
2. Develop an efficient way to detect the risk of a collision by using the information of the occupancy map.
3. Assess in a formal and robust way the need for an intervention at the current time instance to minimize the number of unnecessary interventions.
4. Implement the proposed collision avoidance (CA) system in a vehicle.

This chapter will focus on discussing the result that is presented in Chapter 6 and draw conclusions based on both the result and the discussion. The concluding remarks will determine if the presented system fulfilled the objective of the thesis. The discussion will provide further elements on the performance, stability and robustness of the proposed solution. To cover as many aspects of the system as possible the discussion will be divided in to three sections: system design, system applications and future work.

7.1 System design

In accordance with the objectives of the thesis, the proposed system has been divided in to three parts: path prediction, collision detection and collision avoidance. Each part of the system targets one of the first three objectives. The performance of the system relies on the performance of each part and hence the different parts will be discussed separately.

7.1.1 Path prediction

As mentioned before, the path prediction is performed using the Unscented Kalman filter (UKF). A well defined process noise matrix allowed for predicting possible future paths of a vehicle. These predictions perform well up to at least 1 second ahead in time, as shown in Section 6.2.1. The prediction horizon is enough to perform avoidance manoeuvres in the traffic scenarios studied in this thesis. However increasing this time would enable more possibilities, e.g., implementing a warning to alert the driver of the threat or autonomous adaptation of the speed to avoid the collision as opposed to using emergency braking.

There are several ways to increase the accuracy of the prediction and hence increase the odds of detecting possible collisions earlier. One way would be to integrate statistical knowledge of how a driver usually makes a certain type of manoeuvre. The type of knowledge presented in [27] and [28] could be combined and integrated in the system to know what type of turn the other vehicle would most likely perform based on its state. These turns could then be compared with one or several pre-calculated paths to see which one of the paths is the most likely. This type of multiple hypotheses testing has not been attempted in this thesis, but is common within the field of localization and tracking. An example of UKF integrating multiple hypotheses is found in [29]. This approach will increase the computational burden of the solution, but could also make significant differences for the prediction accuracy.

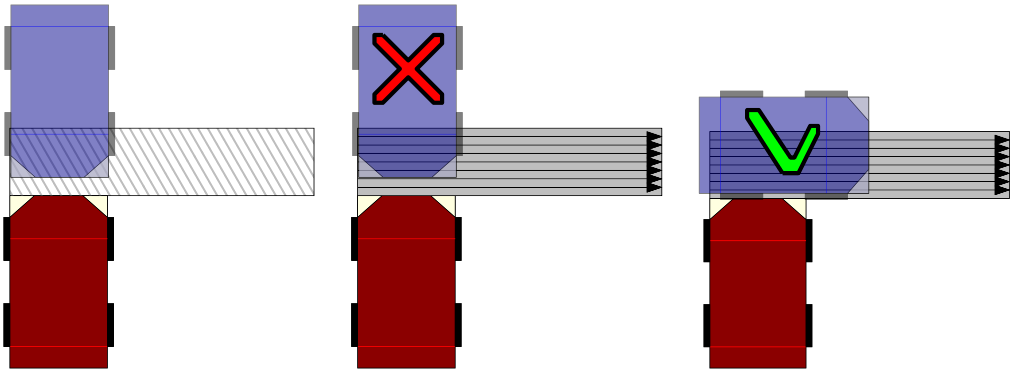
Another way of improving the performance would be to include knowledge of the road geometry. This knowledge could be acquired by gaining access to map data. Petrich. D et al. [30] use map knowledge in conjunction with an Extended Kalman filter (EKF) to determine which path a vehicle belongs to, with a certain probability. This could potentially be used to improve this CA system drastically in terms of prediction time and false detections.

7.1.2 Collision detection

The probabilistic collision detection based on bivariate normal distributions, proposed in Chapter 4, allows an efficient calculation of the probability of a collision. In low noise environments it is able to accurately predict the time to collision, however from the results presented in Section 6.3.1.1 it is evident (see Figure 6.14) that the result might be worse when higher noise levels are introduced. If the noise levels are increased, the proposed detection method struggles to estimate the collision time interval with the same accuracy and is prone to overestimating the time to collision. This can lead to an earlier intervention than necessary (see Figure 6.15). This causes a more conservative behaviour which can lead to more false interventions. The collision will be avoided, but the driver might find the system too intrusive.

A poorly estimated time to collision might also cause other problems. By design, the distance to collision, estimated by the proposed system, is strictly related to the predicted time to collision. This stems from the fact that the distance is calculated as how far the ego vehicle could travel during the remaining time to collision, which is based on the predicted speeds and accelerations. Thus, when the time to collision is under- or overestimated, the distance to collision will increase or decrease accordingly. This means that, using this method, the uncertainty in distance to collision is not captured as a uncertainty of the collision point, but rather as the uncertainty of the distance the ego vehicle can travel up to the estimated collision. This property of the system makes it even more important that the time to collision is estimated accurately.

A way of improving the estimate of the time to collision, and hence the collision detection, could be by incorporating the relative heading into the vector used by the collision detection. This would enable a new degree of freedom to the collision definition. Figure 7.1 illustrates an extreme case where the current system could benefit of the improvement. Figure 7.1(a) shows a front-to-front collision. The scope of the thesis is to design a CA system that focus on frontal collisions in cases of intersecting traffic. The proposed system is hence not designed to avoid front-to-front collision and should not detect that type of collisions either. If the additional degree of freedom was introduced, as illustrated in Figure 7.1(b), the front-to-front collision would not fulfil the collision definition. The only possible collision to fulfil the definition would be when the other vehicle is travelling on an orthogonally intersecting path, as it is illustrated in Figure 7.1(c). This example clearly shows how the additional degree of freedom could be used to more precisely define what should be considered a collision. A more precise definition would make it possible to estimate the time of collision better and make the system less prone to false detections.



(a) Since the heading of the other vehicle is currently not taken in to consideration it is possible to detect front-to-front collisions.

(b) By incorporating the heading of the the other vehicle in to the system, illustrated as arrows, detection of front-to-front collisions could be avoided.

(c) Since the other vehicle's heading is alligning with heading incorporated in the system, illustrated by arrows, a frontal collision with intersecting traffic can be correctly defined.

Figure 7.1: An example of when the proposed collision detection might trigger a false intervention and how the introduction of the other vehicle's heading could improve the system.

7.1.3 Collision avoidance

The CA system proposed in Chapter 5 is designed to be minimally intrusive and emergency braking is only triggered, to prevent the ego vehicle to go passed the point of no return. After this instant, the collision could still be avoided by another type of intervention, e.g. a combination of steering and braking. To only include one intervention type in the system was a design choice to decrease the complexity of the system and narrow down the scope of the thesis. The proposed system has proven efficient to avoid collision, in the scenarios evaluated in Chapter 6, but applying more intervention types could improve the effectiveness of the CA system. To be able to include steering interventions, the CA system would need to include a Decision Making (DM) algorithm suggesting how the new intervention should be executed in a safe way with respect to all threats in the ego vehicles surroundings. Inspiration for future work on this subject can be found in [21].

Expanding the system with a DM algorithm and new interventions types is however not the only way to improve upon the proposed solution. A critical part of the intrusiveness of the system is that the Threat Assessment (TA) algorithm triggers a minimal amount of false interventions. The TA algorithm, proposed in Section 5.3, integrates the variance of the distance to collision, σ_{d_c}, and the variance of the distance the ego vehicle has travelled over the reaction time, $\sigma_{d_{t_r+1}}$. The integration of these variances makes the system less intrusive in uncertain situations, hence minimizing the number of false interventions. This could be further improved by incorporating the effect that the sensor noise have on the variance of the distance to collision. This could be achieved by the following change to Equation (5.16),

$$\sigma^2_{d_{c,t_r+1}} = \sigma^2_{d_c} + \sigma^2_{d_{t_r+1}} + \sigma^2_{d_c,sns}. \tag{7.1}$$

The estimation of $\sigma^2_{d_c,sns}$ should preferably be done online since the noise of real sensors is often dependent on external factors such as range to objects and attenuation due to weather.

7.2 System applications

The primary focus was on the left turn across path (LTAP) scenario where the ego vehicle is going straight, but the system was developed for general cases. The path prediction is not dependent on anything else than the constrains introduced by the vehicle model used and the sensor readings of how the ego and other vehicle is behaving. This makes it possible to adapt the proposed system several other scenarios with no or just a few simple modifications.

For example, if the ego vehicle was equipped with a sensor with wide enough field of view, the system could be used in 4-way intersections or roundabouts where the other vehicle is approaching from the left. Also if the collision areas defined in Chapter 4 were changed according to Figure 7.2, the system would also work for vehicles approaching from the right. Another interesting scenario would be the LTAP scenario where the

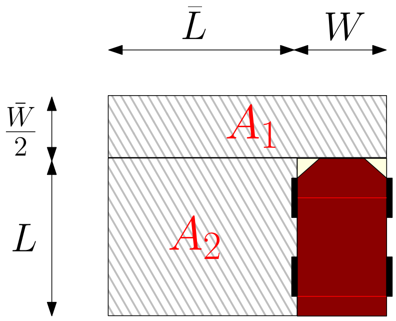


Figure 7.2: Example of how the collision box could be defined for detection of vehicles approaching from the right.

ego vehicle is performing the left turn. To tackle such scenarios, modifications on the modelling need to be performed in order to account for the rotational reference system. Identically, the collision definition should also be reformulated accordingly.

Additionally, increasing the length of the prediction horizon where a collision can be accurately detected would not only make it possible to introduce warnings and new intervention approaches, as discussed in Section 7.1.1. It would also make the system usable in other scenarios, such as collisions related to lane overtaking.

7.3 Future work

All improvements mentioned in this chapter should of course be taken into consideration, but some of them should be prioritized.

- Implementation of knowledge of the road geometry.
- Incorporating the relative heading as the third variable in collision detection, to make the estimation of the time to collision more precise and more strict to specific types of collisions.
- Upgrade of the vehicle model to include the rotational reference frame to the vehicle model to potentially enable collision avoidance when the ego vehicle is turning.

These improvement would solve the mayor short comings in the system design and would improve the potential of the system with respect to performance and system applications.

However, before continuing with development, some time should be spent on further studies of the proposed system. Finding the answer to how robust the system is, to e.g. sensor noise, is vital to know how the system would perform if further tests of implementations in a real vehicle would be pursued. This is something that was considered during the project, but due to time limitations and difficulties running the needed simulations in a stable way the results were inconclusive. The system was also planned to be further tested in a real vehicle, but due to technical problems, the attempted experiments failed. The functionality test, presented in Section 6.4, was performed in a real vehicle, but it was solely meant as a proof of concept and to verify that it was feasible to run the system in real-time on a vehicle computer. Successfully completing these studies would give a great insight to how usable this system would be as a commercial system. Hence, these studies should be considered top priority in any future work related to the findings presented in this thesis.

7.4 Concluding remarks

The result and analysis presented Chapter 6 shows the capabilities of the proposed system. While the discussion in the previous sections highlights a few shortcomings, the proposed system still fulfils many of the objectives for this thesis. There are still improvements to be done, as stated in Section 7.3, but the proposed sub-systems presented in chapter 3, 4 and 5 do, in the authors opinion, fulfil the first three objectives:

1. Track and predict the paths of the surrounding traffic providing a probabilistic, time dependent information of the vehicles positions.
2. Develop an efficient way to detect the risk of a collision by using the acquired information.
3. Assess in a formal and robust way the need for an intervention at the current time instance to minimize the number of unnecessary interventions.

The fourth objective, stating that the CA system should be implemented in a vehicle, has been partially fulfilled. The result of the functionality test presented in Section 6.4 does only verify, that it is feasible to run the system in real-time on a vehicle computer. Since this objective was only partially fulfilled, it was assigned top priority in discussion regarding future work, Section 7.3, and it should be mentioned that further test of the implementation is currently being performed at Denso Sales Sweden.

References

[1] Transportstyrelsen. (2014, May) Årsstatistik: Dödade och svårt skadade efter olyckstyp. [Online]. Available: http://www.transportstyrelsen.se/sv/Press/Statistik/Vag/Olycksstatistik-gammal/Olycksstatistik-vag/Nationell-statistik1/Arsvis-statistik/Historik-olyckstyp/

[2] G. de Campos, A. H. Runarsson, and F. Granum, "Collision avoidance at intersections: A probabilistic threat-assessment and decision-making system for safety interventions," in *Intelligent Transportation Systems (ITSC), 16th International IEEE Conference on*, Oct 2013.

[3] H. Godthelp, "Vehicle control during curve driving," *Human Factors The Journal of the Human Factors and Ergonomics Society*, vol. 28, no. 2, pp. 211–221, 1986.

[4] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using monte carlo sampling," *Intelligent Transportation Systems (ITS), IEEE Transactions on*, vol. 9, no. 1, pp. 137–147, March 2008.

[5] F. Rohrmuller, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic mapping of dynamic obstacles using markov chains for replanning in dynamic environments," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, Sept 2008, pp. 2504–2510.

[6] D. Petrich, T. Dang, D. Kasper, G. Breuel, and C. Stiller, "Map-based long term motion prediction for vehicles in traffic environments," in *Intelligent Transportation Systems (ITSC), 16th International IEEE Conference on*, Oct 2013, pp. 2166–2172.

[7] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, March 1960.

[8] M. Nasri and W. Kinsner, "Extended and unscented kalman filters for the identification of uncertainties in a process," in *Cognitive Informatics Cognitive Computing (ICCI*CC), 12th IEEE International Conference on*, July 2013, pp. 182–188.

[9] S. J. Julier and J. K. Uhlmann, “A new extension of the kalman filter to nonlinear systems,” in *Proce. SPIE*, vol. 3068, no. 26, July 1997, pp. 182–193.

[10] M. Tsogas, A. Polychronopoulos, and A. Amditis, “Unscented kalman filter design for curvilinear motion models suitable for automotive safety applications,” in *Information Fusion, 2005 8th International Conference on*, vol. 2, July 2005, pp. 1295–1302.

[11] S. Panzieri, F. Pascucci, and R. Setola, “Simultaneous localization and map building algorithm for real-time applications,” in *Proc. of 16th IFAC World Congress*, July 2005.

[12] S. Särkkä, *Bayesian filtering and smoothing.* Cambridge University Press.

[13] G. H. Golub and C. F. Van Loan, *Matrix computations.* JHU Press, 2012, vol. 3.

[14] B. Teixeira, L. Torres, L. Aguirre, and D. Bernstein, “Unscented filtering for interval-constrained nonlinear systems,” in *Decision and Control (CDC), 47th IEEE Conference on*, Dec 2008, pp. 5116–5121.

[15] S. Kim, S. Lee, I. Yoon, M. Yoon, and D.-H. Kim, “The vehicle collision warning system based on gps,” in *Computers, Networks, Systems and Industrial Engineering (CNSI), First ACIS/JNU International Conference on.* IEEE, 2011, pp. 263–264.

[16] Y. Wang, E. Wenjuan, D. Tian, G. Lu, G. Yu, and Y. Wang, “Vehicle collision warning system and collision detection algorithm based on vehicle infrastructure integration,” in *Advanced Forum on Transportation of China (AFTC)*, Oct 2011, pp. 216–220.

[17] G. Weidl, G. Breuel, and V. Singhal, “Collision risk prediction and warning at road intersections using an object oriented bayesian network,” in *Proceedings of 5th International Conference Automotive-UI13.* ACM, 2013, pp. 270–277.

[18] M. Althoff, O. Stursberg, and M. Buss, “Model-based probabilistic collision detection in autonomous driving,” *Intelligent Transportation Systems (ITS), IEEE Transactions on*, vol. 10, no. 2, pp. 299–310, June 2009.

[19] Z. Drezner, “Computation of the Bivariate Normal Integral,” *Mathematics of Computation*, vol. 32, no. 141, pp. 277–279, 1978.

[20] M. Brannstrom, E. Coelingh, and J. Sjoberg, “Model-based threat assessment for avoiding arbitrary vehicle collisions,” *Intelligent Transportation Systems (ITS), IEEE Transactions on*, vol. 11, no. 3, pp. 658–669, Sept 2010.

[21] N. Kaempchen, B. Schiele, and K. Dietmayer, “Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios,” *Intelligent Transportation Systems (ITS), IEEE Transactions on*, vol. 10, no. 4, pp. 678–687, Dec 2009.

[22] G. de Campos, P. Falcone, and J. Sjoberg, "Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing," in *Intelligent Transportation Systems (ITSC), 17th International IEEE Conference on*, Oct 2014.

[23] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for linear and hybrid systems*, 2012.

[24] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/~mpt.

[25] MATLAB, *version 8.2.0.701 (R2013b)*. Natick, Massachusetts: MathWorks Inc., 2014.

[26] PreScan, *version 6.8.1*. Helmond, Netherlands: TASS International, 2013.

[27] Trafikverket, VGU 2004:80, "Vägars och gators utformning. Begrepp och grundvärden." *Borlänge. Trafikverket.*, pp. 57–186, 2004.

[28] M. Reed, "Intersection kinematics: A pilot study of driver turning behavior with application to pedestrian obscuration by a-pillars," 2008.

[29] G. Jochmann, S. Kerner, S. Tasse, and O. Urbann, "Efficient multi-hypotheses unscented kalman filtering for robust localization," in *RoboCup 2011: Robot Soccer World Cup XV*. Springer, 2012, pp. 222–233.

[30] D. Petrich, T. Dang, G. Breuel, and C. Stiller, "Assessing map-based maneuver hypotheses using probabilistic methods and evidence theory," in *Intelligent Transportation Systems (ITS), 2014 IEEE 17th International Conference on*, Oct 2014, pp. 995–1002.

[31] C. Grover, I. Knight, F. Okoro, I. Simmons, G. Couper, P. Massie, and B. Smith, "Automated emergency braking systems: Technical requirements, costs and benefits," TRL Limited, Tech. Rep., April 2008.

A

Traffic scenario descriptions

A.1 Equipment

Two model year 2010 Volvo S60 T6 vehicles provided by Denso Sales Sweden were used as the ego and other vehicle. Both vehicles were equipped with the following sensors

- Two inertial measurement and DGPS modules of model OXTS RT2002, one in each vehicle.
- One set of Vehicle-2-Vehicle (V2V) communications modules of model OXTS RT-Range.
- Two AutoBox car PCs, one in each vehicle.

and during logging a portable GNSS Base Station was setup to provide more accurate DGPS signals.

A.2 Real data scenarios

The traffic scenarios included multiple turning behaviours and driving behaviours likely found close to intersections in an urban environment. All scenarios were performed in a fictional intersection constructed for this specific case. The intersection was designed in accordance with [27]. An illustration of the intersection used is shown is Figure B.1 where the intersection was defined in accordance with [27].

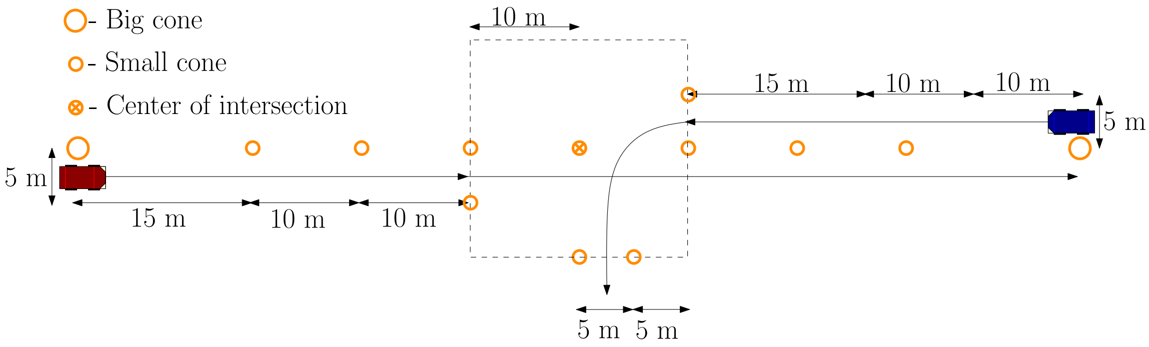


Figure A.1: Arena used during data collection in intersections.

Scenarios 4 and 5 are used for calibration and to gain knowledge if there is some calibration needed of the logged data. Scenarios 1-3 are used for actual testing of the system. Each scenario was done multiple times in case there was some badly logged data.

A.2.1 Scenario 1 - Left turn across path

The ego vehicle 45m from the intersection and the other vehicle starts at 25m. This allowed the other vehicle to complete the left turn while the ego vehicle still had 10m to reach the intersection. This offset is then later removed to simulate a collision.

A.2.2 Scenario 2 - Abandoned turn

The other vehicle performs a left turn, but stops before entering the ego vehicle's lane, allowing it to pass. No offset is used here since none is needed so the vehicles both start at 45m from the intersection.

A.2.3 Scenario 3 - Avoidance manoeuvre

Both vehicles start at 45m away from the intersection. The driver of the other vehicle performs an avoidance manoeuvre in his own lane at 10m away from the ego vehicle.

A.2.4 Scenario 4 - Straight

Both vehicles start at the 45m mark and driver towards each other at a steady speed. No specific manoeuvres were performed.

A.2.5 Scenario 5 - Stationary

The ego vehicle is stationary at different distances from the intersection while the other vehicle performs a left turn.

A.2.6 Details about the scenarios

Table A.1: Specification for the second part of the experiment.

Scenario	Parameters	Measurements	No. of iter.
Scenario 1	$v_1 = 30$ km/h $v_2 = 25$ km/h v_3[4]$= 30$ km/h offset[5] $= 10$ m	Ego and other GPS Position Dist. to Other Vehicle Ego and Other Speed Ego and Other Acceleration Ego and Other Heading Ego and Other Angular Rate	
Scenario 2	$v_1 = 30$ km/h $v_2 = 0$ km/h $v_3 = 30$ km/h	——— ” ———	5
Scenario 3	$v_1 = 30$ km/h $v_3 = 30$ km/h	——— ” ———	5
Scenario 4	$v_1 = 30$ km/h $v_3 = 30$ km/h	——— ” ———	2
Scenario 5	d[1]$= 10, 30$ m v_1[2]$= 30$ km/h v_2[3]$= 25$ km/h	——— ” ———	

A.3 PreScan scenarios

The scenarios introduced here are meant to test driving behaviours that would lead to a potential collision in an intersection and scenarios that could trigger a false intervention. What will be investigated are situations where the driver of the other vehicle underestimates the speed of the ego vehicle, dangerous turning manoeuvres, avoidance manoeuvres and a regular left turn where the other vehicle gives priority to the ego vehicle.

[1] d is referring to the distance from the intersection that the ego vehicle was stationary at.

[2] v_1 is speed of the other vehicle before the turn.

[3] v_2 is speed of the other vehicle through the turn.

[4] v_3 is speed of the ego vehicle.

[5] Offset is referring to longitudinal offset when going though the intersection.

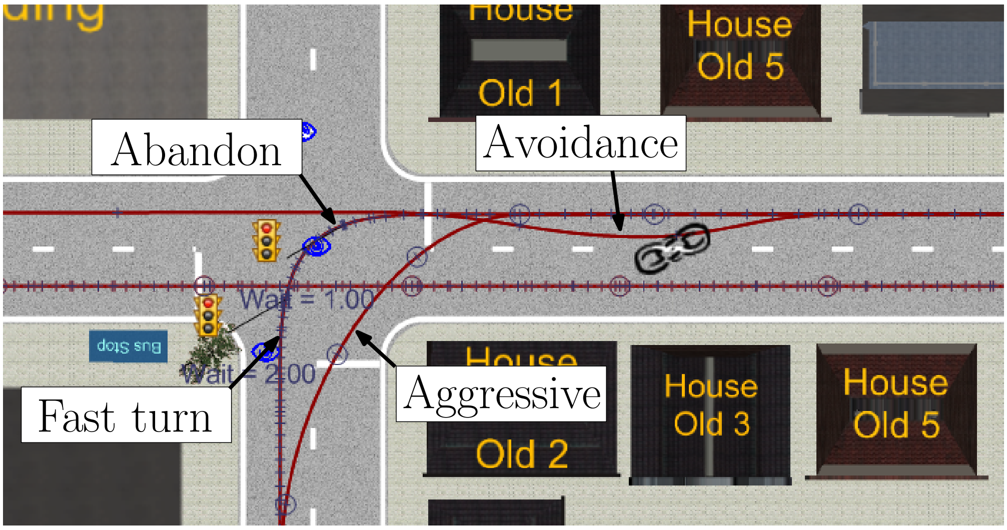


Figure A.2: Screen shot of the PreScan implementation. Showing how the paths look like in the intersection.

A.3.1 Fast left turn

In this scenario, the ego vehicle travels at a steady $13^m/_s$ when the other vehicle makes a left turn. This scenario should illustrate a turn where the driver of the other vehicle slows down to do a turning manoeuvre. After slowing down the "driver" deems that he has enough time to make the turn, as if he underestimates the speed of the ego vehicle, and accelerates again in the turn to try and finish the turn before the ego vehicle gets to the intersection. The control signals are shown in Figure A.3.

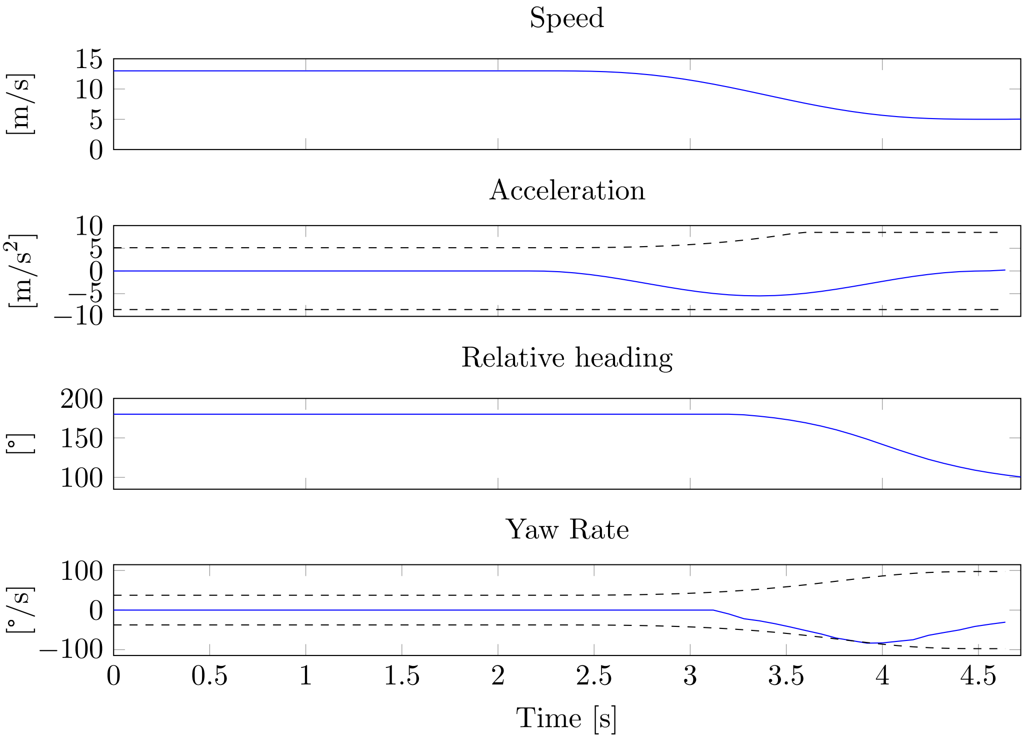


Figure A.3: Showing the different signals of the other vehicle during the simulation up to the time when the ego vehicle crashes into it. The dashed lines in the acceleration and yaw rate show the constraints that are assumed in Section 2.4.6.

A.3.2 Aggressive left turn

In this scenario, the ego vehicle travels at a steady 13$^{m}/_{s}$ when the other vehicle makes an aggressive left turn. By aggressive it assumed that the other car cuts the turn in order to gain more time to complete the manoeuvre. The control signals are shown in Figure A.4.

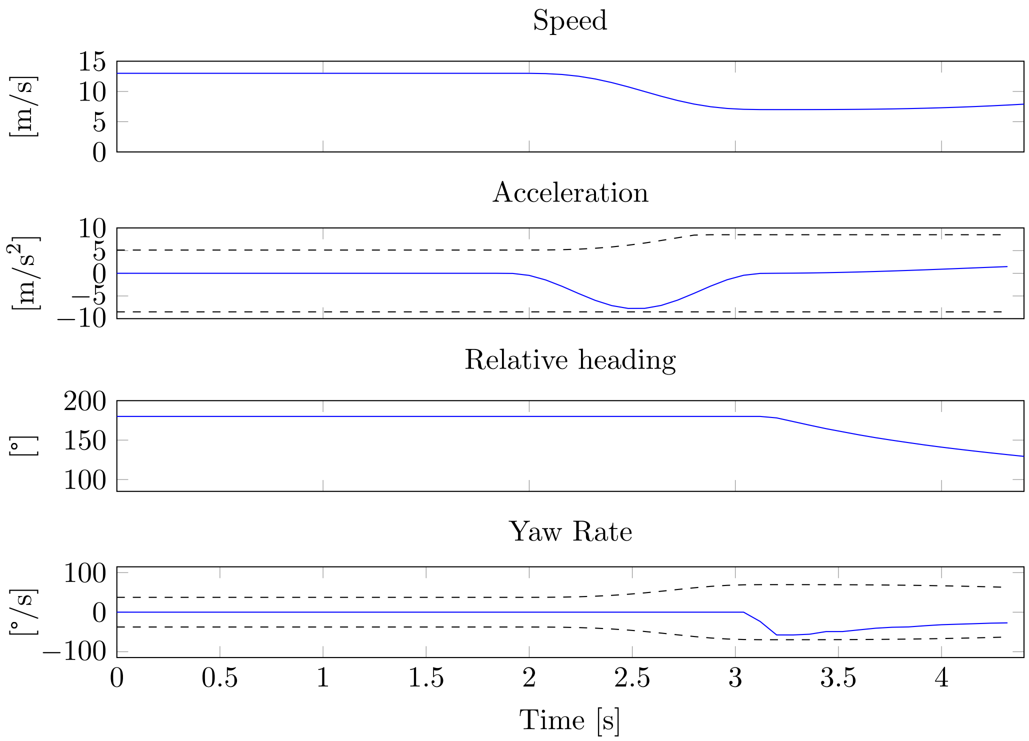


Figure A.4: Showing the different signals of the other vehicle during the simulation up to the time when the ego vehicle crashes into it. The dashed lines in the acceleration and yaw rate show the constraints that are assumed in Section 2.4.6.

A.3.3 Avoidance manoeuvre

As before, the ego vehicle travels at 13$^m/_s$ The other vehicle makes an avoidance manoeuvre in his own lane at such a time the ego vehicle passes it when he is the closest to the ego vehicle. The control signals are shown in Figure A.5.

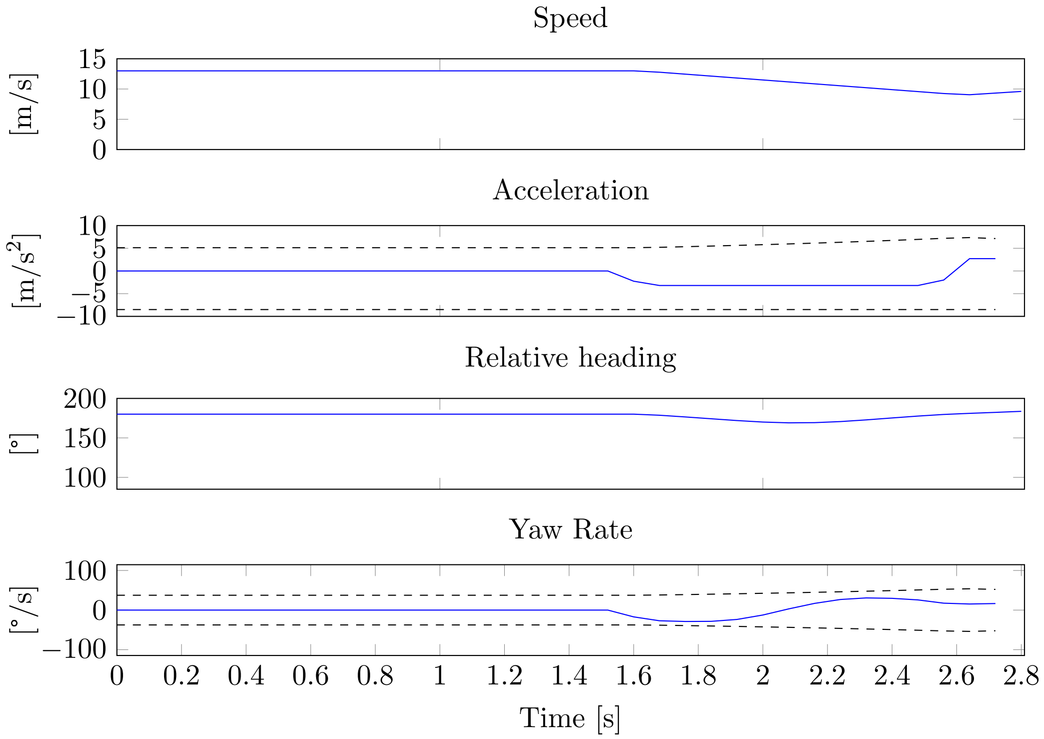


Figure A.5: Showing the different signals of the other vehicle during the simulation up to the time when the ego vehicle crashes into it. The dashed lines in the acceleration and yaw rate show the constraints that are assumed in Section 2.4.6.

A.3.4 Abandon turn

Lastly, as before, the ego vehicle travels at 13$^m/_s$ The other vehicle starts performing a left turn but gives priority to the ego vehicle and waits for it to pass before continuing. The control signals are shown in Figure A.6.

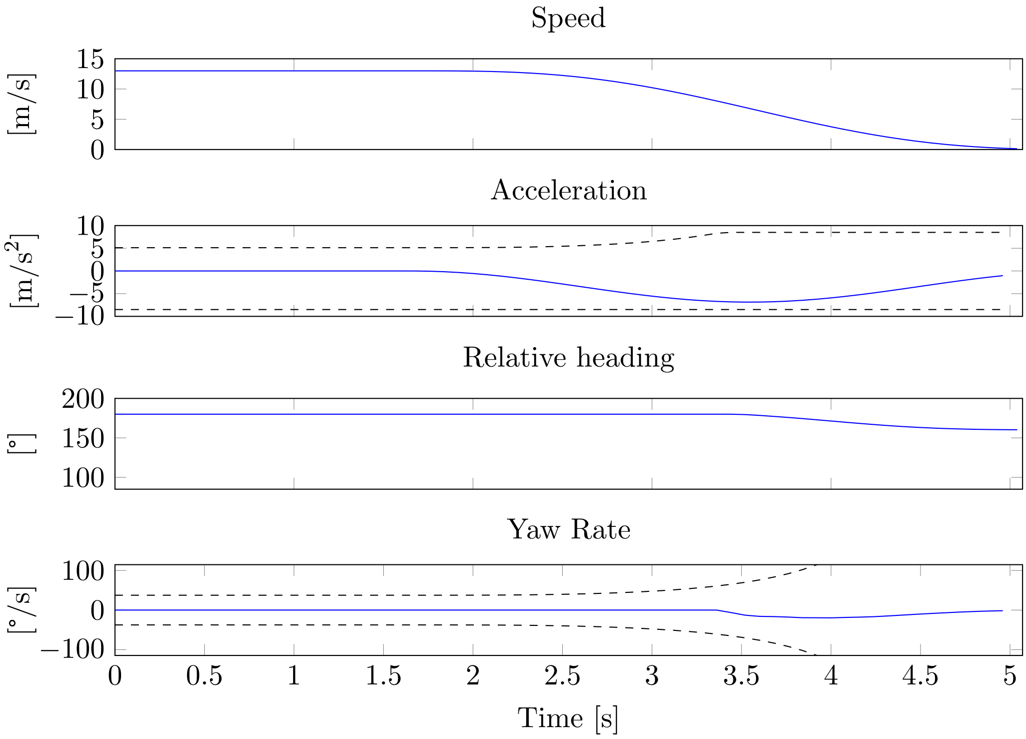


Figure A.6: Showing the different signals of the other vehicle during the simulation up to the time when the ego vehicle crashes into it. The dashed lines in the acceleration and yaw rate show the constraints that are assumed in Section 2.4.6.

B

Parameters

The motivation for setting the different parameters needed for the system is explained in this chapter.

B.1 Process noise

Using the measurement equipment described in Appendix A, a drive session of 10 minutes was logged. The driver was instructed not to perform any aggressive manoeuvres. The driving sessions was done in an urban environment consisting of multiple intersections and were performed at a maximum speed of 40 $^{km}/_{h}$.

From the logged acceleration and yaw rates the process noise is defined by taking the derivatives of the logged signals,

$$\dot{a}(t) = \frac{\partial a(t)}{\partial t}, \qquad \ddot{\theta}(t) = \frac{\partial \dot{\theta}(t)}{\partial t} \tag{B.1}$$

which are respectively known as jerk and yaw acceleration. From these calculated signals, the standard deviation is found by the following equation

$$\sigma^2 = \frac{\sum\limits_{i=1}^{n}(x_i - \mu)^2}{n}, \tag{B.2}$$

where n is the number of samples and μ is the mean of those samples.

Figure B.1 shows the first 6 minutes of the test and the interval the calculations were done on.

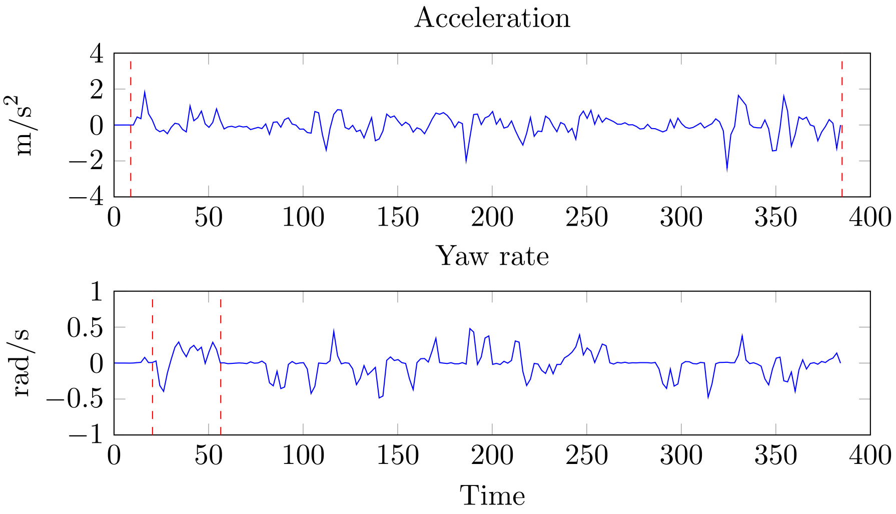


Figure B.1: Logged signal during urban driving and the intervals where the jerk/yaw acceleration where calculated from.

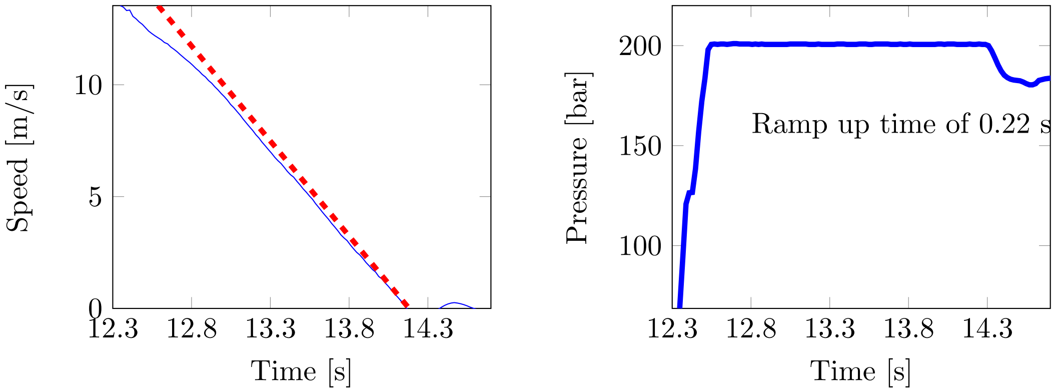
The interval for the yaw-rate needed to be chosen in a way to capture the yaw acceleration during a turning manoeuvre. The standard deviation for the chosen intervals where

- Jerk: 0.35 $^{m}/_{s^3}$
- Yaw Acceleration: 0.0135 $^{rad}/_{s^2}$

which are the values used for the prediction process matrix.

B.2 Deceleration and ramp up time

To evaluate the max deceleration and ramp up time of the test vehicle a series of experiments were performed. These experiments consisted of accelerating the vehicle to speeds of 20,40,50 $^{km}/_{h}$ and perform emergency braking. From this data the max deceleration and ramp up times were evaluated. Only result form the 50 $^{km}/_{h}$ experiment are shown here. In Figure B.2 the speed profile of the test vehicle and the ramp up time in the braking systems during one of our braking tests are shown. In [31] the ramp up time of a light vehicle was concluded to vary between 0.2 and 0.3 s. In the plot to the right in Figure B.2, the ramp up time compares well to the one suggested in [31]. The speed profile is compared with an estimated speed profile. This estimated speed profile is calculated by using a time delay of 0.2 s, to simulate the ramp up time suggested in [31], and then applying a constant deceleration of -8.5 m/s^2.



(a) The speed profile of the test vehicle and the ramp up time in the braking systems during a braking test from 50 kph. The estimated speed profile of the test vehicle is calculated by using a time delay of 0.2 s and then applying a constant deceleration of -8.5 m/s^2.

(b) The speed profile of the test vehicle and the ramp up time in the braking systems during a braking test from 50 kph. The estimated speed profile of the test vehicle is calculated by using a time delay of 0.2 s and then applying a constant deceleration of -8.5 m/s^2.

Figure B.2: The speed profile of the test vehicle and the ramp up time in the braking systems during a braking test from 50 kph. The estimated speed profile of the test vehicle is calculated by using a time delay of 0.2 s and then applying a constant deceleration of -8.5 m/s^2.

C

Thesis developement

For evaluation purposes, a rough summary of the thesis developement is provided.

C.1 Summary

To tackle this project, we decided to split the problem into two parts. One that regards how to predict a collision and the other how to avoid it. After weeks of reading other related work it became clear that predicting a collision could get heavy, and as the goal was for the system to be able to be run in real time, it was decided that an appropriate method was to use the Unscented Kalman filter (UKF). Given the time frame of the project, we decided that it could get time consuming developing a collision avoidance method by turning the vehicle, and as such only braking was considered.

Following this, a few test scripts were developed, where the UKF was used to predict the movement of both created and logged data with acceptable results. At the same time, we got more acquainted with the set theory for deciding on when to start braking, where it became clear, that some manual labour was needed to allow the MPT [24] to work with Simulink.

When the prediction algorithm was working fine, it was time to develop the collision detection. That proved to be quite challenging task, as there do exist a number of procedures of doing this, but most of them were to heavy to be run in real time. After a while we decided on using a vector based collision detection and quick testing showed very good calculation time. Some manual labour was also required during this step for the calculation to be compiled on Simulink, since code generation did not support the built in Matlab functions needed for the calculations.

When all functions were separately working, they were put together and the system was tested using real data described in A. After some parameter tuning we moved onto using PreScan. This allowed us to better control the scenarios that was needed to be tested and provided us with an environment which needed a similar structure as in

AutoBox. During this step, it was decided to change the measurement equations in the model to non-linear ones to represent a more realistic sensor data and seemed not to affect the system in a noticeable way.

In PreScan we were able to test how the algorithm worked when the reference point of the measured vehicle is moving, as can happen with a radar. The system was able to avoid the collision in some cases, but as expected, it seemed that further study is needed to tackle this problem.

After the PreScan model was up and running, it was possible to move onto the AutoBox. Using ControlDesk to replay the previously logged data and feed it to the AutoBox, which simulates how the vehicle would have transmit the data, we were able to test the capability of the system to run in real time. During these tests it was clear that the intervention timings were correct as well as the system was quite capable of running in real time.

The next step was to do a run using the vehicles, with offsets and no active braking interventions. Upon some on-site calibrations on the DGPS two calibration runs were made. Both of which where the other vehicle is stationary and the ego vehicle is moving. In both runs a collision was detected, although a bit later than expected which would have resulted in a collision. Never the less, the system was functioning and we were ready to move onto a moving test scenario. At this point, one of the vehicles overheated and we were unable to continue the experiment. The experiment was conducted very late in the project, and as time was running out, it was decided to focus on writing the report and scrap the real world testing for now.

The project was started 17. January and was more or less finished 25. June whereas during the summer, some extra work and polishing was made on the thesis and a conference paper was written and sent to Intelligent Transportation Systems Conference (ITSC). On 14. August it was announced that the conference paper was accepted.