CHALMERS
UNIVERSITY OF TECHNOLOGY

Intelligent transportation systems

A collision avoidance system incorporating probabilistic threat-assessment and decision-making protocols

Master's Thesis within the division of Systems, Control and Mechatronics

CHARLIE SJÖDIN
ISABELLA JOHANSSON

Department of Signals and Systems (S2)
CHALMERS UNIVERSITY OF TECHNOLOGY, Gothenburg, Sweden
EX075/2015

Master's thesis 2015:01

Intelligent transportation systems

A collision avoidance system incorporating probabilistic
threat-assessment and decision-making protocols

CHARLIE SJÖDIN
ISABELLA JOHANSSON

Department of Signal and Systems (S2)
Chalmers University of Technology
Gothenburg, Sweden 2015

Intelligent transportation systems
A collision avoidance system incorporating probabilistic threat-assessment and decision-making protocols
CHARLIE SJÖDIN
ISABELLA JOHANSSON

Supervisor: Gabriel Rodrigues de Campos, Department of Signals and Systems; Klas Alenljung, DENSO Sales Sweden AB
Examiner: Paolo Falcone, Department of Signals and Systems

Master's Thesis 2015:01
Department of Signals and Systems (S2)
Division of Systems, Control and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Intelligent transportation systems
A collision avoidance system incorporating probabilistic threat-assessment and decision-making protocols
CHARLIE SJÖDIN
ISABELLA JOHANSSON
Department of Signals and Systems (S2)
Chalmers University of Technology

Abstract

The scope of this thesis has been on developing a Collision Avoidance (CA) system being able to cope with the accident-prone urban intersections. These types of complex sections within the road network will have the need to account for both the different road-users but also the topology. This thesis will however constrain the problem to only consider the EGO vehicle and one Perceived Other Vehicle (POV), hence limiting the scope.

The proposed CA-system is divided into three modular blocks for the ability of individual future developments. The first block captures the states and predicts the future paths of the accounted vehicles with the aid of the observed road geometry. Each vehicle's state is computed using an *Interacting Multiple Model* (IMM) filter that matches a motion model to the probable manoeuvre, whereas the topology is observed and accounted for by a *Bayesian Network* (BN). The information given from both the IMM and BN are combined for the ability to accurately predict, through an *Unscented Kalman Filter* (UKF), the future behaviour of each vehicle. At each prediction step, a probability of an imminent collision will, by the second block, be calculated using the joint cumulative distribution. If this probability reaches above a certain threshold, the final and third block will be invoked to evaluate the detected threat for the need of a collision avoidance intervention. The evaluation will be made through a formal threat assessment method based on reachability tools, with the aim of finding the *Point-of-No-Return* stating which point in time the vehicle inevitable ends up in a collision. The only considered intervention method is by the use of emergency braking (AEB) to be able to keep the driver in the loop as the decision maker as long as possible.

The developed CA-system was evaluated through several different scenarios in an urban intersection spanning the possible configurations that could occur. The unfolded result revealed a robust system being able to sufficiently predict the future paths of both accounted vehicles for the ability to detect a probable collision. When the detected collision was evaluated to be unavoidable by the driver, the collision avoidance system triggered an emergency brake intervention being able to prevent or at least mitigate the collision.

Keywords: Threat Assessment, Decision-making, Collision Avoidance, Active Safety, Bayesian Network, Interactive Multiple Model Filtering, Unscented Kalman Filter, Driver Intention, Urban Intersections.

Acknowledgements

We would like to sincerely thank our supervisor at Chalmers University of Technology, Gabriel Rodrigues de Campos and our supervisor at DENSO Sales Sweden AB, Klas Alenljung for their guidance and support throughout the project.

We would also like to give a special thanks to both Johan Degerman and Tommy Johansson at DENSO for their support and passionate involvement. Our examiner Paolo Falcone also deserves our gratitude for the guidance and helpful meetings. We are also thankful for the support we have got from DENSO's active safety department at whole for their helpfulness, support and making us feel welcome.

A thanks is also directed to Diogo Silva for the support with mathematical formulations and feedback on the report. Our opponents Herman Fransson, Malin Karlsson and Tomas Nilsson also deserve to be acknowledged for their helpful review of our work.

Finally, we would like to thank our family and friends for the support both through the good times and the bad. Because folks, this has been an adventure that we will never forget.

Charlie Sjödin and Isabella Johansson, Gothenburg, December 2015

Contents

Acronyms

AEB - Autonomous Emergency Braking
BN - Bayesian Network
CA - Collision Avoidance
CAcc - Constant Acceleration (model)
BND - Bivariate Normal Distribution
CCA - Constant Curvature and Acceleration (model)
CPT - Conditional Probability Table
CT - Constant Turn-rate (model)
CTRA - Constant Turn Rate and Acceleration (model)
CTRV - Constant Turn Rate and Velocity (model)
CV - Constant Velocity (model)
DAG - Directed Acyclic Graph
DTC - Distance To Collision
DM - Decision Making (algorithm)
EGO - Ego vehicle (the vehicle equipped with the collision avoidance system)
EKF - Extended Kalman Filter
IMM - Interacting Multiple Model
JCDF - Joint Cumulative Distribution Function
LTAP - Left Turn Across Path
PNR - Point of No Return
POV - Perceived Other Vehicle (all vehicles except the EGO)
UKF - Unscented Kalman Filter
TA - Threat Assessment
TTC - Time To Collision
ΔTTC - Collision Time Interval
V2I - Vehicle-to-Infrastructure (communication)
V2V - Vehicle-to-Vehicle (communication)
V2X - Vehicle-to-X (V2V, V2I etc.)

Variable Notations

Notation	Description	Definition
$\mathbf{x}_{EGO}, \mathbf{x}_{POV}$	State vector EGO resp. POV	Eq. 2.1
ΔT	Sample time	
k	Discrete time step	
P_{Left}	Probability of going left	Eq. 3.13
P_{Right}	Probability of going right	Eq. 3.14
$P_{Straight}$	Probability of going straight	Eq. 3.15
Π	Transition matrix	Eq. 3.16
mo	Indicated motion model	Sec. 3.3
N	No. of models used	Sec. 3.3
x	Motion model	Eq. 3.17
$\hat{z}$	Measurement model	Eq. 3.17
z	Observed measurement	Sec. 3.3
q	Process noise	Eq. 3.17
ε	Measurement noise	Eq. 3.17
f_{mo}	Motion function dependent on model	Eq. 3.17
h_{mo}	Measurement function dependent on model	Eq. 3.17
$A_{mo}, B_{mo}, C_{mo}, D_{mo}$	Matrices dependent on model	Eq. 3.18
Q_{mo}	Process noise covariance matrix	Sec. 3.3
R	Measurement noise covariance matrix	Sec. 3.3
$\bar{\mu}$	Previous mixing probability	Sec. 3.3
$\hat{\mu}$	Updated mixing probability	Sec. 3.3
$\bar{x}$	Previous state	
$\hat{x}$	Updated state	
$\check{x}$	IMM state output	
$\bar{P}$	Previous covariance	
$\hat{P}$	Updated covariance	
$\check{P}$	IMM covariance output	
S	Innovation covariance	Eq. 3.25
K	Kalman gain	Eq. 3.26
Λ	Likelihood of each updated filter model	Eq. 3.29
i	Prediction time steps	
$n_{predictions}$	No. of prediction steps	
χ	Sigma points	Sec. 3.4

Notation	Description	Definition
n	No. of states	Sec. 3.4
$\tilde{\mathbf{x}}$	Augmented state mean	Sec. 3.4
$\tilde{\mathbf{P}}$	Augmented state covariance	Sec. 3.4
Q_{pred}	Process noise covariance matrix in UKF	Sec. 3.4
λ	Scaling parameter	Eq. 3.36
α, κ	Parameters determining the spread of χ	Sec. 3.4
β	Prior information parameter	Sec. 3.4
W	Weights in UKF	Sec. 3.4
$v_{predictions}$	Predicted velocity	
l, w	Length and width of EGO	Chap. 4
$\bar{l}, \bar{w}$	Length and width of POV	Chap. 4
η	No. of points spread out over vehicles	Chap. 4
j	Point used	Chap. 4
$\mathcal{P}_{collision}$	Probability of collision	Eq. 4.7
ϕ	Joint density function	Eq. 4.9
D_{ms}	Squared Mahalanobis distance	Eq. 4.10
m_E	Mean of estimated position EGO	Chap. 4
Σ_E	Covariance of estimated position EGO	Chap. 4
m_O	Mean of estimated position POV	Chap. 4
Σ_O	Covariance of estimated position POV	Chap. 4
m_{Ep^j}	Mean of point j for the EGO	Chap. 4
m_{Op^j}	Mean of point j for the POV	Chap. 4
t_C	Time to collision	
Δt_C	Collision time interval	
d_C	Distance to collision	
$\sigma^2_{d_C}$	DTC variance	
$Danger$	Boolean value for the collision risk	
T_{CD}	Threshold for collision detection	
t_{stop}	Stop time (reaction time + brake time)	
v_{max}	Maximum allowable velocity	Chap. 5
x_{CA}	State for CA	Eq. 5.5
u	Control input	Chap. 5
$\mathcal{S}$	Target set	Eq. 5.6
$\mathcal{X}$	Constraints on state	Eq. 5.7
$\mathcal{U}$	Constraints on input	Eq. 5.8
$-a_{max}$	Maximum deceleration	Chap. 5
d_{max}	Maximum travel distance	Chap. 5
$\mathcal{A}$	Attraction set	Eq. 5.9
L	Last attraction set	Chap. 5
t_{limit}	Time limit to avoid collision	Chap. 5
$\hat{d}_C$	Estimated DTC	Eq. 5.13
$\mathcal{P}_{Belong}$	Prob. of belonging to $\mathcal{A}$	Eq. 5.14
p	No. of points uniformly distributed	Chap. 5
T_{CA}	Threshold for collision avoidance	
$Brake$	Brake signal	Chap. 5

List of Figures

1
Introduction

Development within the automotive industry has in recent years had significant progress in the area of active safety for passenger cars. Yet one challenging problem still needing attention is collision avoidance at intersections. For new intelligent safety systems focus on collision avoidance at intersections is therefore of importance, since road intersections are among the most accident-prone and complex sections within the road network. This compels Advanced Driver Assistance Systems (ADAS) the need to cope with highly complex road scenarios, such as urban intersections, where the hard-to-predict road-users and surrounding environment need to be accounted for. This thesis will propose a collision avoidance (CA) system with the ability to deal with these accident-prone urban intersections.

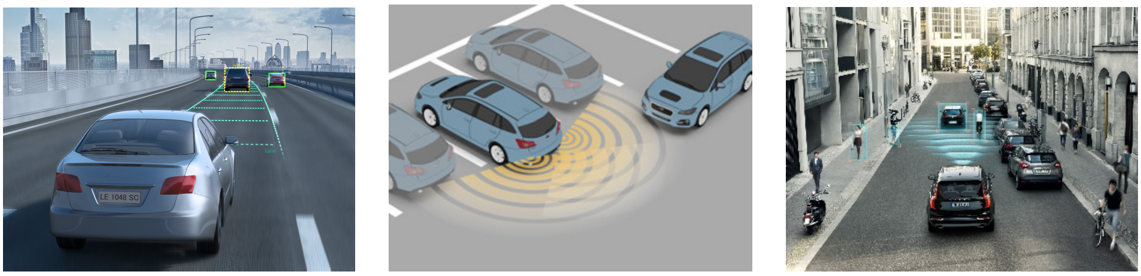
1.1 Background

The automotive industry is an area that is in constant development, where Intelligent Transport Systems (ITS) is a key factor to increase safety and to tackle congestion and growing emission. With the use of ITS, transport can get safer, more efficient and more sustainable [1].

Today, there are several research projects within the automotive industry with focus on developing fully automated vehicles to be put on the roads in a near future. Extensive research has been conducted within Vehicle-to-Vehicle (V2V) [2][3] and Vehicle-to-Infrastructure (V2I) [4][5] communication, but as it will take several decades before the traffic network is fully utilised by automated vehicles, the need for a standalone automatisation solution is necessary. Even though substantial research has been conducted for standalone automatisation such as lane keeping [6], adaptive cruise control [7] and self-driving cars [8] as illustrated in Figure 1.1, coping with all parts of the complex road network has not yet been fully addressed.

An especially accident-prone part of the complex traffic network is urban intersections as there are several different road-users co-existing in the same road section, all having multiple choices of direction. Road safety statistics for Europe [9][10][11] discloses that around 20% of the fatalities and around 43% of the overall accidents are at intersections, which in fact has been the trend for over a decade.

The first problem encountered in intersection scenarios, both for humans as well as machines, is to determine the paths of the other road users. The path prediction is needed to evaluate if and where the paths of the vehicles will intersect.



(a) Lane keeping system from Toshiba [12]

(b) Rear breaking system by Subaru [13]

(c) City safety system from Volvo Cars [14]

Figure 1.1: Research within active safety functions for the three different manufacturers Toshiba, Subaru and Volvo Cars.

If the paths are evaluated to intersect at the same time step, then consequently a collision will occur. In theory this seems to be trivial, but in reality is the time span from a collision is detected to when the collision actually occurs often very short. A human is affected by both reaction and decision latency as well as distractions, which also adds a time delay when evaluating the collision risk. A machine on the other hand cannot get distracted, but can nevertheless have uncertainties in the sensor measurements.

The safest approach would be to stop the vehicle as soon as there is a risk of collision, but for a machine to gain acceptance among drivers as a tool to avoid collisions, the system will need to give very few false interventions. If and when to initiate a braking intervention is therefore of importance.

1.2 Purpose

In this thesis, the goal is to develop and demonstrate formal decision-making and threat-assessment algorithms with a particular attention to complex traffic intersection scenarios. Given an unknown complex intersection, the objective is to provide a decision-making protocol preventing/mitigating a potential collision. Moreover should formal analysis methods, verifying safety requirements, be provided for novel intelligent control strategies.

This Master thesis project will, in collaboration between DENSO Sales Sweden AB and Chalmers University of Technology, develop a Collision Avoidance (CA) system. The purpose is to develop a system dealing with a long-term path prediction for the included vehicles, a collision risk detection and finally a collision avoidance strategy as later illustrated in Figure 1.2. The proposed system shall be kept modular for the possibility to improve individual parts of the system in the future.

1.3 General objectives

In accordance with the background and purpose described in the previous sections, the general objectives for this thesis are stated as:

- Proposing a method to merge pertinent information of the surrounding environment in order to accurately predict the behaviour of oncoming traffic.
- Developing a stochastic solution evaluating the risk of collision using the previous mentioned predictions.
- Defining a formal, robust decision making procedure for least-invasive braking interventions.
- Evaluating the proposed Collision Avoidance (CA) system with both simulations as well as with real-time implementation.

1.4 Contributions

The contributions of this thesis are:

- A combination of a Bayesian Network (BN) and an Interacting Multiple Model (IMM) filter together with an Unscented Kalman filter (UKF) to be able to compute long-term path predictions of vehicles.
- A novel way of defining a collision, resulting in the ability to detect a big variety of possible collision configurations. The method is probabilistic, i.e. calculates the probability of collision.
- A CA system being able to predict and avoid/mitigate several different collision configurations in intersection scenarios.

The content of this thesis also serves as a basis for a technical article which will be submitted in the nearby future.

1.5 System overview of the proposed solution

A graphical overview of the approach of the presented system is illustrated in Figure 1.2. Here, a *Long-term path prediction* is executed first which henceforth progresses into a *Collision detection* assessment, and finally continues into a *Collision avoidance* procedure. A flow chart of the information distributed through the system is shown in Figure 1.3. The figure illustrates how map and sensor data serves as inputs to the Bayesian Network (BN), where the result from the BN is transferred, together with the sensor data, to the IMM. The combined result from the BN and IMM then provides a decision of probable vehicle motion to the Unscented Kalman Filter (UKF), which finally provides a trajectory propagation along the whole prediction horizon. All of these cooperative functions are referred to as the long-term path prediction block. The predictions are thereafter sent to the collision detection block for evaluation of probable collision and finally sent to the collision avoidance block for an intervention decision. The respective blocks are described in Chapter 3, 4 and 5.

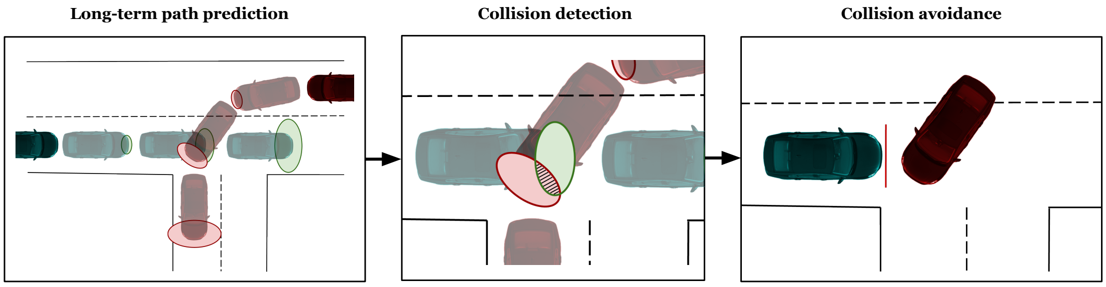


Figure 1.2: Block scheme showing a graphical overview of the total system with the three different objectives

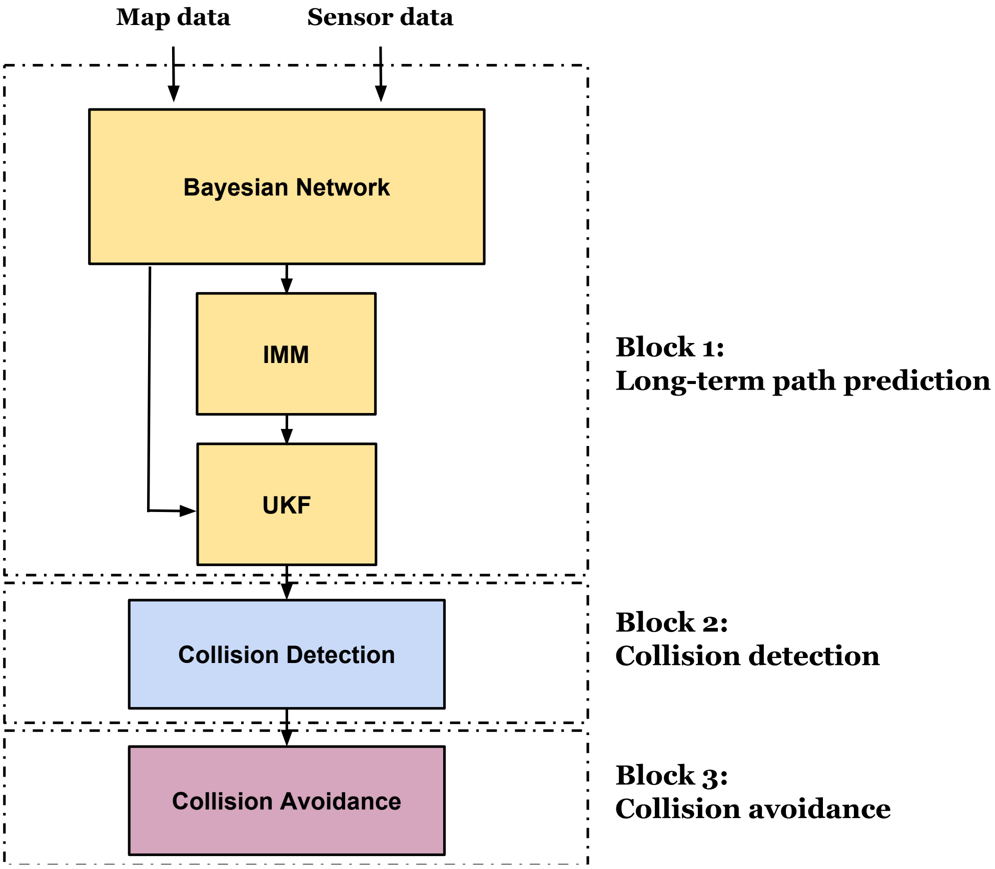


Figure 1.3: Block scheme showing a flow of information throughout the total system with the three different objectives.

1.6 Thesis organisation

Chapter 2 provides the objectives, assumptions, defined models and other prerequisites necessary for the reader to get an underlying understanding of the system design. The focus of Chapter 3 lies at explaining the theory and design of how to track and predict the states of both the Perceived Other Vehicle (POV) as well as the EGO vehicle. Chapter 4 defines the risk of collision, evaluated based on the probabilistic predictions from Chapter 3. Chapter 5 are thereafter outlining the threat assessment, based on the probability of collision obtained in Chapter 4.

In Chapter 6 are the results from the long-term path prediction, Chapter 3, collision detection, Chapter 4, and collision avoidance, Chapter 5, subsystems evaluated both separately, as well as combined into the complete CA system. A discussion of the obtained results and concluding remarks with possible future developments is presented in Chapter 7.

2

Preliminaries

This chapter describes the prerequisites, defined problems and preliminary work prior to the research conducted in regard to the first three objectives presented in Section 1.3. Section 2.1 provides a problem description, further explaining the general objectives from Section 1.3, whereas the made assumptions will be outlined in Section 2.2. This chapter is also giving the vehicle and environment representations, in Section 2.3 and 2.4 respectively. Finally is the intended evaluation setup described in Section 2.5.

2.1 Problem description

As described in the general objectives, the overall problem to be solved is the detection and avoidance (or at least the mitigation) of a collision. Moreover, the intention is to keep the number of false interventions few enough[1]. The stated problems are divided into three main areas, which as stated before are; *Long-term path prediction*, *Collision detection* and *Collision avoidance*.

The problems to be addressed for the *Long-term path prediction* are:

- How to fuse the information from map and sensor data to make long-term predictions for both the ego vehicle (EGO) and the Perceived Other Vehicle (POV)?
- How to incorporate sensor imperfections?

The *Collision detection* should consider:

- How to detect a collision?
- How to determine and define the probability of a collision?
- How to define the characteristics of a collision (time to collision, duration of collision, distance to collision)?

The *Collision avoidance* should finally answer:

- How to decide whether an intervention is needed?
- How to make an avoidance intervention (by performing an emergency braking) as late as possible and with very few false interventions?

[1] ISO26262 requires less than one false Autonomous Emergency Braking (AEB) intervention during 10 years or within 100 000 km of driving. This is though out of the scope of the thesis since extensive test hours will be needed to evaluate the system fully.

2.2 Assumptions

When solving the problems stated in Section 2.1, some assumptions should be regarded to simplify and delimit the scope of the thesis. The following assumptions are therefore considered throughout the entire thesis report:

- Only two vehicles are considered in each scenario, the EGO and one POV. The EGO is the vehicle with the Collision Avoidance (CA) system.
- Perfect weather and road conditions, i.e. little or no slip is considered.
- The vehicles are moving with a maximum velocity according to the speed limit in urban areas of 50 km/h.
- Reliable digital map data is assumed to be given as input, i.e. no investigation of how map data is created will be concerned.
- Only one lane in each driving direction will be considered, i.e. no lane identification will be needed and collision with vehicles in the own driving lane is not considered.
- The POV and EGO are assumed to be cars (no other road-users will be considered).
- No communication with surrounding objects (moving or stationary), as for example Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I) communication, is considered in this work.
- The sensors used for the EGO are assumed to give global state information. The positions in longitude and latitude as well as global heading. Velocity and angular velocity is assumed to be measured by the vehicles internal measurement unit (IMU) and wheel speed sensors.
- The sensors used to observe the POV are assumed to give measurements relative the EGO. This could be achieved for example by the use of a radar or a camera.

2.3 Vehicle model

The mathematical representation of the EGO and the POV play a crucial role in the robustness and precision of the developed system. The state information of the EGO is easily obtained by on-board sensors such as GPS, IMU and wheel speed sensors. The POV on the contrary, will be observed by sensors mounted on the EGO such as, for example, camera or radar. The state information of the POV will thus never be completely accurate, due to imperfections introduced by the sensors such as for example measurement speed, measurement accuracy or reflection misinterpretation.

With the sensorial information at hand, the representation of the EGO and POV will be made with polar and spherical state space representation, respectively. The state vector configuration will therefore be:

$$\mathbf{x}_{EGO} = \begin{bmatrix} x \\ y \\ \theta \\ v \\ w \end{bmatrix} \text{ and } \mathbf{x}_{POV} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \\ w \end{bmatrix} \tag{2.1}$$

where x and y are the global cartesian positions and w represents the angular velocity, being the same notations for both the EGO and POV. The disparity between the two representations is that the EGO's heading is denoted as θ and the speed as v, whereas the POV's velocities, measured in x- and y-direction, will be denoted as v_x and v_y respectively.

The spherical representation is used due to the intention of considering a radar as on-board sensor, which can only accurately measure positions and the radial components of the velocity, i.e the instantaneous change in range between the radar and the target. Heading and angular velocity can therefore not be observed for the POV. The angular velocity will not be measured, but is nevertheless in the state vector for the ability to include a constant turn rate to be used in a turn model.

To model a vehicle in the most accurate way possible is a well known problem, for example is a review of the most common motion models described by [15]. Here the authors define models describing a constant motion such as with the Constant Velocity/Acceleration (CV/CAcc) or the Constant Turn-rate (CT) model. More comprehensive models are also presented, where some include several motions in the same model such as the Constant Turn Rate and Velocity/Acceleration (CTRV/CTRA) model or with the inclusion of topology knowledge through the Constant Curvature and Acceleration (CCA) model.

These models all have pros and cons with respect to their ability to accurately model a vehicle, but where none really manages to be as exact as the other in their respective ”motion of expertise”. The conclusion is that each model describing a single motion (CV, CT or CAcc), is the best suited model to use in their respective motion. The problem thus occurs when trying to represent a vehicle that transitions between different motions.

One solution is an Interacting Multiple Model (IMM), which uses a specific model when it is suited. This methodology will be described in Chapter 3, where the CV- and CT-models for both the EGO and POV denoted according to (2.1) is given as:

$$\textbf{CVP:} \quad \underbrace{\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \theta \\ \mathbf{v} \\ \mathbf{w} \end{bmatrix}}_{\mathbf{x}(t+\Delta T)} = \underbrace{\begin{bmatrix} x \\ y \\ \theta \\ v \\ 0 \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} v \cdot cos(\theta) \cdot \Delta T \\ v \cdot sin(\theta) \cdot \Delta T \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{x}(\Delta T)} \tag{2.2}$$

$$\textbf{CTP:}\quad \underset{\mathbf{x}(t+\Delta T)}{\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\\ \theta\\ \mathbf{v}\\ \mathbf{w}\end{bmatrix}} = \underset{\mathbf{x}(t)}{\begin{bmatrix}x\\ y\\ \theta\\ v\\ 0\end{bmatrix}} + \underset{\mathbf{x}(\Delta T)}{\begin{bmatrix}v\cdot cos(\theta)\cdot\Delta T\\ v\cdot sin(\theta)\cdot\Delta T\\ w\cdot\Delta T\\ 0\\ v/R\end{bmatrix}} \tag{2.3}$$

$$\textbf{CV:}\quad \underset{\mathbf{x}(t+\Delta T)}{\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\\ \mathbf{v}_x\\ \mathbf{v}_y\\ \mathbf{w}\end{bmatrix}} = \underset{\mathbf{x}(t)}{\begin{bmatrix}x\\ y\\ v_x\\ v_y\\ 0\end{bmatrix}} + \underset{\mathbf{x}(\Delta T)}{\begin{bmatrix}v_x\cdot\Delta T\\ v_y\cdot\Delta T\\ -v_y\cdot w\cdot\Delta T\\ v_x\cdot w\cdot\Delta T\\ 0\end{bmatrix}} \tag{2.4}$$

$$\textbf{CT:}\quad \underset{\mathbf{x}(t+\Delta T)}{\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\\ \mathbf{v}_x\\ \mathbf{v}_y\\ \mathbf{w}\end{bmatrix}} = \underset{\mathbf{x}(t)}{\begin{bmatrix}x\\ y\\ 0\\ 0\\ 0\end{bmatrix}} + \underset{\mathbf{x}(\Delta T)}{\begin{bmatrix}\frac{v_x}{w}\cdot sin(w\cdot\Delta T)-\frac{v_y}{w}\cdot(1-cos(w\cdot\Delta T))\\ \frac{v_x}{w}\cdot(1-cos(w\cdot\Delta T))+\frac{v_y}{w}\cdot sin(w\cdot\Delta T)\\ v_x\cdot cos(w\cdot\Delta T)-v_y\cdot sin(w\cdot\Delta T)\\ v_x\cdot sin(w\cdot\Delta T)+v_y\cdot cos(w\cdot\Delta T)\\ v/R\end{bmatrix}} \tag{2.5}$$

where ΔT denotes the time step between samples. CVP (Constant Velocity Polar) and CTP (Constant Turn-rate Polar) describes the EGO's motion, where CVP will be the used model for straight path and CTP for turning. The CV model, used for straight path, and CT model, used for turning manoeuvre, describe the POV's intended motion.

2.4 Environmental representation

Since the representation of the EGO and POV should be made as accurately as possible, the environment representation plays a key part in the Collision Avoidance (CA) system's ability to work as intended.

Hence, since sensors collect data from different point-of-views, all the data need to be brought to the same reference frame. For example, a GPS gives information of longitudinal and lateral position at the earths surface (Geodetic frame) whereas mounted sensors such as a radar or a camera generates data relative the object its mounted on (Local frame).

Since the observations of the EGO will be captured in the geodetic frame and the POV in the local frame, these will need to be transformed into a common coordinate frame. To also be able to benefit from map-based constraints and traffic rules, the states of the accounted vehicles will need to be mapped up to the global cartesian coordinate frame.

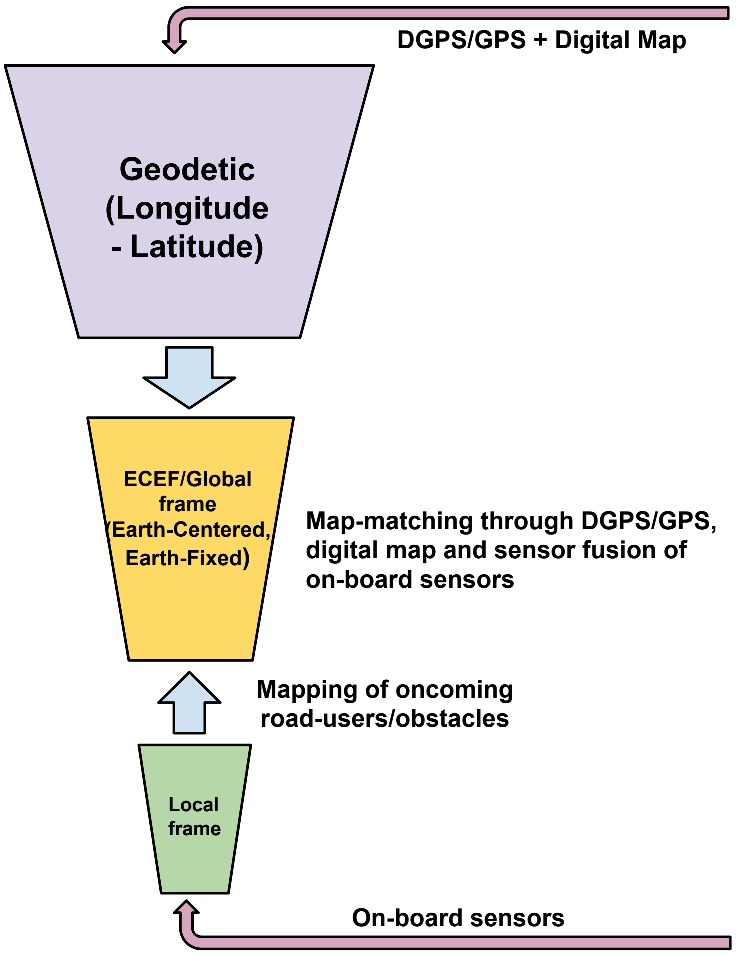

DGPS/GPS + Digital Map
Geodetic (Longitude - Latitude)
ECEF/Global frame (Earth-Centered, Earth-Fixed)
Map-matching through DGPS/GPS, digital map and sensor fusion of on-board sensors
Mapping of oncoming road-users/obstacles
Local frame
On-board sensors

Figure 2.1: Block scheme showing how the different objectives will be regarded as seen to different coordinate systems

This conversion process is illustrated in Figure 2.1, where the use of the three different coordinate systems are described as:

- *Geodetic*; which is a global coordinate system seen to the earths surface, e.g. positions in longitude and latitude.
- *Earth-Centered, Earth-Fixed* (ECEF); which is a global cartesian coordinate system.
- *Local*; is a local cartesian coordinate system, with origin on the EGO vehicle.

The chosen strategy is to transform all the state information to the middle layer, ECEF, thus being able to combine all the data, with the additional possibility of locating the vehicles on a digital map.

2.5 Evaluation setup

The proposed algorithms should be integrated in simulation environments such as *MATLAB/Simulink* and *PreScan* [16]. Furthermore, experimental validation tests should be driven, in collaboration with DENSO's technical staff, using fully equipped Volvo S60 demo vehicles.

Verification of the system functionality is conducted through a variety of test scenarios, and is evaluated in both four-way intersections as well as T-intersections. The scenarios are including both non-evasive and evasive manoeuvres and are further explained in Chapter 6. For full evaluation of the system, different restrictions on the vehicles' possible paths are set offline e.g. the possible driving directions in an intersection, which in reality would have been given by a digital map.

3
Long-term path prediction

In a traffic situation where an imminent threat has been observed, the time before colliding is often very short. Hence within an occurrence of a collision, it is not sufficient to start braking when the observed threat has entered your designated path, instead the brakes need to be applied before it has entered. Being able to accurately predict the road participants future behaviour, to gain invaluable extra time to act, will increase a Collision Avoidance (CA) system's performance immensely.

The focus of this chapter is therefore to present a solution being able to evaluate information of the ego vehicle's (EGO's) surrounding environment to accurately predict future occurrences. The information comprises of road network and road user knowledge along the intended path, which could be collected by a variety of sensors and tools such as GPS, Inertial Measurement Unit (IMU), radar and digital map. This chapter thus contains a description of how support from map and sensor data can improve the ability to recognise a drivers intended manoeuvre, illustrated in Figure 3.1.

The intended design of the long-term path prediction as a part of the full system is depicted in Figure 3.2. Here the propagated trajectories are calculated in parallel for both the EGO and the Perceived Other Vehicle (POV) at each time step. The subsystem contains three in series coupled parts taking care of data collection and validation, filtering and finally prediction.

This chapter is divided into five sections. First, an introduction of the related work will be given in Section 3.1. Secondly, the method for incorporation of driver behaviour and map data to predict a future manoeuvre is explained in Section 3.2. Section 3.3 describes the implemented motion models with associated filtering method to represent a vehicle's state. Thereafter is the method to perform long-term path predictions explained in Section 3.4. Finally, the combination of the three blocks leading to the complete long-term path prediction, is described in Section 3.5

3.1 Related work

In the literature, several approaches are used for the purpose of positioning the EGO and POV as well as predicting their probable paths. The most common method used to estimate the current position and predicting the future position, is by the use of a Kalman filter, or its extensions such as the *Extended Kalman Filter* (EKF) or the *Unscented Kalman Filter* (UKF) as has been done in [17]. With the use of a Kalman filter comes the ability of adding state constraints, which as for example has been made in [18]. Here, the authors use map data providing information of road restrictions to be used as constraints. With the use of such constraints, some approaches resulted in methods able to detect in which lane a vehicle is travelling. As for example in [19], where a Bayesian Network is used to identify the most probable driving lane by fusing map and sensor data.

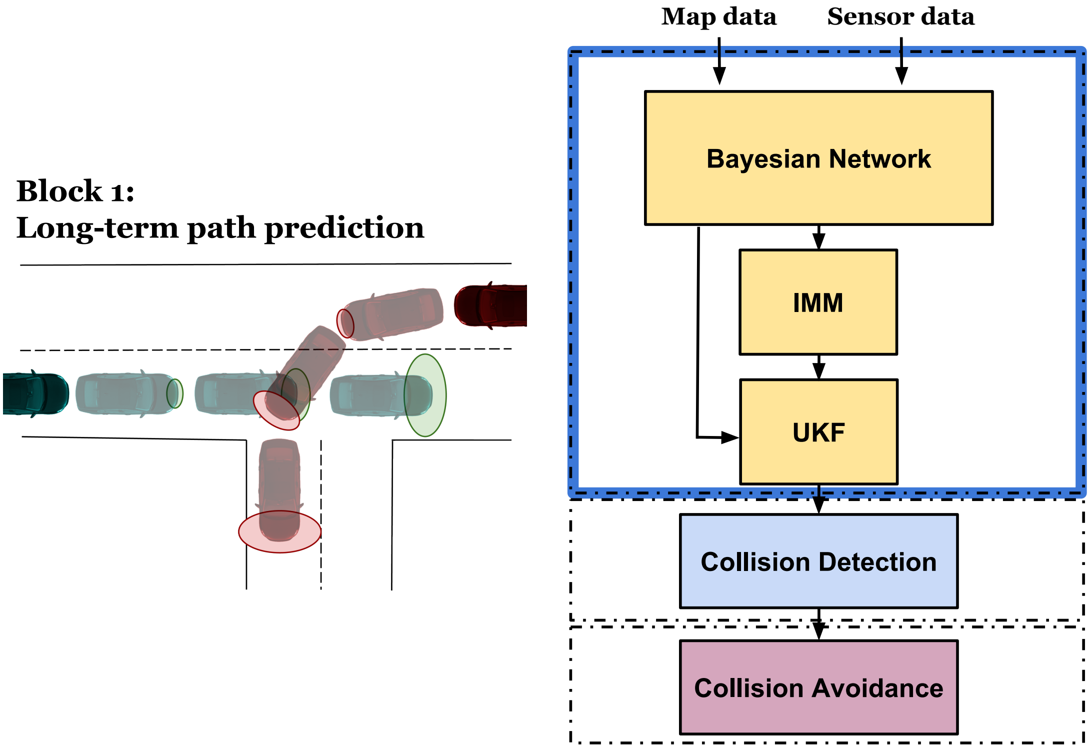


Figure 3.1: Overview of the Long-term path prediction subsystem as a part of the whole system, where the *Bayesian Network* (BN) is covered in Section 3.2, *Interacting Multiple Models* (IMM) in Section 3.3 and *Unscented Kalman Filter*(UKF) described in Section 3.4. The left part of the figure illustrates the intended procedure with predictions (shown as ellipses) for both vehicles included in the scenario.

The approach of pinpointing a vehicles position on a map is denoted as *map-matching*, which has been extensively studied within the literature. Since the developments in this area are so extensive, the map data could either be assumed to be fully reliable as in [19] or with a small uncertainty as in [20]. In [21] the positioning is achieved with an UKF assuming ”perfect” map knowledge, where the POV's location on the map is made through *Vehicle-to-Vehicle* (V2V) communication. This is also the underlying assumption for the propossed design, hence will only consider reliable map data to be given as input, i.e. no investigation of how map data is created will be concerned.

With the position of a vehicle considered to be reliably defined, focus could instead lie on looking at how to predict the future trajectory of the vehicle. An approach presented by [22], combines information from traffic rules, digital map and sensor data to result in a predicted trajectory for the considered vehicle.

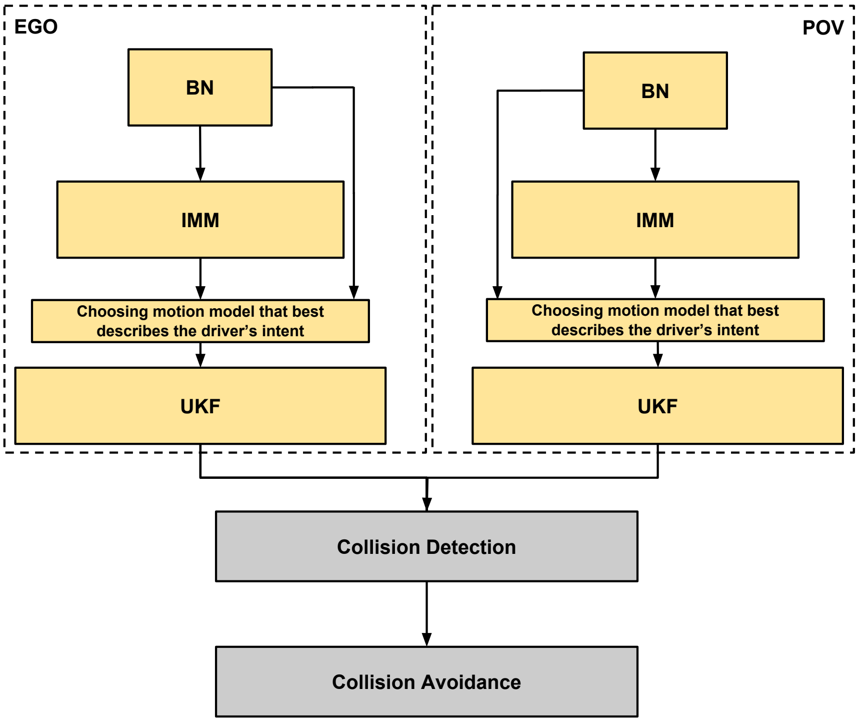

EGO
POV
BN
IMM
Choosing motion model that best describes the driver's intent
UKF
BN
IMM
Choosing motion model that best describes the driver's intent
UKF
Collision Detection
Collision Avoidance

Figure 3.2: Overview of the connection between BN, IMM and UKF, which is running in parallel for the EGO and POV. Their respective probable paths are then compared and analysed with aim at predicting and preventing an imminent collision through the Collision Detection and Collision Avoidance blocks.

This approach uses an EKF for vehicle positioning in accordance with the provided observations. Additionally an Object Oriented Bayesian Network (OOBN) is used to define probabilistic relations between the different information parameters for the ability to predict a probable future trajectory.

In [23] the authors use a Bayesian Network (BN) to combine driver behaviour (i.e. turn signal) and the layout of a specific intersection in order to estimate drivers' probable manoeuvres. A similar solution using Dempster Shafer's theory is presented in [24], where instead the different probabilities are set as hypotheses (stop before the intersection or taking a right, left or straight path) dependent on velocity changes. The hypotheses could thereafter be given altered probabilities depending on road network constraints. The profiles for the hypotheses were in this approach statistically created using real data.

The filtering technique, such as in Kalman filters, usually requires a model to represent the moving objects probable motion. A common approach incorporating a vehicles movement is by the use of a singular motion model such as the *Constant Velocity* (CV), *Constant Turn rate* (CT) or *Constant Acceleration* (CAcc) models.

Tracking driver intentions is however complicated since vehicles do not exhibit one type of motion but rather tend to switch between a set of typical motions. An alternative way of including several motion models simultaneously (such as CT for turning and CV or CAcc for straight driving) and switching to the most suitable model, could be done with the use of *Interacting Multiple Model (IMM)* theory as in [25][26][27].

The proposed solution presented in this chapter is inspired by [26], where they use a combination of BN hypothesis and an IMM-filtering technique for the possibility to estimate the EGO and POV's probable path. This thesis's proposed design will thus use both a BN and an IMM for validation of sensor and map data. The BN will observe the acquired data and compare these to predefined thresholds. If above these thresholds, an answer could be given of probable driver manoeuvre. The IMM will however use the data to tune the motion models representing the different probable behaviours. The BN and IMM will then collaborate to gain a robust decision of which motion model best describing the future probable manoeuvre. The decided model will thereafter be used in an UKF to make predictions some time steps into the future.

Note that the CA system will only be implemented in the EGO, hence all calculations will be made in respect to the own driven vehicle.

3.2 Bayesian Network

A *Bayesian Network* (BN) is a structured graphical representation of probabilistic relationships between different independent variables. A general description of BNs, with theory according to [28], is presented in the sequel, where as the BN applied in the proposed solution is presented in Section 3.2.2.

3.2.1 General description

For the sake of clearness, the main ideas behind the BN will be explained throughout an example. Consider Figure 3.3, where the BN is composed of *nodes* (A to E) and interconnected with arrows to demonstrate the nodes respective dependencies. A node in a BN represents a random independent variable in the sense that they may be observable quantities, latent variables, unknown parameters or hypotheses. The arrows represent probabilistic links, also called *edges*, being conditional dependencies between two nodes.

A description of the intermediate dependencies between all nodes in a network can be defined in a *Conditional Probability Table* (CPT). Nodes not being connected represent variables that are conditionally independent of each other. A *parent* is defined as the predecessor of a node, e.g. node A is the parent of node C in Figure 3.3. Each node is thus associated with a probability distribution, taking different values depending on the observations made by the considered node and connected nodes (child and parent nodes).

The key components of a BN are therefore how the structure of the network as well as how the CPT are built up. For example, since the random variable E, in Figure 3.3, has parents B, C, D a CPT needs to be defined as $P(E|B, C, D)$. If instead E would have had no parents (predecessor nodes), an *Unconditional Probability Table* $P(E)$ should be defined.

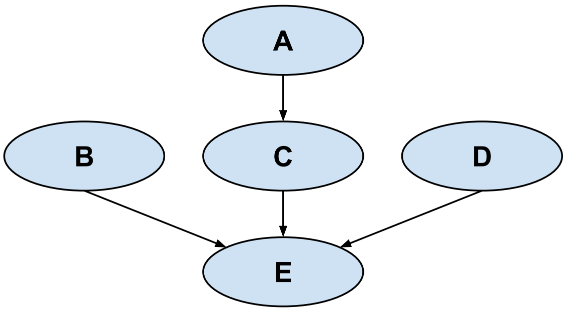


Figure 3.3: Example of DAG (Directed Acyclic Graph) for a BN. A to E are called nodes and the arrows are called edges. The graph shows for example that A, B and D are independent and that A is a parent to C.

Furthermore, to combine observations for different nodes, a well known equation in probability theory is used. The equation is called Bayes' rule, defined in [28], which calculates a *posterior* probability distribution of a parameter V given an observation y. This is given as:

$$P(V|y) = \frac{P(y|V)P(V)}{P(y)} \tag{3.1}$$

This rule can thus be used to update our beliefs regarding V, given the obtained information of the observation y. $P(V)$ denotes the *prior* probability distribution of V and $P(y|V)$ is the *likelihood* of y given the information of V. If V can take a number of m different values ($V = [V_1, V_2, ..., V_m]$), the normalisation constant $P(y)$ can be calculated as:

$$P(y) = \sum_{j=1}^{m} P(y|V_j)P(V_j) \tag{3.2}$$

On the contrary, if there exist multiple observations $y_1, ..., y_n$ being conditionally independent, the joint likelihood distribution of all the observations will be the product of each individual observation. This results in a posterior probability distribution based on Bayes' rule, called *Naive Bayes*, which is defined as:

$$P(V|y_1, ..., y_n) = c_{norm}P(V)\prod_{i=1}^{n} P(y_i|V) \tag{3.3}$$

where c_{norm} is a normalisation constant given as $c_{norm} = 1/P(y_1, ..., y_n)$, and calculated with the use of (3.2) according to:

$$\frac{1}{P(y_1, ..., y_n)} = \sum_{j=1}^{m} P(V_j)\prod_{i=1}^{n} P(y_i|V_j) \tag{3.4}$$

3.2.2 Application

A BN can be used as a tool to process and evaluate information gained from a digital map accompanied with sensor data. The fusion of sensor and map data can infer a drivers behaviour, hence the BN's purpose in the proposed design is to evaluate surrounding information in the same manner as a human driver. Similar to if you would drive a vehicle, you would regard information such as lane alignment of oncoming vehicles, if they decrease in speed or have the turn signal activated. Additional information could be how you interpret the road network to be connected. Each of these perceptions could infer a probable behaviour of the oncoming vehicle. The BN could in the same fashion evaluate this information to predict an EGO's probable behaviour. The desired result is to verify if the driver's intended trajectory leads to a collision, meanwhile performing the avoidance intervention if concluded to be necessary.

Figure 3.4 shows a graphical representation of the proposed BN as a Directed Acyclic Graph (DAG). The BN is built up by a number of nodes, as seen in the figure, namely *Road restrictions*, *Angular velocity*, *Turn signal*, *Acceleration changes*, *Road markings/Traffic rules*, *Lateral alignment* and *Probable manoeuvre*. Static profiles for acceleration, angular velocity and lane alignment have been created by the use of real data, to continuously compare with acquired observations such as map and sensor data. The profiles are built up by statistical information and will thus work in a similar fashion as a lookup-table.

The node *Probable manoeuvre* will finally gather the parent nodes' resulting probabilities and compose these into the probability of left P_{Left}, right P_{Right} and straight $P_{Straight}$ driving direction for each vehicle separately.

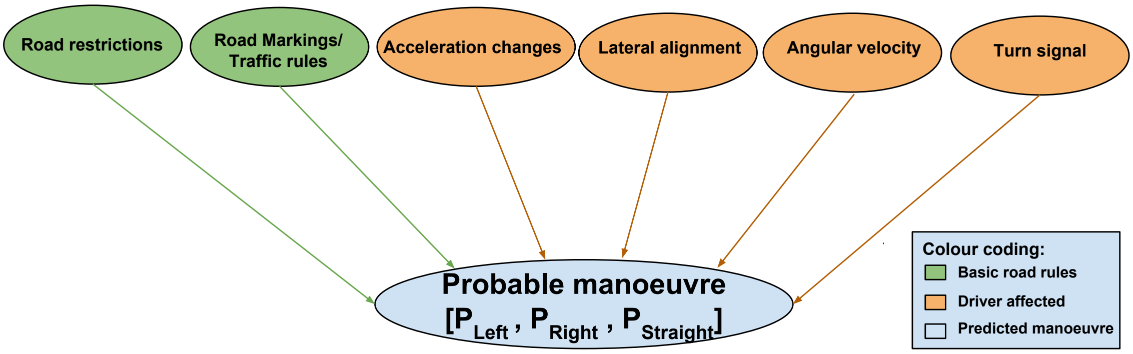


Figure 3.4: Directed Acyclic Graph (DAG) for the BN depicting the different nodes. When new observations are made for each node a new posterior probability will be calculated in node Probable manoeuvre, which is the output of the BN. The Probable manoeuvre will give the probability of left P_{Left}, right P_{Right} and straight $P_{Straight}$

.

The network's different nodes and associated profiles are described in Section A to F, where the last Section G describes the final node *Probable manoeuvre* giving the probability of left, right and straight manoeuvre respectively. The statistical profiles for the nodes *Acceleration changes*, *Lateral alignment* and *Angular velocity*, in Section C, D and F, were created from driving data in an intersection at Bäckebol, Gothenburg seen in Figure 3.5 and 3.6. The statistical profiles were thus created from the behaviour at intersections when performing either a straight, left or right driving manoeuvre.

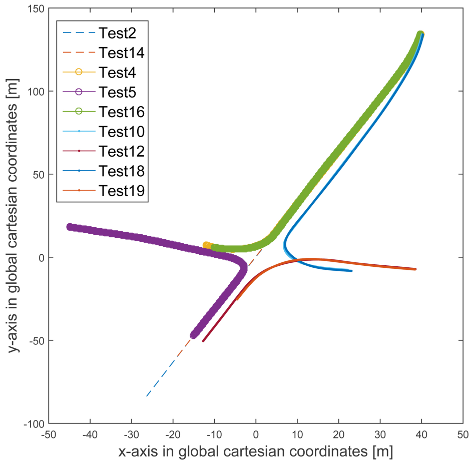


Figure 3.5: Global driving paths for the vehicle when data was collected for creation of the profiles. The distances are measured in meters, with the origin placed in the center of the intersection. North is up and south is down in the figure. The intersection was driven from north to south in the sense that the driven vehicle went from the upper to lower part of the figure.

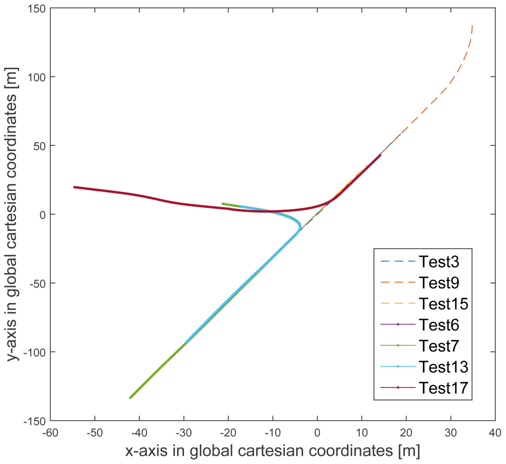


Figure 3.6: Global driving paths for the vehicle when data was collected for creation of the profiles. The distances are measured in meters, with the origin placed in the center of the intersection. North is up and south is down in the figure. The intersection was driven from south to north in the sense that the driven vehicle went from the lower to upper part of the figure.

A Road restrictions

The node *Road restrictions* describes how the roads, in for example an intersection, are linked together. Based on the possible connections for the road the vehicle is travelling on, probabilities of possible manoeuvres will be given. Observe that this node is to be distinguished from *Road markings/traffic rules* since it only depends on the interconnection between the roads, without any consideration of traffic rules. The information for the road restrictions is collected from a digital map. In this work, it will give the probability of a left, right and straight path in an intersection, and can be described by the likelihood of the *Road restrictions* RR given the manoeuvre M as:

$$P(RR|M) = f_{roadRestrict}(roadL(k), roadR(k), roadS(k)) \tag{3.5}$$

where $f_{roadRestrict}$ is a look-up table dependent on the three boolean variables in time step k $roadL, roadR$ and $roadS$ which are set true if there exists a road to the left, right and straight respectively.

B Road markings/traffic rules

The difference between *Road restrictions* and *Road markings/traffic rules* is, that this node evaluates information about possible legal driving directions. The information for the *Road markings/traffic rules* can similar to the observation of *road restrictions* be collected from a digital map, or by observation of markings in the road, such as painted arrows. With respect to the predefined traffic rules for the interconnected roads, the probability of going left, right and straight will be given as the likelihood of the *Road markings*, RM, given the manoeuvre M, according to:

$$P(RM|M) = f_{trafficRules}(LinkL(k), LinkR(k), LinkS(k)) \tag{3.6}$$

where $LinkL$, $LinkR$ and $LinkS$ are boolean variables in time step k set to true if a legal manoeuvre can be made in the direction of left, right and straight respectively. These variables serves as inputs for the function $f_{trafficRules}$, where the function works as a look-up table.

C Acceleration changes

The node regarding the acceleration changes has a dependency on stop lines. If there is a stop line, a vehicle is restricted to be stopping in either case, hence the acceleration behaviour cannot be included in the overall evaluation. If braking occurs and there is a stop line, consequently left, right and straight manoeuvre will be set with equal probability.

If there is no stop line and the vehicle is at a certain distance to the intersection, the acceleration is collected and compared to the predefined profile seen in Figure 3.7 to determine the probability of a left, right and straight driving direction. The profile was created from collected data for normal driving where turning or straight driving manoeuvres were performed. As seen in Figure 3.7, the profile has three different level curves; -0.3, -0.5 and -0.7 $[m/s^2]$, to be used in the absence of stop line. Furthermore, the reason for setting the first level at -0.3 $[m/s^2]$ can also be observed in the figure. The data for the straight manoeuvre can be seen to always be above the -0.3 $[m/s^2]$ limit, hence this is the limit which distinguishes a turn from a straight manoeuvre. The other limits are used to increase the certainty that a turning manoeuvre will be performed, i.e. the higher deceleration, the higher the probability will be of a future turn to be made.

The likelihood of the *Acceleration Changes*, AC, given the manoeuvre M can be described by:

$$P(AC|M) = f_{accChange}(stopline(k), acc(k)) \tag{3.7}$$

where $f_{accChange}$ serves as a the look-up table with the boolean input *stopline*, which is true if a stopline exists, and the input describing the acceleration *acc*, both in time step k.

D Lateral alignment

This node evaluates information of vehicles' alignment in the lane to estimate the probabilities of a straight, left and right manoeuvre. If the vehicle is located more on the left side of the lane, an indication of a probable turning manoeuvre to the left can be given. Likewise, if the vehicle is located more on the right side, a probable turning manoeuvre to the right can be indicated. If instead the vehicle aligns itself with the centerline, this provides information of a probable straight manoeuvre.

The node's resulting probabilities will never be set high for any of the manoeuvres since this evidence is considered to be uncertain. For instance, drivers handle their

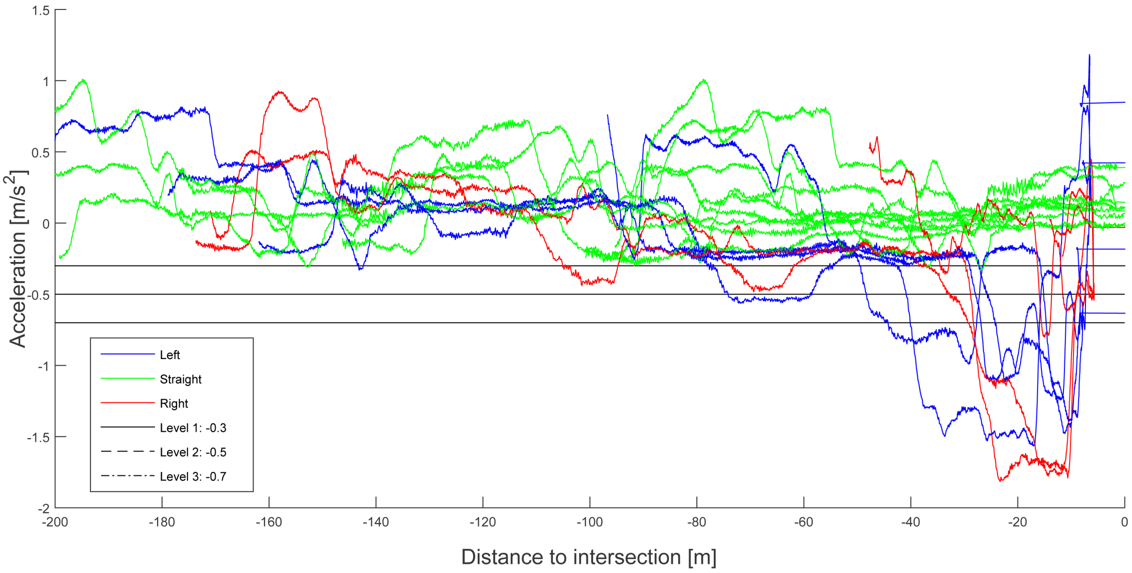


Figure 3.7: Collected real data used for creation of the acceleration profile for the acceleration evidence in the Bayesian Network. The profile levels can be seen as the black lines and are placed on -0.3, -0.5 and -0.7 $[m/s^2]$. Blue colour represents the driving behaviour when turning left, green when going straight and red representing right turn.

vehicles differently, because of for example how drivers have been taught to drive or depending on vehicle properties. Some drivers align their vehicles to make a big turn hence acts in the opposite way of what was described previously. The node can therefore be used to detect if a turn will be made, but not in which direction it will be.

The node will thus monitor how much the vehicle addresses either side of the road, by looking at the ratio between the distances from the vehicle to each side of the lane. The data used to create the profile, for the ability of defining the probabilities of each manoeuvre, can be seen in Figure 3.8. The different test scenarios, used when collecting the data, can be compared and seen in Figure 3.5 and 3.6. The result from the collected data shows that if the vehicle is detected to lean more than 50 [cm] towards either side of the own lane, a high probability will be given to the two turning manoeuvres. Otherwise a higher probability will be given to the straight manoeuvre. These probabilities are described by the function $f_{lateralAlign}$, which is used to compute the likelihood of the lateral alignment LA, given the manoeuvre M:

$$P(LA|M) = f_{lateralAlign}(LeftDelim(k), RightDelim(k)) \tag{3.8}$$

where $LeftDelim$ and $RightDelim$ are the distances to the lane markers on the left and the right side of the vehicle respectively, in time step k.

E Turn signal

The node regarding the turn signal is observing if, and in which direction, the turn signal is activated. If a turn signal is activated the probability of a turn is set high.

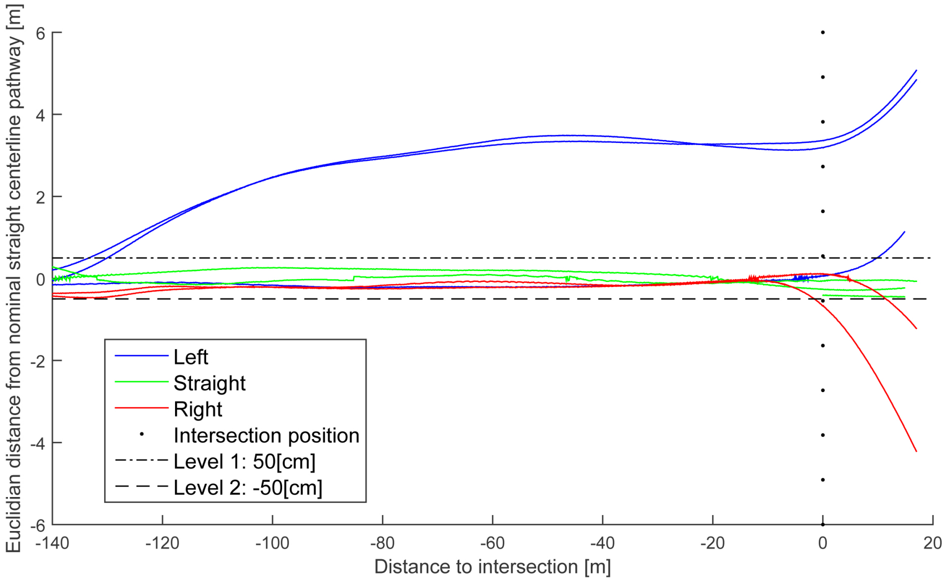


Figure 3.8: Profile for the lateral alignment node in the Bayesian Network.

If a left turn signal is activated, then it is highly probable that a turn to the left will be made and vice versa for a right turn signal. If no turn signal is activated, the probability for a straight manoeuvre is higher, but not equally high as for left/right when a turn signal is on, since the driver could have forgotten to activate the turn signal. Likewise for the left/right turn, the indicator could have been set on by mistake or been set in the opposite direction from the intended, hence could not be fully reliable. Either way, as there in most countries [1] exist laws for the use of turn signals, this is considered to be a strong evidence. The likelihood for *Turn signal*, TS, given the manoeuvres M, can be calculated as:

$$P(TS|M) = f_{turnSignal}(turnsignal(k)) \tag{3.9}$$

where $f_{turnSignal}$ referres to the look-up table used for the input $turnsignal(k)$, set to the activated turn signal (left, right, none), in time step k.

F Angular velocity

Since the *Lateral alignment* node has uncertainty in the alignment, the *Angular velocity* node observes the turn rate as a complement to give a more certain indication of which direction the vehicle will travel. By analysing collected data, seen in Figure 3.9, two thresholds were found and set to ±10 [deg/s]. If the observed data is above the threshold of 10 [deg/s] the probability of right turn will be high. Likewise, if the data is below -10 [deg/s], the probability of left turn will be high. Otherwise a high probability is set for the straight path. The likelihood of the *Angular velocity* AV given the manoeuvre M is given by:

$$P(AV|M) = f_{yaw}(yawRate(k)) \tag{3.10}$$

[1] In Sweden regulated in ”Trafikförordning (1998:1276)”

where f_{yaw} is a look-up table, with the angular velocity, $yawRate$, in time step k as input.

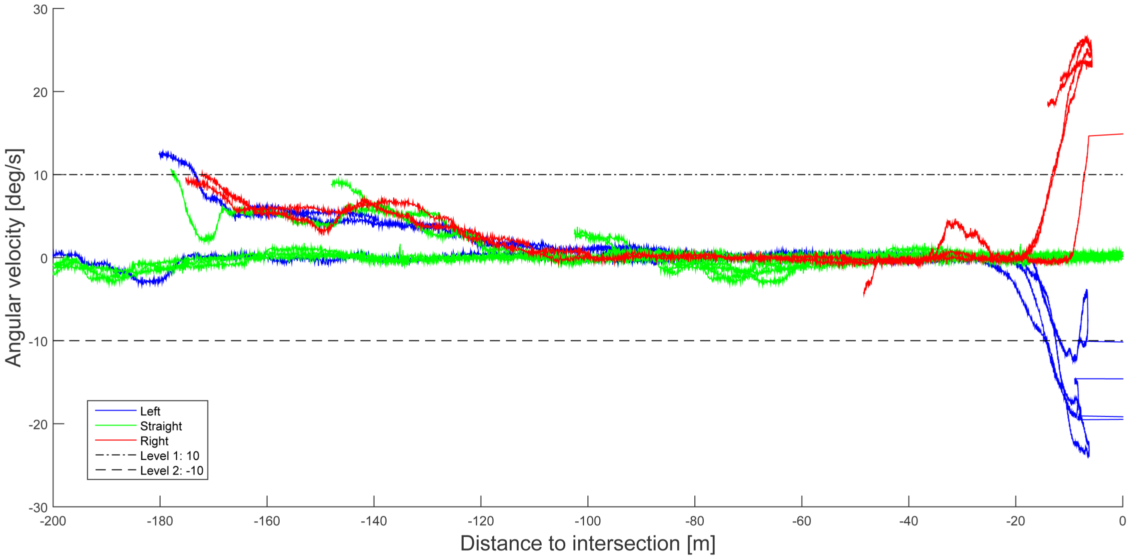


Figure 3.9: Profile for the angular velocity node in the Bayesian Network.

G Probable manoeuvre

The *Probable manoeuvre* node gives the final probability for each manoeuvre based on the probabilities given from the parent nodes, previously described. The resulting probability distribution can be described as:

$$P(M|e_1, ..., e_6) = c_{norm} P(M) \prod_{i=1}^{6} P(e_i|M) \tag{3.11}$$

similar to (3.3), where c_{norm} is a normalisation constant given as $c_{norm} = 1/P(e_1, ..., e_6)$ and calculated according to (3.4) as:

$$\frac{1}{P(e_1, ..., e_6)} = \sum_{j=1}^{6} P(M_j) \prod_{i=1}^{n} P(e_i|M_j) \tag{3.12}$$

Here, M represents the probability distribution of the driving manoeuvres $[L, R, S]$, where L is left, R is Right and S is Straight. The different observations e_i are given by $e = \{RR, AC, RM, AV, LA, TS\}$, which corresponds to the parent nodes *RR* (*Road Restrictions*), *AC* (*Acceleration Change*), *RM* (*Road Markings/Traffic rules*), *AV* (*Angular velocity*), *LA* (*Lateral alignment*) and *TS* (*Turn signal*). The probabilities are thereafter calculated separately as:

$$P_{Left} = P(L|RR, AC, RM, AV, LA, TS) \tag{3.13}$$
$$P_{Right} = P(R|RR, AC, RM, AV, LA, TS) \tag{3.14}$$
$$P_{Straight} = P(S|RR, AC, RM, AV, LA, TS) \tag{3.15}$$

where P_{Left}, P_{Right} and $P_{Straight}$ are the probabilities for left, right and straight manoeuvre respectively. Observe that the prior probability $P(M)$, stated in (3.11), is set equal for all three directions, i.e. 1/3 on each of them. Recall that the likelihoods ($P(e_i|M)$) were described in (3.5) to (3.10).

The result of the *Probable manoeuvre* node will act as complementary weights, via a *Transition matrix*, to the filtering part of the *Long-term path prediction* subsystem. This matrix is used in the Interacting Multiple Model (IMM) filter to provide the probabilities of transitioning from one model to another. The *Transition matrix* is therefore formed, based on (3.13), (3.14) and (3.15), as:

$$\Pi = \begin{bmatrix} \Pi_{Left|Left} & \Pi_{Right|Left} & \Pi_{Straight|Left} \\ \Pi_{Left|Right} & \Pi_{Right|Right} & \Pi_{Straight|Right} \\ \Pi_{Left|Straight} & \Pi_{Right|Straight} & \Pi_{Straight|Straight} \end{bmatrix} = \begin{bmatrix} P_{Left} & P_{Right} & P_{Straight} \\ P_{Left} & P_{Right} & P_{Straight} \\ P_{Left} & P_{Right} & P_{Straight} \end{bmatrix} \tag{3.16}$$

where each row corresponds to the currently used model (Left, Right, Straight) as well as the probability of either staying or transitioning to another model in the next time step. For example, if the current model is left, the matrix gives the probability of either transitioning to the straight or right model or of staying in the left model.

3.2.3 Proposed Bayesian Network algorithm

The input to the BN is the acquired digital map data as well as the state information of the EGO and a possible POV, collected via sensor data. The output from the subsystem is the transition matrix, Π, giving the probability associated with left, right and straight driving direction. The BN for both the EGO and POV will be executed in parallel, resulting in one transition matrix for the EGO and one for the POV.

Algorithm 1 Bayesian Network

```
Input:  Map data, Sensor data
Output:  $\Pi$

Each of the functions are called with associated observations as input
Collect $P(RR|M)$ according to (3.5)
Collect $P(RM|M)$ according to (3.6)
Collect $P(AC|M)$ according to (3.7)
Collect $P(LA|M)$ according to (3.8)
Collect $P(TS|M)$ according to (3.9)
Collect $P(AV|M)$ according to (3.10)

Compute $P_{Left}$ according to (3.13) with use of (3.11)
Compute $P_{Right}$ according to (3.14) with use of (3.11)
Compute $P_{Straight}$ according to (3.15) with use of (3.11)

return $\Pi$                                    ▷ Formed using Equation 3.16
```

3.3 Interacting Multiple Model (IMM) filter

As Chapter 1.2 and 2.1 mentioned, one of the stated problems is how to model the EGO and POV as accurate as possible. As previously explained, tracking driver intentions is however complicated by the fact that vehicles do not exhibit one type of motion but rather tend to transition between a set of typical motions. A way of including several motion models simultaneously and switching to the model best describing the intended driver behaviour, can be done with the use of an Interacting Multiple Model (IMM) filter.

The purpose of using an IMM is the possibility to include two or more filters, each associated with different motion models for target tracking. Filtering can be done with any desired filter, such as for example an *Extended Kalman Filter* (EKF) or *Unscented Kalman Filter* (UKF). The IMM will moreover form a weighted sum of each filter's output, hence being able to rapidly adjust to the observed vehicles probable manoeuvre. Computational complexity can become a problem since there will be as many filters running in parallel as incorporated motion models. The EKF is therefore chosen in the proposed design due to being an admitted and computational light filtering technique.

In an IMM approach, the system state is described by a certain motion model denoted as *mo*. The estimated *true model* is thereafter determined from a finite and predefined set of alternative models. If a change of manoeuvre has been made by the observed driver, a transition to another model best describing the new vehicle dynamics is required. The motion model that best matches the observed behaviour, is therefore said to be changeable between each consecutive time step k.

To be able to accurately predict which motion model best representing the probable next manoeuvre, each model and associated filter will need to be continuously updated. The state and measurement estimate is therefore computed for each of the N number of models. The mathematical formulations in this section have notations according to [29], with the linear state $\mathbf{x}$(k) and measurement $\hat{z}$(k) representation given as:

$$\begin{aligned} \mathbf{x}(k) &= f_{mo}(k-1, \mathbf{x}(k-1), q(k-1)) \\ \hat{\mathbf{z}}(k) &= h_{mo}(k-1, \mathbf{x}(k), \varepsilon(k)) \end{aligned} \tag{3.17}$$

where f_{mo} and h_{mo} are dependent on the corresponding model. Here, *q(k-1)* and *ε(k)* represent the process and measurement noise respectively. The reason why the state and measurement representation should be on the same form, is due to the IMM-algorithms requirement of being computed the same way irrespectively of motion model. These requirements are reflected in the state space representation of (3.17) as:

$$\begin{aligned} \mathbf{x}(k) &= A_{mo}\mathbf{x}(k-1) + B_{mo}q(k-1) \\ \hat{\mathbf{z}}(k) &= C_{mo}\mathbf{x}(k) + D_{mo}\varepsilon(k) \end{aligned} \tag{3.18}$$

where A_{mo}, B_{mo}, C_{mo} and D_{mo} are defined matrices in accordance with the associated motion model, as defined in Section 2.3. These matrices are restricted by the requirement that they need to be of the same size irrespectively of which model in use.

A summary of the IMM approach is thus that the main purpose is to estimate the *true model* from a finite and predefined set of motion models. This *true model* is assumed to change in time, hence both previous state information and future probability need to be considered in the evaluation.

A cycle of an IMM-algorithm can be computed in four steps, illustrated in Figure 3.10, which comprises of:

1. Calculation of *Mixing probability update* and *Mixing*, which integrates the previous state with the probable future behaviour given by both the Bayesian Network (BN) and the result from the previous iteration time step, hence mixing the provided information. This is described in Section 3.3.1.

2. *Filtering*, delivering the current state for each motion model, further described in Section 3.3.2.

3. Calculation of *Model probability update*, hence delivers the updated belief of probable manoeuvre. This is described more thoroughly in Section 3.3.3.

4. *Output estimate calculation*, giving the weighted output of the different models with probability of next manoeuvre incorporated into the resulting state and covariance. The final step is given in Section 3.3.4.

Section 3.3.5 describes the intended application of the IMM with the associated pseudo-code given in Section 3.3.6.

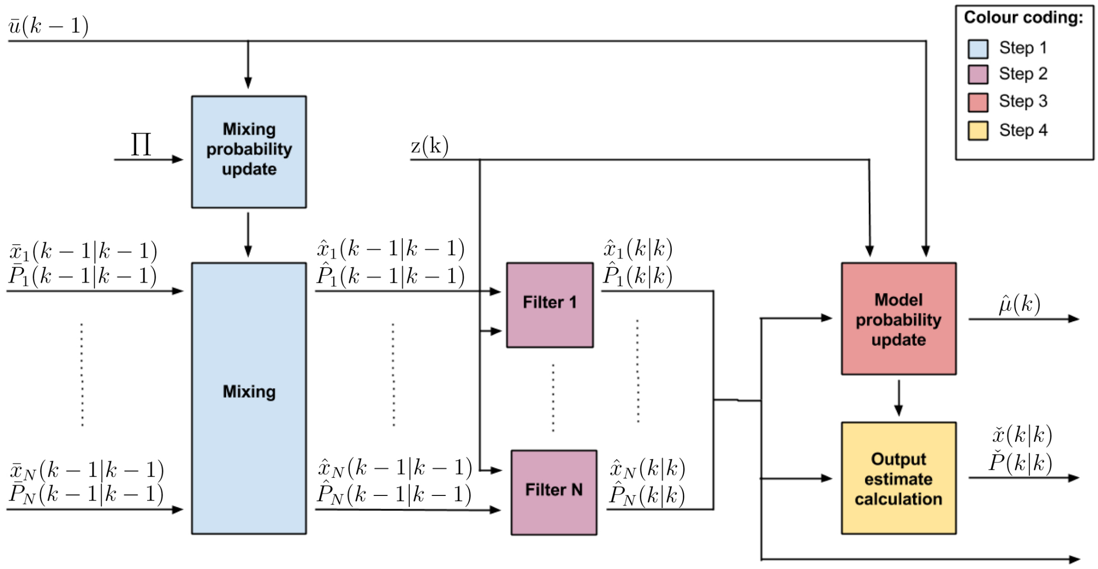


Figure 3.10: Overview of a cycle in the IMM-filter comprised of four computation steps.

3.3.1 Mixing probability update and mixing

The mixing step of the IMM-filter has the purpose of integrating/mixing knowledge of state and covariance from previous iteration, accompanied with the probability of future driver manoeuvre. The mixing probability update $\hat{\mu}(k-1)$ is defined in regard to the probable future manoeuvre predicted by both the Bayesian Network, through the *Transition matrix* Π(k) given in (3.16), and model probability given from previous iteration $\bar{\mu}(k-1)$.

The output from the mixing step will be an update of the previously estimated state and covariance, working as an input to the filtering step.

The updated mixing probability for each model $\hat{\mu}_{mo}(k-1)$ can be calculated as:

$$\hat{\mu}_{mo}(k-1) = \frac{\Pi_{mo} \cdot \bar{\mu}(k-1)}{\sum_{r=1}^{N} \Pi_r \cdot \bar{\mu}(k-1)}, \text{ for } mo = 1, ..., N \tag{3.19}$$

where Π_{mo} (and similarly Π_r in the summation) denotes each model's probability for left, right and straight in the matrix described in (3.16).

The mixed initial state estimate for each model, denoted as $\hat{\mathbf{x}}_{mo}(k-1|k-1)$, can thereafter be updated with the combined knowledge of each model's previous state $\bar{\mathbf{x}}_{mo}(k-1|k-1)$ and mixing probability $\hat{\mu}_{mo}(k-1)$ as:

$$\hat{\mathbf{x}}_{mo}(k-1|k-1) = \sum_{r=1}^{N} \bar{\mathbf{x}}_r(k-1|k-1) \cdot \hat{\mu}_{mo}(k-1), \text{ for } mo = 1, ..., N \tag{3.20}$$

and correspondingly can the initial estimated covariance for each model $\hat{P}_{mo}(k-1|k-1)$ be updated, using each model's previous covariance $\bar{P}_{mo}(k-1|k-1)$ together with the difference between previous and updated state estimate, as:

$$\hat{P}_{mo}(k-1|k-1) = \sum_{r=1}^{N} \hat{\mu}_{mo}(k-1) \cdot \Big\{ \bar{P}_{mo}(k-1|k-1) + [\bar{\mathbf{x}}_r(k-1|k-1) - \hat{\mathbf{x}}_{mo}(k-1|k-1)] \cdot [\bar{\mathbf{x}}_r(k-1|k-1) - \hat{\mathbf{x}}_{mo}(k-1|k-1)]^T \Big\}, \text{ for } mo = 1, ..., N \tag{3.21}$$

3.3.2 Filtering

The initial state estimate $\hat{\mathbf{x}}(k-1|k-1)$ and corresponding initial covariance $\hat{P}(k-1|k-1)$, acquired from Section 3.3.1, are thereafter used for a prediction through the predefined filters, described in Subsection A, for each model. The state is thereafter compared with an observed measurement $z(k)$ in Subsection B, for the ability to make an accurate prediction and update. The result will be an update of the initial state estimate, denoted as $\hat{\mathbf{x}}_{mo}(k|k)$ and covariance $\hat{P}_{mo}(k|k)$ working as inputs in both the *Model probability update* and *Output estimate calculation*, as well as being used in the next iteration step of the IMM-filter.

A Prediction update

The predicted state $\hat{\mathbf{x}}_{mo}(k|k-1)$ and covariance $\hat{P}_{mo}(k|k-1)$ are calculated for each model according to:

$$\hat{\mathbf{x}}_{mo}(k|k-1) = A_{mo} \cdot \hat{\mathbf{x}}_{mo}(k-1|k-1) \tag{3.22}$$

$$\hat{P}_{mo}(k|k-1) = A_{mo} \cdot \hat{P}_{mo}(k-1|k-1) \cdot A_{mo}^T + B_{mo} \cdot Q_{mo} \cdot B_{mo}^T \tag{3.23}$$

where Q_{mo} is the process noise covariance matrix and A_{mo} together with B_{mo} are the defined matrices acquired from Section 2.3, for each motion model.

B Measurement update

The state $\hat{\mathbf{x}}_{mo}(k|k)$ and covariance $\hat{P}_{mo}(k|k)$ are computed using the observed measurement $z(k)$, predicted state $\hat{\mathbf{x}}_{mo}(k|k-1)$ and covariance $\hat{P}_{mo}(k|k-1)$. This is calculated as:

$$\hat{z}_{mo}(k) = C_{mo} \cdot \hat{\mathbf{x}}_{mo}(k|k-1) \tag{3.24}$$

$$S_{mo}(k) = C_{mo} \cdot P_{mo}(k|k-1) \cdot C_{mo}^T + D_{mo} \cdot R \cdot D_{mo}^T \tag{3.25}$$

$$K_{mo}(k) = \hat{P}_{mo}(k|k-1) \cdot C_{mo}^T \cdot S_{mo}(k)^{-1} \tag{3.26}$$

$$\hat{\mathbf{x}}_{mo}(k|k) = \hat{\mathbf{x}}_{mo}(k|k-1) + K_{mo}(k) \cdot (z(k) - \hat{z}_{mo}(k)) \tag{3.27}$$

$$\hat{P}_{mo}(k|k) = \hat{P}_{mo}(k|k-1) - K_{mo}(k) \cdot S_{mo}(k) \cdot K_{mo}(k)^T \tag{3.28}$$

where $\hat{z}_{mo}(k)$ denotes the predicted measurement, $S_{mo}(k)$ the innovation covariance matrix and $K_{mo}(k)$ the Kalman gain for each motion model. C_{mo} and D_{mo} are defined matrices according to Section 2.3. Notice that the measurement noise covariance matrix R is defined irrespectively of which motion model in use.

3.3.3 Model probability update

After the filtering has been conducted, the probability of using each motion model needs to be updated in accordance with the newly estimated covariance $\hat{P}_{mo}(k|k)$ and innovation covariance $S_{mo}(k)$ matrix. The updated model probability $\hat{\mu}_{mo}(k)$, also denoted as the weights, will thereafter be passed to the *Output estimate calculation* step as well as being used in the next iteration step of the IMM-filter.

With the state and covariance as well as the observed measurement $z(k)$ and predicted measurement $\hat{z}_{mo}(k)$, the likelihood of probable future motion model can be expressed as:

$$\Lambda_{mo}(k) = \frac{1}{\sqrt{|2\pi\hat{P}_{mo}(k|k)|}} \exp\left(-0.5\left(z(k) - \hat{z}_{mo}(k)\right)^T \left(S_{mo}(k)\right)^{-1} \left(z(k) - \hat{z}_{mo}(k)\right)\right) \tag{3.29}$$

The model probabilities $\hat{\mu}_{mo}(k)$ will thereafter be updated to be used in the next iteration, using the likelihood $\Lambda_{mo}(k)$, according to:

$$\hat{\mu}_{mo}(k) = \frac{\Lambda_{mo}(k) \cdot \bar{c}_{mo}}{\sum_{r=1}^{N} \Lambda_r(k) \cdot \bar{c}_r}, \text{ for } mo = 1, ..., N \tag{3.30}$$

where the normalisation constant $\bar{c}_{mo}$ is calculated, in accordance with (3.19), as:

$$\bar{c}_{mo} = \sum_{r=1}^{N} \Pi_r \cdot \hat{\mu}_{mo}(k-1), \text{ for } mo = 1, ..., N \tag{3.31}$$

3.3.4 Output estimate calculation

The resulting output estimated state $\check{\mathbf{x}}(k|k)$ and covariance $\check{P}(k|k)$ are finally computed using the weighted sum which incorporates the updated model probability $\mu_i(k)$ as well as state $\hat{x}_{mo}(k|k)$ and covariance $\hat{P}_{mo}(k|k)$ for each of the corresponding models. This is given as:

$$\check{\mathbf{x}}(k|k) = \sum_{r=1}^{N} \hat{\mathbf{x}}_r(k|k) \cdot \hat{\mu}_r(k) \tag{3.32}$$

$$\check{P}(k|k) = \sum_{r=1}^{N} \hat{\mu}_r(k) \cdot \left(\hat{P}_r(k|k) + (\hat{\mathbf{x}}_r(k|k) - \check{\mathbf{x}}(k|k)) \cdot (\hat{\mathbf{x}}_r(k|k) - \check{\mathbf{x}}(k|k))^T \right) \tag{3.33}$$

the variables used for the next iteration is finally assigned as:

$$\begin{aligned} \bar{x}_{mo}(k) &= \check{\mathbf{x}}(k) \\ \bar{P}_{mo}(k) &= \check{P}(k) \\ \bar{\mu}(k) &= \hat{\mu}(k) \end{aligned}$$

3.3.5 Application

As presented in Section 2.3, three different motion models are incorporated in the proposed solution describing left, right and straight manoeuvre. The estimated state and covariance will thus be updated at every time step for each motion model separately using the IMM-filter. These estimates are thereafter combined to result in a weighted sum to be used in the output state and covariance computation.

Instead of only relying on the manoeuvre probability computed by the Bayesian Network (via the *Transition matrix*), the manoeuvre probability from the IMM calculated at the current time step will also be accounted for. This choice is motivated by the fact that the two subsystems give results with respect to different time aspects, where the IMM delivers near-term predictions while the BN makes long-term predictions. The intended application therefore uses a combined manoeuvre probability in the decision process of which motion model best representing the future manoeuvre.

Since the mixing step considers the IMM's manoeuvre probability at the previous iteration step, this needs to be initially defined as:

$$\hat{\mu}(0) = \begin{bmatrix} 0.05 & 0.05 & 0.9 \end{bmatrix} \tag{3.34}$$

where $\bar{\mu}(0)$ is represented by the probabilities for left, right and straight model. The manoeuvre for straight path is initially set to have the highest likelihood (90% certainty), due to that for analysis of data it is assumed that a vehicle's initial pathway has a straight direction.

3.3.6 Proposed IMM algorithm

The proposed IMM-algorithm will continuously compute the filtered state and covariance for the EGO as long as a detected movement has been monitored. If a detection of a POV occurs, simultaneous filtering computations will be conducted using the same IMM-algorithm. To be noted is that the initial covariance matrix $P(1)$ is a predefined matrix with values given in Appendix A.4.

The filtered estimates of each accounted vehicle will thereafter be sent to the final part of the long-term path prediction block, the prediction part by the use of UKF.

Algorithm 2 Interacting Multiple Models filter

```
Input: $\Pi(k)$, Sensor data $z(k)$, Model specifications $f_{mo}$, $Q_{mo}$
Output: $\check{\mathbf{x}}(k)$, $\check{P}(k)$

Initialise: $\bar{\mathbf{x}}_{mo}(0) = z(1), \bar{P}_{mo}(0) = P(1)$
for $mo = 1 : N$ do
    Perform integration of previous estimations using (3.19) to (3.21)
    $[\hat{\mathbf{x}}_{mo}(k-1),\ \hat{P}_{mo}(k-1)] = \text{Mixing}\Big(\bar{\mathbf{x}}_{mo}(k-1), \bar{P}_{mo}(k-1), \Pi(k), \bar{\mu}(k-1)\Big)$

    Perform filtering using the defined model $f_{mo}$ according to (3.23)
    to (3.28)
    $[\hat{\mathbf{x}}_{mo}(k),\ \hat{P}_{mo}(k)] = \text{Filtering}\Big(\hat{\mathbf{x}}_{mo}(k-1), \hat{P}_{mo}(k-1), z(k), f_{mo}, Q_{mo}, R\Big)$
end for
Update the model probability using (3.30)
$[\hat{\mu}(k)] = \text{ModelProbabilityUpdate}\Big(\hat{P}(k), z(k), \hat{\mu}(k-1)\Big)$

Compute the output estimates using (3.33)
$[\check{\mathbf{x}}(k),\ \check{P}(k)] = \text{OutputEstimate}\Big(\hat{\mathbf{x}}_{mo}(k), \hat{P}_{mo}(k), \hat{\mu}(k)\Big)$

Assign the next iteration variables
$\bar{x}_{mo}(k) = \check{\mathbf{x}}(k),\ \bar{P}_{mo}(k) = \check{P}(k),\ \bar{\mu}(k) = \hat{\mu}(k)$

return $\check{\mathbf{x}}(k), \check{P}(k)$
```

3.4 Trajectory prediction using UKF

The *Unscented Kalman Filter* (UKF)-algorithm is comprised of two parts. The first part computes a prediction of the states with corresponding covariances for the chosen motion model, where in the second part these predictions are updated through comparison with received measurements. Thus if the aim is to predict a future trajectory, only the first part of the UKF will be used.

The main advantage with an UKF is the possibility of incorporating the filtering technique irrespectively of the system functions linearity. Another purpose of using an UKF for trajectory prediction is the possibility of having Gaussian distributions to represent driver uncertainty.

The UKF is using a sampling technique called the *Unscented Transform* which selects a minimal set of sample points, denoted as *sigma points*, around the state mean. The sigma points χ, are thereafter propagated through the linear or nonlinear function to accurately capture the mean and covariance of the estimates at each time step. The result is a filter that subsequently can be used to calculate a new estimated mean and covariance.

For a state model with added noise, as described in (3.17), the prediction part of the UKF forms the Gaussian approximation (with the underlying equations from [30]) according to step 1 and 2 below. This is computed for the whole prediction horizon $i = 1,...,n_{predictions}$.

1. A set of 2n+1 sigma points are derived for the augmented state mean $\tilde{\mathbf{x}}$ and covariance $\tilde{P}$, with n being the state dimension, as:

$$\begin{aligned} \chi^0 &= \tilde{\mathbf{x}}(i) \\ \chi^c &= \tilde{\mathbf{x}}(i) + \sqrt{n+\lambda} \cdot \sqrt{\tilde{P}^c(i)} \\ \chi^{c+n} &= \tilde{\mathbf{x}}(i) - \sqrt{n+\lambda} \cdot \sqrt{\tilde{P}^c(i)} \text{ for c = 1,..,n} \end{aligned} \tag{3.35}$$

where c denotes the column of a matrix. The scaling parameter λ is given as:

$$\lambda = \alpha^2(n+\kappa) - n \tag{3.36}$$

where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{P}}$ is given from the previous iterations predicted state and covariance, which at the first prediction step will be initiated with the IMM-filter output ($\check{x}$ and $\check{P}$). The constants α and κ determines the spread of the sigma points around the mean (how large uncertainty you have).

2. Compute the predicted moments by propagation of the sigma points through the function:

$$\tilde{\mathbf{x}}(i+1) = \sum_{s=0}^{2n} \bar{f}_{mo}(\chi^s) \cdot W_s^{\tilde{\mathbf{x}}} \tag{3.37}$$

$$\tilde{\mathbf{P}}(i+1) = \sum_{s=0}^{2n} (\bar{f}_{mo}(\chi^s) - \tilde{\mathbf{x}}(i))(\bar{f}_{mo}(\chi^s) - \tilde{\mathbf{x}}(i))^T \cdot W_s^{\tilde{\mathbf{P}}} + \tilde{Q}_{mo}(i) \tag{3.38}$$

where $\bar{f}_{mo}$ is the function and $\tilde{Q}_{mo}$ is the process noise covariance matrix for the chosen motion model described in Section 2.3. The weights W, are calculated as:

$$W_0^{\tilde{\mathbf{x}}} = \frac{\lambda}{n+\lambda} \tag{3.39}$$

$$W_0^{\tilde{\mathbf{P}}} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \tag{3.40}$$

$$W_\Gamma^{\tilde{\mathbf{x}}} = W_\Gamma^{\tilde{\mathbf{P}}} = \frac{1}{2(n+\lambda)} \text{ , for } \Gamma = 1, ..., 2n \tag{3.41}$$

with β as a parameter to use for incorporating prior information of the input state's distribution.

3.4.1 Application

The proposed long-term prediction design follows the conventional filtering algorithm of the UKF, hence resulting in an estimated state and covariance at each propagated time step. The propagation will be made along a predefined prediction horizon using the chosen model that describes the most probable vehicle trajectory.

An optional part is added to the subsystem being able to monitor the distance to intersection if this is given as input data. The long-term path predictions could thus benefit from the result given by the Bayesian Network. For instance, it could rely on which motion model to use for the predictions once being in the intersection, making for example better predictions of a turn even before the vehicle enters the curve. If the distance to intersection can not be acquired, the chosen prediction model will solely be the result from the IMM.

3.4.2 Proposed trajectory prediction algorithm

The proposed trajectory prediction algorithm follows the mathematical formulations stated in (3.35) to (3.41). The resulting output from the prediction algorithm will be a state vector and covariance matrix describing the whole prediction horizon for each considered vehicle. The calculated state and covariance will be subject to the motion model chosen for the predictions and will be denoted $\tilde{\boldsymbol{x}}_{\boldsymbol{EGO}}$ and $\tilde{\boldsymbol{P}}_{\boldsymbol{EGO}}$ for the EGO as well as $\tilde{\boldsymbol{x}}_{\boldsymbol{POV}}$ and $\tilde{\boldsymbol{P}}_{\boldsymbol{POV}}$ for the POV.

Algorithm 3 Trajectory prediction using UKF

```
Input: $\check{\mathbf{x}}(k), \check{P}(k), \bar{f}_{mo}$
Output: $\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{P}}$

Initialise: $\tilde{x}(0) = \check{x}(k), \tilde{P}(0) = \check{P}(k)$
for $i = 1 : n_{predictions}$ do
    Derive a set of 2n+1 sigma points by the use of (3.35) and (3.36)
    $\chi = \text{FormSigmaPoints}\Big(\tilde{x}(i-1), \tilde{P}(i-1), n, \alpha, \kappa\Big)$

    Compute state and covariance prediction using (3.37) to (3.41)
    $[\tilde{x}(i),\ \tilde{P}(i)] = \text{ComputePredictions}\Big(\chi, \bar{f}_{mo}, \tilde{x}(i-1), \tilde{P}(i-1), \tilde{Q}_{mo}\Big)$
end for
return $\tilde{\boldsymbol{x}} = [\tilde{x}(0), ..., \tilde{x}(n_{predictions})], \tilde{\boldsymbol{P}} = [\tilde{P}(0), ..., \tilde{P}(n_{predictions})]$
```

3.5 Long-Term Path Prediction Procedure

As the introduction in this chapter described, the *Long-term path prediction* subsystem comprises of the three parts *Bayesian Network* (BN), *Interacting Multiple Model* (IMM) filter and *Unscented Kalman Filter* (UKF). These have been defined in Section 3.2, 3.3 and 3.2 respectively, and will in this last section be tied together to describe the overall procedure of the first subsystem, as illustrated in Figure 3.11.

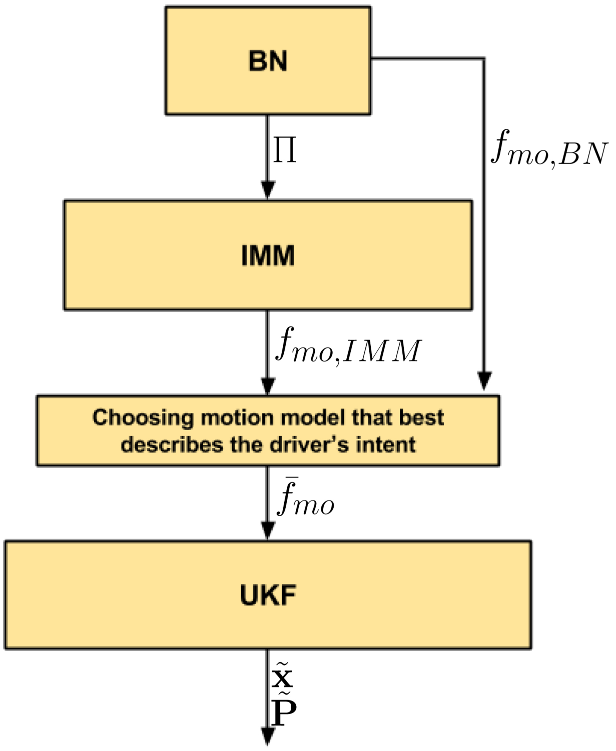


Figure 3.11: Overview of the *Long-term path prediction* subsystem's flow of input and output signals from the different parts.

As illustrated in figure 3.11, the BN sends the *Transition matrix* Π to the IMM. Both the BN and IMM is thereafter sending the state and covariance for the chosen motion model giving the highest probability as output, denoted as $f_{mo,BN}$ and $f_{mo,IMM}$ respectively. The IMM's choice of motion model is however partially affected by the input *Transition matrix*. Note that both the BN as well as the IMM will have the measurement data as input.

Both the BN's and IMM's output will therafter be sent to an evaluation algorithm determining which motion model being best suited for the prediction horizon. If a distance to the intersection is known, the decided motion model to use in the UKF part will vary along the prediction horizon. Namely, the model given by the IMM will be used for the predictions up to the intersection and the BN's chosen model will be used for the prediction horizon in and through the intersection. However, if the distance to intersection is unobtainable, then only the IMM's computed best fit motion model will be used. This evaluation algorithm is in the presented solution integrated into the UKF algorithm, but for the sake of clarity it is described as a stand-alone algorithm.

As the last part of the subsystem, will the UKF use the input motion model properties to propagate the state and covariance along the predefined prediction horizon. The state and covariance for both observed vehicles will thereafter be sent to the *Collision detection* subsystem for evaluation of a probable collision.

3.5.1 Long-term Path Prediction Algorithm

The algorithm described in this section is comprised of the algorithms stated in Section 3.2.3, 3.3.6 and 3.4.2. As mentioned before, the evaluation algorithm have been integrated into the UKF, hence will not be a separate procedure as was illustrated in Figure 3.11.

Algorithm 4 Long-term path prediction algorithm

```
Input: Map data, Sensor data
Output: $\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{P}}$

Make a long-term prediction using the BN:
$[\Pi(k), f_{mo,BN}(k)] = \text{BN(Map data, Sensor data)}$

Make a near-term prediction using the IMM:
$[\check{\mathbf{x}}(k), \check{\mathbf{P}}(k)] = \text{IMM}(\Pi(k), \text{Sensor data } z(k))$

Propagate the state and covariance using the chosen motion model along the predefined prediction horizon:
$[\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{P}}] = \text{UKF}(f_{mo,BN}, \check{\mathbf{x}}(k), \check{\mathbf{P}}(k))$

return $\tilde{\boldsymbol{x}} = [\tilde{x}(0), ..., \tilde{x}(n_{predictions})], \tilde{\boldsymbol{P}} = [\tilde{P}(0), ..., \tilde{P}(n_{predictions})]$
```

4

Collision detection

This Chapter describes how to detect a collision, based on the predictions provided by the Unscented Kalman Filter (UKF) from Chapter 3, as well as how to characterise a collision. Figure 4.1 provides an overview of the collision detection block as a part of the full system, marked in blue. Furthermore, the connections with the other subsystems are also illustrated, where primarily a direct link from the UKF and another link to the Collision Avoidance (CA) subsystem can be seen.

This chapter is organised as follows. First an introduction to collision detection is given with related work described in Section 4.1. Section 4.2 explains how a collision is considered and defined in the scope of this work. Section 4.3 and 4.4 describes a deterministic and probabilistic collision detection algorithm respectively. Finally, the connection and inter-dependencies between the collision detection subsystem and the overall system is outlined in Section 4.5.

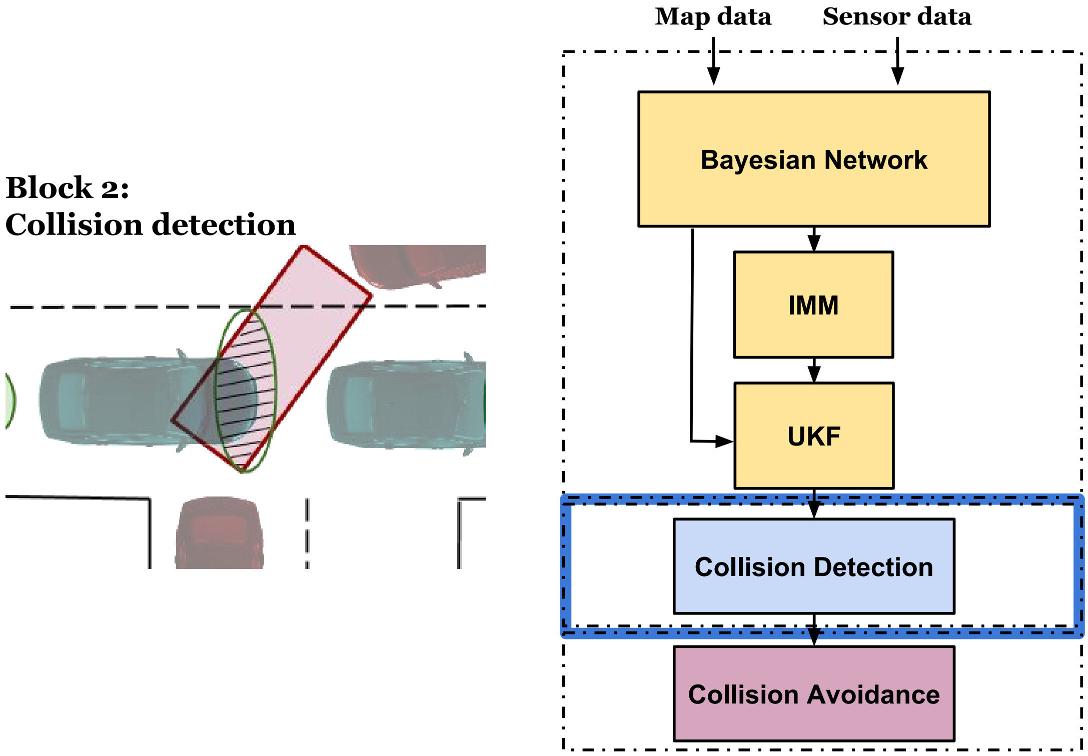


Figure 4.1: Overview of the full system with the collision detection subsystem's position marked in blue.

4.1 Related Work

There are several different methods in the literature to define a collision. One method presented in [31], is based on the intersection of two vehicles' respective areas. Firstly, the authors establish a pessimistic approximation of possible intersections between the ego vehicle (EGO) and the Perceived Other Vehicles (POVs) using circular areas to define the vehicles. If the areas intersect, a more accurately analysis is made using tighter and more representative areas of the vehicles to evaluate if a collision indeed is imminent. This approach is used to reduce the computational burden as well as analysing oncoming threats at an earlier stage. It is worth mentioning that this approach however is fully deterministic.

In [17], a collision detection is performed by defining appropriate collision and collision free domains/areas. The collision areas are defined with respect to the center of both vehicles' front bumper whereas the collision free area is defined according to the center of the POV's rear bumper and the center of the EGO's front. These areas are dependent on the length and the width of both vehicles and are later used to evaluate the eminency of a collision. The evaluation is conducted by computing the probability density over the area of interest with the use of the joint cumulative distribution. Note that their collision detection algorithm only can be applied in a specific traffic situation, namely a left turn by the POV across the EGO's path.

Instead of defining the collision area around the EGO, some research has been conducted on defining one or more collision areas on a map [2][3][22]. This means that, by making assumptions that each vehicle follows a specific path in a lane, the area where a collision can occur is the intersection of two (or more) lanes. Within each specific traffic intersection, the collision area would therefore be defined dependent on the possible intersecting paths of the vehicles.

The objective of this work is to provide a collision detection algorithm suitable for all traffic situations, even if no map/road topology information is available. Since the positions of the vehicles are estimations, the idea is to use a probabilistic approach for evaluation of an imminent collision where [17] has worked as an inspiration.

4.2 Collision Definition

There are a number of different ways to define a collision, as well as the area occupied by a vehicle. The chosen representation of a vehicle area is here based on sets. Simply speaking, when two vehicles are considered, the area of one of the vehicles is represented as a set bounded by the size of the vehicle, whereas the other vehicle is represented by a set, bounded by the size of the vehicle but also rotated according to both vehicles heading.

To explain further, the first step is to define the sets representing the areas of both vehicles. The sets of the EGO and the POV, A_{EGO} and A_{POV} respectively, are bounded by their length and width relative their center point as:

$$A_{EGO} = \left\{ \begin{matrix} A_{xEGO}: & -\frac{l}{2} & \leq & A_{xEGO} & \leq & \frac{l}{2} \\ A_{yEGO}: & -\frac{w}{2} & \leq & A_{yEGO} & \leq & \frac{w}{2} \end{matrix} \right\} \tag{4.1}$$

$$A_{POV} = \left\{ \begin{matrix} A_{xPOV}: & -\frac{\bar{l}}{2} & \leq & A_{xPOV} & \leq & \frac{\bar{l}}{2} \\ A_{yPOV}: & -\frac{\bar{w}}{2} & \leq & A_{yPOV} & \leq & \frac{\bar{w}}{2} \end{matrix} \right\} \tag{4.2}$$

where l is the length and w is the width of the EGO whereas $\bar{l}$ and $\bar{w}$ are the length and width of the POV, respectively. Furthermore, A_{xEGO} and A_{xPOV} are the restrictions in the x-axis, whereas A_{yEGO} and A_{yPOV} are the restrictions in the y-axis. An illustration is presented in Figure 4.2.

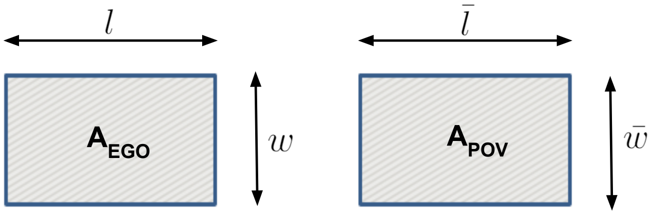


Figure 4.2: Illustration of sets representing the vehicle areas. Here l is the length and w is the width of the EGO, resulting in the EGO set A_{EGO}, whereas $\bar{l}$ and $\bar{w}$ are the length and width of the POV representing the set, A_{POV}.

To be able to evaluate the collision risk, the next step is to define the vehicle areas as global sets of points. Since the measurements, and therefore also the predictions, of the positions are assumed to be the center point of the vehicle, the vehicle areas are defined around this point, e.g. $(xEGO(i), yEGO(i))$ the center point of the EGO in prediction step i. An illustration of the global sets, defined around each vehicle's global cartesian position, can be seen in Figure 4.3. For each prediction step $i = 1, 2, ..., n_{predictions}$ (where $n_{predictions}$ is the number of predictions) the sets are then globally defined as:

$$A_{EGO}^{global}(i) = \left\{ \begin{matrix} A_{xEGO}(i): & xEGO(i) - \frac{l}{2} & \leq & A_{xEGO}(i) & \leq & xEGO(i) + \frac{l}{2} \\ A_{yEGO}(i): & yEGO(i) - \frac{w}{2} & \leq & A_{yEGO}(i) & \leq & yEGO(i) + \frac{w}{2} \end{matrix} \right\} \tag{4.3}$$

$$A_{POV}^{global}(i) = \left\{ \begin{matrix} A_{xPOV}(i): & xPOV(i) - \frac{\bar{l}}{2} & \leq & A_{xPOV}(i) & \leq & xPOV(i) + \frac{\bar{l}}{2} \\ A_{yPOV}(i): & yPOV(i) - \frac{\bar{w}}{2} & \leq & A_{yPOV}(i) & \leq & yPOV(i) + \frac{\bar{w}}{2} \end{matrix} \right\} \tag{4.4}$$

where $xEGO(i)$, $yEGO(i)$ and $xPOV(i)$, $yPOV(i)$ are the cartesian coordinates of the predicted position in prediction step i for the EGO and POV respectively. Furthermore, $A_{xEGO}(i)$ and $A_{xPOV}(i)$ are the restrictions in the global x-axis, whereas $A_{yEGO}(i)$ and $A_{yPOV}(i)$ are the restrictions in the y-axis.

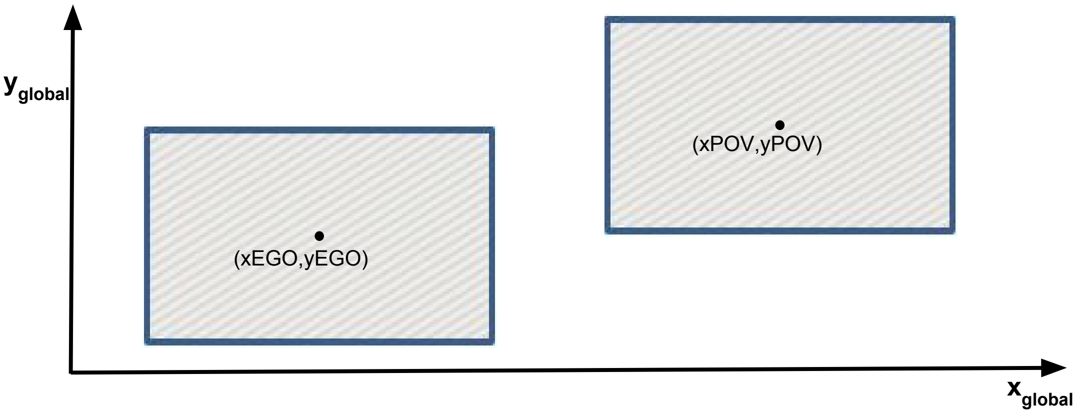


Figure 4.3: Illustration of definition of occupied areas by the vehicles, represented as sets. The sets, here for a specific i, are defined in global coordinates xEGO, yEGO and xPOV, yPOV, without any consideration of heading.

The next step is to rotate the global sets (4.3) and (4.4) according to heading with the use of a rotation matrix. More specifically, first the sets are rotated according to the own vehicle's heading (θ_{EGO} or θ_{POV}) and secondly, rotated according to the other vehicle's heading (θ_{POV} or θ_{EGO}). An illustration can be seen in Figure 4.4, where an example of a scenario is depicted in the leftmost figure. The next two figures show how the scenario is interpreted, where the green colour depicts the vehicles in neutral position (zero heading) according to (4.3) and (4.4) and red demonstrates the sets rotated according to heading. In the middle figure is the EGO in neutral position, whereas in the rightmost the POV is. These last two figures show two distinct representations of the global situation, each centered on the perspective of one of the vehicles.

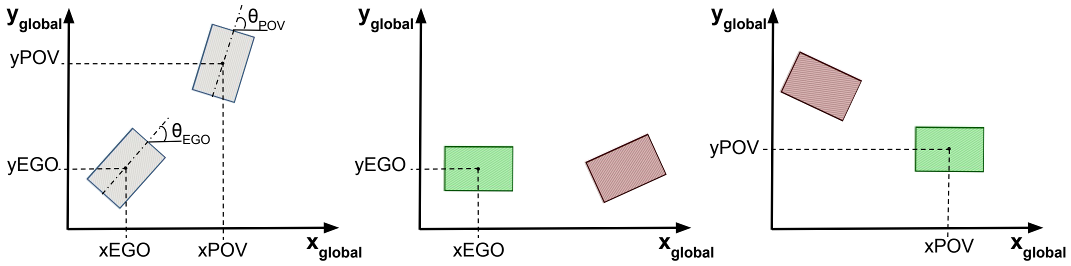


Figure 4.4: Illustration of the interpretation of a traffic scenario, where the scenario can be seen in the leftmost figure. The next two figures shows how the scenario is interpreted, where the green colour depicts the vehicles in neutral position (zero heading) according to (4.3) and (4.4) and red, the global sets rotated according to heading. In the middle figure is the EGO in neutral position, whereas in the rightmost the POV is. These last two figures show two distinct representations of the global situation, each centered on the perspective of one of the vehicles.

With the two representations of the vehicle areas explained above, a collision can be defined. Intuitively, a collision occurs when some part of each vehicle overlap or touches the other. For the defined global sets this can be translated into: if at least one point of a set belongs to the other area, a collision is present.

Note that for simplicity and computational reasons, the global sets defined according to (4.3) and (4.4) are chosen to be represented by a fixed number of points η evenly spread over the vehicle. Moreover, these η rotated points in each prediction step i will be denoted by $Ep^j(i)$ and $Op^j(i)$ for the EGO and POV respectively, where $j = 1, 2, ..., \eta$. $Ep^j(i)$ is used for the points when the interpretation is made as in the rightmost figure of Figure 4.4, whereas $Op^j(i)$ is used when the interpretation is made as in the middle figure. The two interpretations are needed to be able to find the correct probability of collision, since in some configurations one of the representations can give a higher probability of collision than the other. Observe that it is also possible to distribute the points in other patterns rather than uniformly distributed, since discrete points are used.

4.3 Deterministic Collision Detection

The η rotated points distributed over the EGO in each prediction step i is defined by $Ep^j(i)$, where $j = 1, 2, ..., \eta$ is denoting each point. Additionally, the predicted middle point of the POV in the ith prediction step, $xPOV(i), yPOV(i)$, is used as a reference point for the POV's vehicle area, referred to as $O(i)$. For each point on the EGO, $Ep^j(i)$, a translation is made by considering $O(i)$ as the origin of the frame space according to:

$$OEp^j(i) = [Ep^j_x(i) - O_x(i), Ep^j_y(i) - O_y(i)] \quad \text{for } j = 1, 2, ..., \eta \tag{4.5}$$

Here $OEp^j(i)$ denotes the position of $Ep^j(i)$ after the translation, where $(.)_x$ and $(.)_y$ are the cartesian coordinates of each point. Similarly for the POV, η rotated points $Op^j(i)$ are distributed in each prediction step i for each point $j = 1, 2, ..., \eta$. For the EGO the predicted middle point in prediction step i, $xEGO(i), yEGO(i)$, is denoted by $E(i)$. The translation of $Op^j(i)$, by placing the origin in $E(i)$, is thereafter obtained by:

$$EOp^j(i) = [Op^j_x(i) - E_x(i), Op^j_y(i) - E_y(i)] \quad \text{for } j = 1, 2, ..., \eta \tag{4.6}$$

Similarly, here $EOp^j(i)$ denotes the position of $Op^j(i)$ after the translation where, as stated before, $(.)_x$ and $(.)_y$ are denoting the cartesian coordinates of each point.

The existence of a collision, in each prediction time step i, is then calculated by:

$$\mathcal{P}_{collision}(i) = \begin{cases} 1 & \text{if } \exists\, j : OEp^j(i) \in A_{POV} \\ 1 & \text{if } \exists\, j : EOp^j(i) \in A_{EGO} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } j = 1, 2, ..., \eta \tag{4.7}$$

where A_{EGO} and A_{POV} are defined according to (4.1) and (4.2). For the deterministic case, where both vehicle's exact position is known, a collision is then present in prediction step i when (4.7) equals 1. Observe that the reason why there are two equations leading to collision (equals 1) is that a discrete number of points are picked out from the sets. This can thus, in some configurations, lead to one of the equations being set to 1 and the other one not, if only one of the points is inside the other vehicle.

Note that since the trajectories of the vehicles are represented in a stochastic way, a probabilistic collision detection is needed.

4.4 Probabilistic Collision Detection

In this section, a probabilistic collision detection method is described. Firstly, a bivariate normal distribution will be described, and secondly will the joint cumulative distribution be presented for the calculation of the probability of collision.

4.4.1 Bivariate Normal Distribution

A bivariate normal distribution (BND) is a multivariate normal distribution of dimension two. For example, a random cartesian point's position $\boldsymbol{r} = (x, y)$ can be described as a BND according to:

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\overbrace{\begin{bmatrix} m_x \\ m_y \end{bmatrix}}^{\boldsymbol{m}}, \overbrace{\begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}}^{\boldsymbol{\Sigma}} \right) \tag{4.8}$$

where m_x and m_y are the mean of x and y respectively. Furthermore, $\sigma_x > 0$ and $\sigma_y > 0$ denote the variance of the position in the x and y axis respectively, whereas $|\rho| < 1$ is the correlation between the two variables. For the previously mentioned point, the joint density function $\phi(\boldsymbol{r})$ can then be calculated as:

$$\phi(\mathbf{r}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|^{1/2}} e^{-D_{ms}(\mathbf{r})/2} \tag{4.9}$$

where $|\boldsymbol{\Sigma}|$ denotes the determinant of the covariance matrix. The squared Mahalanobis distance between the random point $\mathbf{r}$ and the mean $\mathbf{m}$, denoted $D_{ms}(\boldsymbol{r})$, can be calculated according to:

$$D_{ms}(\mathbf{r}) = (\mathbf{r} - \boldsymbol{m})^T \boldsymbol{\Sigma}^{-1} (\mathbf{r} - \boldsymbol{m}) \tag{4.10}$$

Next, the goal is to find the probabilistic equivalent to the deterministic translation between two BNDs. In accordance with (4.5) and (4.6) this results in linear combinations of the distributions. As an example, for a bivariate distribution with two independent random variables, $X_1 \sim \mathcal{N}(\boldsymbol{m_1}, \boldsymbol{\Sigma_1})$ and $X_2 \sim \mathcal{N}(\boldsymbol{m_2}, \boldsymbol{\Sigma_2})$, with mean $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$ as well as covariances $\boldsymbol{\Sigma_1}$ and $\boldsymbol{\Sigma_2}$, the linear combination of the distributions is generally formulated as:

$$a_1X_1 + a_2X_2 \sim \mathcal{N}(a_1\boldsymbol{m_1} + a_2\boldsymbol{m_2}, a_1^2\boldsymbol{\Sigma_1} + a_2^2\boldsymbol{\Sigma_2} + 2a_1a_2\boldsymbol{\Sigma_{12}}) \tag{4.11}$$

where a_1, a_2 are arbitrary constants and $\boldsymbol{\Sigma_{12}}$ represents the cross-correlation between X_1 and X_2.

In a similar way, the estimated positions of the EGO and POV in prediction step i can be seen as the bivariate normally distributed variables $E(i) \sim \mathcal{N}(\boldsymbol{m_E}(i), \boldsymbol{\Sigma_E}(i))$ and $O(i) \sim \mathcal{N}(\boldsymbol{m_O}(i), \boldsymbol{\Sigma_O}(i))$, respectively. Similarly, the η points, distributed over each vehicle area and rotated according to heading, can be seen as bivariate normal distributed variables, $Ep^j(i) \sim \mathcal{N}(\boldsymbol{m_{Ep^j}}(i), \boldsymbol{\Sigma_E}(i))$ and $Op^j(i) \sim \mathcal{N}(\boldsymbol{m_{Op^j}}(i), \boldsymbol{\Sigma_O}(i))$. Here, $\boldsymbol{m_{Ep^j}}(i)$ and $\boldsymbol{m_{Op^j}}(i)$ are the mean of point j for the EGO and POV respectively, whereas the variances are the same as for the estimated position. Equation (4.5) and (4.11) are thereafter used to define the distribution $D_1^j(i)$, for the new representation of the points $Ep^j(i)$ with the reference frame located in $O(i)$, as:

$$D_1^j(i) \sim \mathcal{N}(\boldsymbol{m_{Ep^j}}(i) - \boldsymbol{m_O}(i), \boldsymbol{\Sigma_E}(i) + \boldsymbol{\Sigma_O}(i)) \qquad \text{for } j = 1, 2, ..., \eta \tag{4.12}$$

Similarly are (4.6) and (4.11) used to define a second distribution $D_2^j(i)$, for the new representation of $Opj(i)$ with the reference frame located in $E(i)$, as:

$$D_2^j(i) \sim \mathcal{N}(\boldsymbol{m_{Op^j}}(i) - \boldsymbol{m_E}(i), \boldsymbol{\Sigma_O}(i) + \boldsymbol{\Sigma_E}(i)) \qquad \text{for } j = 1, 2, ..., \eta \tag{4.13}$$

Note that the covariance $\boldsymbol{\Sigma_{OE}}(i) = \boldsymbol{0}$, given that the positions of the vehicles are uncorrelated.

4.4.2 Joint Cumulative Distribution

To be able to compute the probability of a collision, the joint cumulative distribution function (JCDF) of two variables is used. To get to a probabilistic version of Equation (4.7), first the probability of collision for each point j needs to be calculated according to:

$$\mathcal{P}_r^j(i) = \iint\limits_A D_r^j(i)\,\mathrm{d}x\,\mathrm{d}y = \int\limits_a^b \int\limits_c^d D_r^j(i)\,\mathrm{d}x\,\mathrm{d}y \qquad \text{for } j = 1, 2, ..., \eta \text{ and } r = 1, 2 \tag{4.14}$$

where A denotes the set A_{POV} when $r = 1$ or A_{EGO} when $r = 2$ and j is the η points distributed over the EGO or POV. With the use of the sets, the limits are defined according to (4.1) and (4.2) where a and b are the lower and upper limits in x-axis whereas c and d are the in lower and upper limits in y-axis. The integral can then be solved according to [32].

This results in a vector containing all the collision probabilities for each point $j = 1, 2, ..., \eta$ on both vehicles as:

$$\mathcal{P}_{1,2}(i) = \begin{bmatrix} \mathcal{P}_1^1(i) & \mathcal{P}_1^2(i) & \cdots & \mathcal{P}_1^\eta(i) & \mathcal{P}_2^1(i) & \mathcal{P}_2^2(i) & \cdots & \mathcal{P}_2^\eta(i) \end{bmatrix} \tag{4.15}$$

The probability of collision in prediction step i will thereafter be defined as the maximum probability found in the vector according to:

$$\mathcal{P}_{collision}(i) = max(\mathcal{P}_{1,2}(i)) \tag{4.16}$$

4.5 Collision Detection Procedure

The following section describes the connection between the collision detection block and the rest of the system. First, the acquired inputs given by the UKF will be presented, secondly, the characteristics of a collision required by the collision avoidance subsystem will be defined. Lastly, the collision detection algorithm in pseudo-code will be given.

4.5.1 Predicted states

The predicted state vectors and the corresponding covariance matrices for both the EGO and POV are given from the UKF as output from the long-term path prediction subsystem. The given states are denoted $\tilde{\boldsymbol{x}}_{\boldsymbol{EGO}}$, $\tilde{\boldsymbol{x}}_{\boldsymbol{POV}}$ and the covariances $\tilde{\boldsymbol{P}}_{\boldsymbol{EGO}}$, $\tilde{\boldsymbol{P}}_{\boldsymbol{POV}}$, as seen in Section 3.4.2. The states, whose structure follows the definition in (2.1), and the covariances are given as inputs to the collision detection subsystem as:

$$\tilde{\boldsymbol{x}}_{\boldsymbol{EGO}}(i) = \begin{bmatrix} x(i) \\ y(i) \\ \theta(i) \\ v(i) \\ \vdots \end{bmatrix} = \begin{bmatrix} xEGO(i) \\ yEGO(i) \\ \theta_{EGO}(i) \\ v_{predictions}(i) \\ \vdots \end{bmatrix}, \tag{4.17}$$

$$\tilde{\boldsymbol{P}}_{\boldsymbol{EGO}}(i) = \begin{bmatrix} \sigma_x^2(i) & \sigma_{x,y}(i) & \vdots \\ \sigma_{x,y}(i) & \sigma_y^2(i) & \vdots \\ & & \ddots \; \vdots \\ \cdots & \cdots & \cdots \end{bmatrix} = \begin{bmatrix} \sigma_{x_{EGO}}^2(i) & \sigma_{x,y_{EGO}}(i) & \vdots \\ \sigma_{x,y_{EGO}}(i) & \sigma_{y_{EGO}}^2(i) & \vdots \\ & & \ddots \; \vdots \\ \cdots & \cdots & \cdots \end{bmatrix} \tag{4.18}$$

$$\tilde{\boldsymbol{x}}_{\boldsymbol{POV}}(i) = \begin{bmatrix} x(i) \\ y(i) \\ v_x(i) \\ v_y(i) \\ \vdots \end{bmatrix} = \begin{bmatrix} xPOV(i) \\ yPOV(i) \\ v_{x_{POV}}(i) \\ v_{y_{POV}}(i) \\ \vdots \end{bmatrix}, \tag{4.19}$$

$$\tilde{\boldsymbol{P}}_{\boldsymbol{POV}}(i) = \begin{bmatrix} \sigma_x^2(i) & \sigma_{x,y}(i) & \vdots \\ \sigma_{x,y}(i) & \sigma_y^2(i) & \vdots \\ & & \ddots \; \vdots \\ \cdots & \cdots & \cdots \end{bmatrix} = \begin{bmatrix} \sigma_{x_{POV}}^2(i) & \sigma_{x,y_{POV}}(i) & \vdots \\ \sigma_{x,y_{POV}}(i) & \sigma_{y_{POV}}^2(i) & \vdots \\ & & \ddots \; \vdots \\ \cdots & \cdots & \cdots \end{bmatrix} \tag{4.20}$$

where the estimated positions with their respective part in the covariance matrix are collected and thereafter defined for the EGO and POV as:

$$E(i) \sim \mathcal{N}\left(\overbrace{\begin{bmatrix} xEGO(i) \\ yEGO(i) \end{bmatrix}}^{\boldsymbol{m_E}(i)}, \overbrace{\begin{bmatrix} \sigma^2_{x_{EGO}}(i) & \sigma_{x,y_{EGO}}(i) \\ \sigma_{x,y_{EGO}}(i) & \sigma^2_{y_{EGO}}(i) \end{bmatrix}}^{\boldsymbol{\Sigma_E}(i)}\right) \tag{4.21}$$

$$O(i) \sim \mathcal{N}\left(\overbrace{\begin{bmatrix} xPOV(i) \\ yPOV(i) \end{bmatrix}}^{\boldsymbol{m_O}(i)}, \overbrace{\begin{bmatrix} \sigma^2_{x_{POV}}(i) & \sigma_{x,y_{POV}}(i) \\ \sigma_{x,y_{POV}}(i) & \sigma^2_{y_{POV}}(i) \end{bmatrix}}^{\boldsymbol{\Sigma_O}(i)}\right) \tag{4.22}$$

Since the global sets are rotated according to predicted heading, as stated in Section 4.2, both vehicles heading must be known. As seen in (4.17) and (4.19), the EGO's heading can be collected directly whereas the POV's heading needs to be calculated with respect to its values of $v_{x_{POV}}$ and $v_{y_{POV}}$.

The EGO's velocity, $v_{predictions}$, for all prediction steps is also collected from (4.17) since the velocity is needed for the *Collision Avoidance* subsystem.

4.5.2 Characteristics of a collision

A vector defining the probability of collision in each prediction step i is given by (4.16) for $n_{predictions}$ predictions as:

$$\mathcal{P}_{collision} = \begin{bmatrix} \mathcal{P}_{collision}(1) & \mathcal{P}_{collision}(2) & \cdots & \mathcal{P}_{collision}(n_{predictions}) \end{bmatrix} \tag{4.23}$$

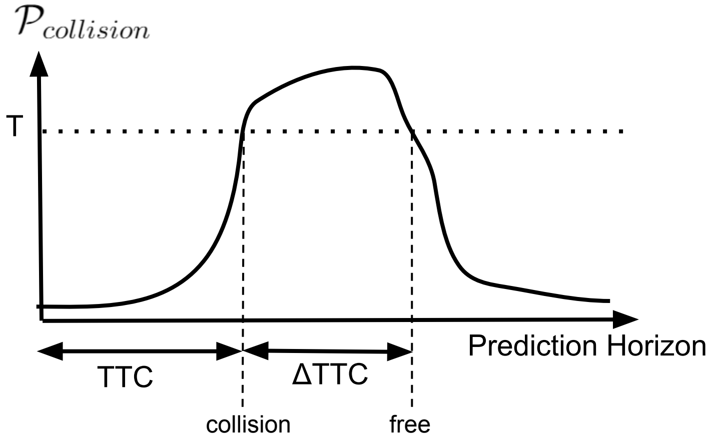


Figure 4.5: Illustration of the definition of Time To Collision (TTC) and collision time interval (ΔTTC), where the y-axis denotes the probability of collision, $\mathcal{P}_{collision}$, over the predicted time horizon. TTC is calculated from the current time step and up to the probability to be above a threshold T, whereas ΔTTC is calculated as the first time the probability is above the threshold until is it below the same mentioned threshold.

This vector will result in a curve, an example depicted in Figure 4.5. The vector can be used to define the characteristics of a collision (i.e. Time To Collision (TTC), collision time interval (ΔTTC) and Distance To Collision (DTC)) needed as inputs to the collision avoidance subsystem. As illustrated in Figure 4.5, a collision is detected when $\mathcal{P}_{collision}$ is above the threshold T. Moreover, the TTC is calculated as the first time step when the probability crosses the threshold, whereas the ΔTTC is the time difference between TTC and the time when $P_{collision}$ goes below the previously mentioned threshold. This can mathematically be described as:

$$P_{collision}(i_{collision}) \geq T \tag{4.24}$$
$$P_{collision}(i_{free}) < T \quad \wedge \quad i_{free} > i_{collision} \tag{4.25}$$
$$t_C = i_{collision} \cdot \Delta T \tag{4.26}$$
$$\Delta t_C = (i_{free} - i_{collision}) \cdot \Delta T \tag{4.27}$$

where $i_{collision}$ denotes the first time step when the probability of collision is above the threshold T and i_{free} the first time step where the probability of collision again reaches values lower than T. Furthermore, ΔT is the time interval between each prediction, t_C and Δt_C denote the TTC and the ΔTTC, respectively.

Additionally, the velocity of the EGO in each prediction step, $v_{predictions}$, and the DTC, d_C, as well as the corresponding variance, $\sigma^2_{d_C}$, are needed to characterise a collision. As mentioned in Section 4.5.1, the velocity is collected directly from the predictions, but the DTC needs to be computed appropriately.

The DTC is defined as the distance the EGO will travel before it reaches the collision point and is calculated as the euclidean distance between two consecutive points. The DTC from the EGO's current position until it reaches prediction time step i can then be calculated by:

$$d_C(0) = 0 \tag{4.28}$$
$$d_C(i) = d_C(i-1) + \sqrt{\Delta xEGO^2 + \Delta yEGO^2} \qquad \text{for } i = 1, 2, ..., n_{predictions} \tag{4.29}$$

with:

$$\Delta xEGO = xEGO(i) - xEGO(i-1) \tag{4.30}$$
$$\Delta yEGO = yEGO(i) - yEGO(i-1) \tag{4.31}$$

where $xEGO$ and $yEGO$ are the cartesian coordinates for the EGO.

To transform a two-dimensional variance of $xEGO$ and $yEGO$ into a combined variance in the one-dimensional space for the DTC, the Jacobian $G(i)$ of $d_C(i)$, and the variance $P_{var}(i)$ need to be defined as:

$$G(i) = \begin{bmatrix} \frac{\Delta xEGO}{h(x,y)} & \frac{\Delta yEGO}{h(x,y)} & \frac{-\Delta xEGO}{h(x,y)} & \frac{-\Delta yEGO}{h(x,y)} \end{bmatrix} \tag{4.32}$$

where:

$$h(x,y) = \sqrt{\Delta xEGO^2 + \Delta xEGO^2} \tag{4.33}$$

and:

$$P_{var}(i) = \begin{bmatrix} \Sigma_E(i) & 0 \\ 0 & \Sigma_E(i-1) \end{bmatrix} \tag{4.34}$$

where $\Sigma_E(i)$ and $\Sigma_E(i-1)$ are the variances in x and y position for prediction step i and $i-1$ respectively, according to (4.21). The resulting variance for the DTC can then finally be calculated according to:

$$\sigma^2_{d_C}(i) = G(i) \cdot P_{var}(i) \cdot G(i)^T \tag{4.35}$$

which will serve as an input to the collision avoidance subsystem.

4.5.3 Collision Detection Algorithm

The presented collision detection algorithm will have inputs as defined in Section 4.5.1, as well as the length and width of both vehicles. An additional input is the time interval between each prediction, ΔT. Outputs will be the variables described in Section 4.5.2, as well as an additional boolean variable *Danger* being set to *true* if a risk of collision is detected.

Note that when a collision risk has been detected, the predictions will maybe not include a time step where the probability of collision is below the predefined threshold T. The index i_{free} is then consequently not defined, which is required in the presented solution as was seen in (4.27). This index will therefore, in these situations, be set to the time step after the prediction horizon ($n_{predictions} + 1$).

Algorithm 5 Collision detection

```
Input: $\tilde{\boldsymbol{x}}_{\boldsymbol{EGO}}, \tilde{\boldsymbol{P}}_{\boldsymbol{EGO}}, l, w, \tilde{\boldsymbol{x}}_{\boldsymbol{POV}}, \tilde{\boldsymbol{P}}_{\boldsymbol{POV}}, \bar{l}, \bar{w}, \Delta T$
Output: $Danger, t_C, \Delta t_C, d_C, \sigma^2_{d_C}, v_{predictions}$

Define the sets $A_{EGO}$ and $A_{POV}$ according to (4.1) and (4.2)
Collect $E$ and $O$ according to (4.21) and (4.22)
Collect $\theta_{EGO}$ from $\theta$ and $\theta_{POV}$ from $v_x$ and $v_y$ in (4.17) and (4.19)

Initialise: $d_C(0) = 0$
for $i = 1 : n_{predictions}$ do
    Define global sets according to (4.3) and (4.4)
    Rotate sets according to $\theta_{EGO}$ and $\theta_{POV}$, choose $\eta$ points i.e. create $Ep$ and $Op$
    for $j = 1 : \eta$ do
        Create the distributions according to (4.12) and (4.13)
        Create the vector $\mathcal{P}_{1,2}$ by the use of (4.14)
    end for
    Calculate $\mathcal{P}_{collision}$ according to (4.16)
    Calculate $d_C$ and $\sigma^2_{d_C}$ according to (4.29) to (4.35)
end for

if $any(P_{collision}) > T$ then
    $i_{collision} = find(\mathcal{P}_{collision} \geq T, 1)$
    $i_{free} = find(\mathcal{P}_{collision} < T)$
    $tempindex = find(i_{free} > i_{collision})$

    If no collision free index is found along the predictions
    if $isempty(tempindex)$ then
        $i_{free} = n_{predictions} + 1$
    else
        $i_{free} = i_{free}(tempindex(1))$
    end if
    $t_C = i_{collision} \cdot \Delta T$
    $\Delta t_C = (i_{free} - i_{collision}) \cdot \Delta T$
    Collect $\boldsymbol{v}_{\boldsymbol{predictions}}$ from (4.17)
    $Danger = true$
    $d_C = d_C(i_{collision})$
    $\sigma^2_{d_C} = \sigma^2_{d_C}(i_{collision})$
else
    $Danger = false$
end if
return $Danger, t_C, \Delta t_C, d_C, \sigma^2_{d_C}, v_{predictions}$
```

5
Collision avoidance

A Collision Avoidance (CA) system consists of two major parts, detection of an oncoming threat and an avoidance decision in concern to that threat. Since the detection of a probable threat is described in Chapter 4, this chapter focuses on the assessment and decision made in regard to this threat.

The assessment should aim at finding the point in time where the driver is unable to avoid a collision. Within the automotive industry, a false intervention is looked on with disapproval, hence the biggest difficulty when assessing a threat is to find a good level of conservativeness. This level is found when having a balance of preventing a too high level of false interventions meanwhile having a fully functional system. The system will therefore need to cope with these requirements[1] i.e., keeping the false interventions to a minimum.

A decision should thereafter culminate into an intervention aiming at avoiding or mitigating the collision, where an intervention could be comprised of different motions such as braking, steering or acceleration. Since this Thesis only concern *Autonomous Emergency Braking* (AEB) interventions, no further research will be conducted within the area of incorporating a choice for different intervention methods.

Figure 5.1 gives an overview of the collision avoidance part, blue encirclement, in the whole system. The figure also illustrates the intended intervention method in accordance with the formal threat assessment, as shown in the graphical representation. The chapter comprises of the related work within this area given in Section 5.1, the proposed collision avoidance strategy in Section 5.2 and applied collision avoidance procedure in Section 5.3.

5.1 Related work

CA systems have been extensively researched within the field of fully autonomous vehicles such as for cooperative collision avoidance solutions. Here, a scenario can involve several vehicles in motion, where the assessment and decision will aim to cooperatively calculate each vehicle's trajectory under tight time constraints for the ability to avoid a collision. Solutions based on this approach can be viewed in [31][33][34].

Since the intended system aims at supporting the driver and the Decision

[1] ISO26262 requires verification tests to give less than one false AEB intervention during 10 years or within 100 000 km of driving.

Making (DM) algorithm is restricted to emergency braking (AEB) as intervention method, the Threat Assessment (TA) algorithm gains a greater purpose in the proposed solution. The aim of the TA algorithm is to trigger an intervention at the latest possible time instance, hence keeping the driver as the decision making constituent as long as possible.

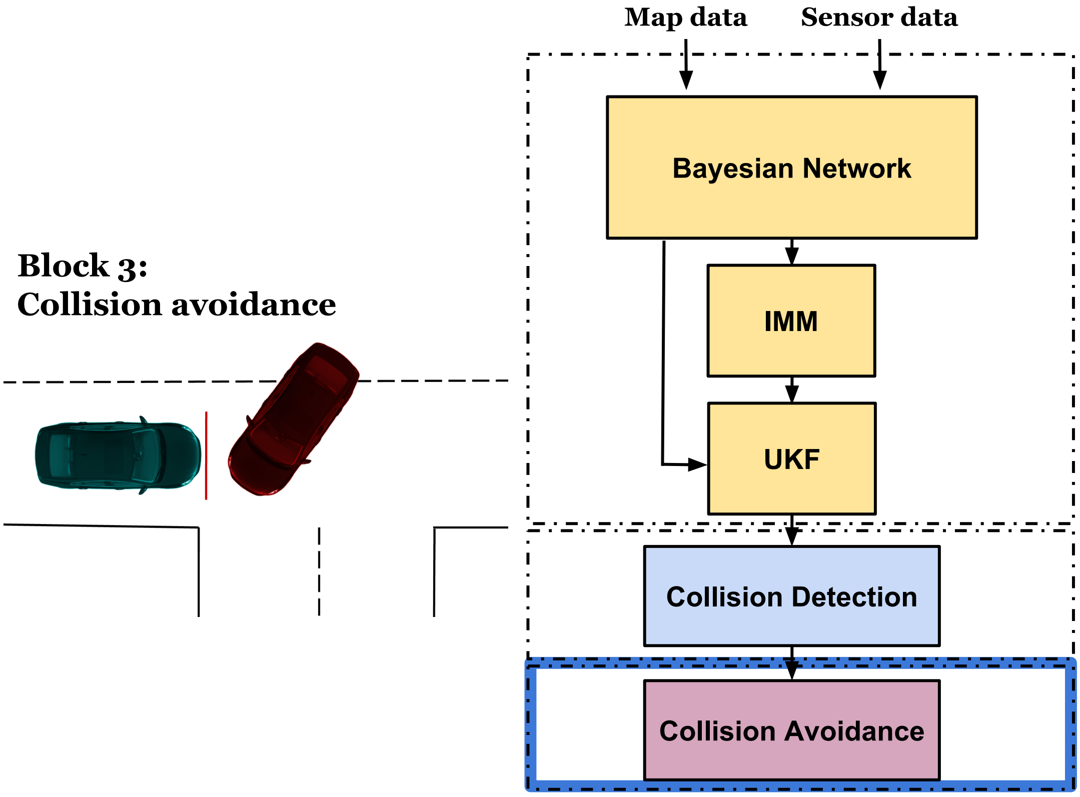


Figure 5.1: Overview of the collision avoidance subsystem as part of the whole system with graphical view of the intended outcome.

A well-known concept within active safety applications is the *Point-Of-No-Return* (PNR) which corresponds to the time instance where the accounted vehicle inevitably ends up in a collision. It is consequently this point in time that will need to be avoided to evade or, if this point is reached, mitigate a collision. The aim of the CA system should therefore be to identify the PNR and consequently intervene just before, with the use of a pre-calculated control input, with as few false interventions as possible.

In literature there are several approaches based on the same avoidance strategy as the above mentioned, presented in [35][36][37]

A way of calculating the PNR, is by the use of reachability analysis. Reachability tools can be used to evaluate if the current state belongs to a set that over a specific prediction horizon evolves into a given target set for all feasible external inputs. In the adaption of identifying the PNR, the reachability analysis has the purpose of, by back propagation, finding the time instance where the vehicle inevitable ends up in a collision. The collision instant will here act as the target set and the reachability tools will define the back propagated sets. The propagated sets will thereafter describe the state in each time instance which consequently could be used to identify the PNR.

This approach has mainly, within the automotive industry, been applied to research regarding fully automated vehicles, as for example within cooperative driving as presented in [38] and [39]. As a result, instead of making a collision avoiding intervention, they use a decision parameter for cooperative crossing of an intersection. The approach presented in [17] reformulated the solution of [38] by defining the back propagated sets as the relation of maximum velocity at a certain distance to collision. Essentially, a comparison were performed between the current state of the EGO and each back propagated state, thus working in the same fashion as a look-up table. They also incorporated emergency braking (AEB) intervention to be triggered just before the PNR.

The approach presented in [17] will work as an inspirational basis for the proposed system with additional adjustments made in accordance with the previously presented subsystems in Chapter 3 and 4.

5.2 Collision avoidance definition

When the collision detection algorithm forwards a signal of an imminent threat, the main objective of the subsystem stated in this chapter is to trigger an intervention with the aim of avoiding this threat. With state predictions and accompanied uncertainties given along the whole prediction horizon, the primary part of the CA system is to determine when to trigger an emergency braking.

The *Collision Detection* subsystem provides the distance and time to detected collision by comparison of each vehicle's propagated trajectory along the prediction horizon. In the scope of this work, in order to avoid a collision, this can therefore be defined as keeping the EGO from travelling the *distance to collision* (DTC), d_C, within the time span of *time to collision* (TTC), t_C and the collision time interval Δt_C. In addition to t_C and Δt_C, for the ability to compute an accurate time left to required emergency braking intervention, a vehicle brake's actuation time need to be taken into consideration. Here, the actuation time t_B comprises of the brake's pressure build up phase as well as the triggering time of a braking intervention.

The collision is then avoided by back propagation of the EGO's predicted velocity $v_{predictions}$ and DTC over the prediction horizon. If $v_{predictions}$ at a certain distance d_C is detected to be above or at the maximum allowed velocity v_{max} given for the system, a braking intervention is triggered. Due to uncertainties in both velocity and position, a way of including those uncertainties is by the use of *Robust Controllable Sets* as presented in [40].

5.2.1 Robust controllable sets

In the scope of this thesis, when integrating the Robust Controllable sets, the systems considered are autonomous systems with additive disturbances. Therefore only the theory and formulations concentrated to these types of systems will be covered in this section.

An autonomous system f_a, can formally be described as:

$$\mathbf{x}(k+1) = f_a(\mathbf{x}(k), w(k)) \tag{5.1}$$

where w is a disturbance which, together with the state $\mathbf{x}$, is subject to constraints as:

$$\mathbf{x}(k) \in \mathcal{X}, w(k) \in \mathcal{W}, \forall k \geq 0 \tag{5.2}$$

where both $\mathcal{X}$ and $\mathcal{W}$ will have the geometrical shape of a polyhedron.

The set of states that could evolve into the target set, $\mathcal{S}$, in one time step are called the *pre-set* to the target set, denoted by $\text{Pre}(\mathcal{S}, \mathcal{W})$, and given as:

$$Pre(\mathcal{S}, \mathcal{W}) = \{\mathbf{x} \in \mathbb{R}^n : f_a(\mathbf{x}, w) \in \mathcal{S} \text{ s.t. } \forall w \in \mathcal{W}\} \tag{5.3}$$

The set $\text{Pre}(\mathcal{S}, \mathcal{W})$ defines the set of arbitrary system states, given by (5.1), to consequently evolve into the predefined target set $\mathcal{S}$ in one time step for all possible disturbances $w \in \mathcal{W}$. An illustration of the one-step Robust Controllable set $\text{Pre}(\mathcal{S}, \mathcal{W})$ can be viewed in Figure 5.2, which depicts the subjected constraints, red color, as disregarded areas.

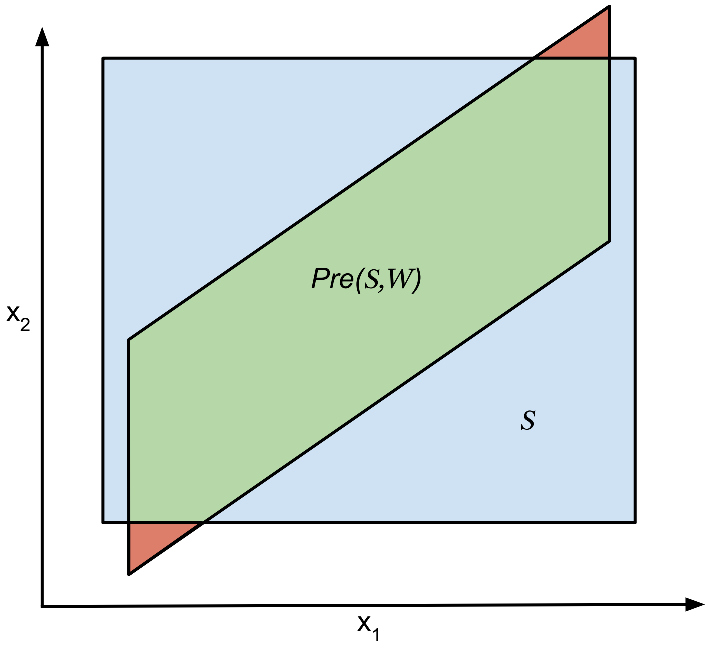


Figure 5.2: Illustration of the one-step Robust Controllable set, $\text{Pre}(\mathcal{S}, \mathcal{W}) \cap \mathcal{S}$, representing the set $\text{Pre}(\mathcal{S}, \mathcal{W})$ to be evolved into the target set $\mathcal{S}$ subject to state constraints.

The N-step *Robust Controllable Set* $\mathcal{K}_N(\mathcal{S}, \mathcal{W})$ could then, by iterating (5.3) given a target set $\mathcal{S} \subseteq \mathcal{X}$, be derived as:

$$\mathcal{K}_\Upsilon(\mathcal{S}, \mathcal{W}) = Pre(\mathcal{K}_{\Upsilon-1}(\mathcal{S}, \mathcal{W}), \mathcal{W}) \cap \mathcal{X},\ \mathcal{K}_0(\mathcal{S}, \mathcal{W}) = \mathcal{S},\ \Upsilon \in \{1, ..., N\} \tag{5.4}$$

This means that all states $\mathbf{x}$, given from (5.1), belonging to the N-step *Robust Controllable Set* $\mathcal{K}_N(\mathcal{S}, \mathcal{W})$ could evolve, for all possible disturbances $w \in \mathcal{W}$, into the target set $\mathcal{S}$ in N steps, while satisfying the constraints stated in (5.2).

5.2.2 Derivation of the attraction sets

In the scope of this Thesis, the notion of Robust Controllable sets is exploited by considering a control input as the defined disturbance. This will thus constitute as a special case of a Robust Controllable set denoted as an *Attraction set*. The derivation of the *Attractions sets* for the intended application is inspired by [38] and [17].

The system, denoted in (5.1), will be reformulated as a linear system with a control input instead of disturbance in accordance with the *Attraction set* definition. This yields the arbitrary system for each time step as:

$$\mathbf{x}_{CA}(k+1) = A\mathbf{x}_{CA}(k) + Bu(k), \text{ with: } \mathbf{x}_{CA}(k) = \begin{bmatrix} d_C(k) \\ v_{predictions}(k) \end{bmatrix} \tag{5.5}$$

where d_C describes the distance to the imminent collision and $v_{predictions}$ the EGO's predicted velocity at each dedicated time step.

The target set $\mathcal{S}$, denoted in (5.3), can be formulated as:

$$\mathcal{S} = \left\{ \begin{matrix} d_C(k): & 0 & \leq & d_C(k) & \leq & d_{max} \\ v_{predictions}(k): & 0 & \leq & v_{predictions}(k) & \leq & v_{max} \end{matrix} \right\}, \forall k \geq 0 \tag{5.6}$$

where v_{max} represents the maximum allowable velocity given for the system and d_{max} the maximum possible distance travelled during a full prediction horizon respectively. The target set thus defines a collision to occur at distance $d_C(k) = 0$

In the derivation of the *Attraction sets*, the system will be subject to state constraints given as:

$$\mathbf{x}_{CA}(k) \in \mathcal{X} = \left\{ \begin{matrix} d_C(k): & -d_{max} & \leq & d_C(k) & \leq & d_{max} \\ v_{predictions}(k): & 0 & \leq & v_{predictions}(k) & \leq & v_{max} \end{matrix} \right\} \tag{5.7}$$

While the control input is defined to belong to the set:

$$u(k) \in \mathcal{U} = \left\{ u(k): \ -a_{max} \ \leq \ u(k) \ \leq \ 0 \right\}, \tag{5.8}$$

Here the control input states the possible actions to be made, hence when solely considering deceleration (emergency braking) interventions, trajectories requiring acceleration to avoid a collision is excluded (upper limit being zero). The lower limit $-a_{max}$, is set as the maximum possible deceleration by the emergency braking system (AEB).

The *Attraction Sets* $\mathcal{A}$, can iteratively be derived using the formulation described in Section 5.2.1 with the system and target set as defined in (5.5) to (5.8). These sets will span all possible configurations of velocity and distance that, for the given deceleration, unavoidably ends up in a collision. The sets can be denoted as:

$$\mathcal{A} = \{\mathcal{A}_1, ..., \mathcal{A}_L\} \tag{5.9}$$

As illustrated in Figure 5.3 the sets will decrease in size, thus the set $\mathcal{A}_L$ denotes the last set inevitably ending up in a collision, hence further away will the sets be empty.

The derived *Attraction Sets* are finally computed using a *Multi-Parametric Toolbox* (MPT) for Matlab [41].

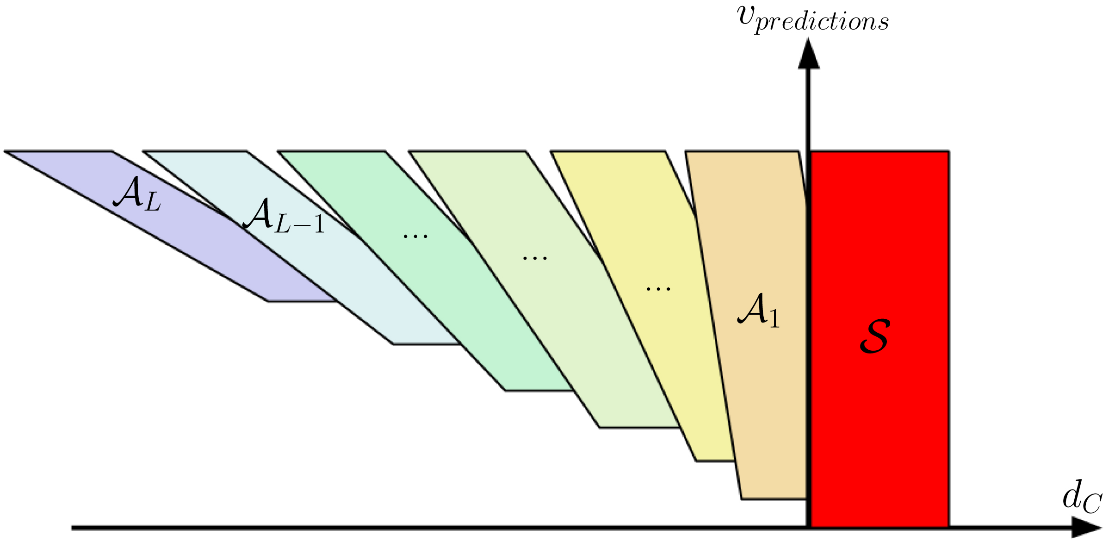


Figure 5.3: Definition of the target set $\mathcal{S}$, and the back propagated *Attraction sets* $\mathcal{A}_\zeta$ for ζ = 1,...,L. The set $\mathcal{A}_L$ denotes the last set inevitably ending up in a collision, hence further away from the target set the sets will be empty.

5.2.3 Formal definition of the collision avoidance constraints

As stated in the beginning of Section 5.2, the main objective for the *Collision Avoidance* subsystem is to keep the EGO from travelling d_C, within the time span of t_C and Δt_C. This can be translated as assuring the EGO's future trajectory to not intersect another vehicle's area, hence ending up in a collision.

If the vehicles involved in a collision could be assumed to have known trajectories driven with constant speed, it would simplify the determination of if the two vehicles were on trajectories leading to a collision. Recall that the definition of the ζth *Attraction set* is to span the configuration of distance left to collision and velocity that in ζ time steps, for all admissible inputs, inevitably ends up in a collision. Using this definition, a formulation of if and when to trigger a braking intervention can be made. Hence, if it is known that the state of a vehicle in one time step from the current time k will belong to the attraction set $\mathcal{A}_{t_C}$ leading to a collision, then a braking intervention should be triggered immediately. Here, $\mathcal{A}_{t_C}$ represents being t_C time steps away from the collision area described by the target set $\mathcal{S}$. This condition can therefore be formally defined as:

$$\mathbf{x}_{CA}(k+1) \in \mathcal{A}_{t_C} \tag{5.10}$$

To assume trajectories with constant speed is however in most cases not a valid hypothesis. It is thus necessary to account for unpredictable driver behaviour beyond the current time step. Since (5.7) does not incorporate driver acceleration or steering, there is consequently a possibility for the EGO to progress into any part of the collision area spanned by the time to collision t_C and the additional duration of collision $t_C + \Delta t_C$. This is formally defined as the states of the EGO that did not belong to the attraction set $\mathcal{A}_{t_C}$, to instead progress into any of the subsequent attraction sets between $\mathcal{A}_{t_C}$ to $\mathcal{A}_{t_C+\Delta t_C}$. This is illustrated in Figure 5.4.

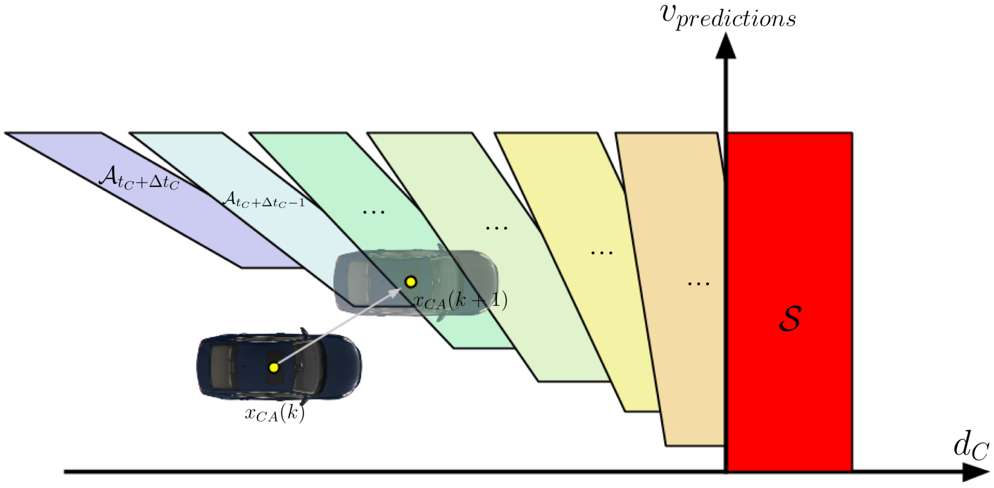


Figure 5.4: Illustration of an unpredictable behaviour such as acceleration or turning manoeuvre that could lead the EGO from being on a *clear path* to an imminent collision.

To ensure a correctly triggered intervention, the safety membership test thus need to ensure that the vehicle will not belong to any of the intermediate attraction sets between $\mathcal{A}_{t_C}$ and $\mathcal{A}_{t_C+\Delta t_C}$. In regard to (5.10), this can be reformulated as:

$$\mathbf{x}_{CA}(k+1) \in \mathcal{A}_{t_C+\tau}, \text{ for: } \tau = 0, ..., \Delta t_C \tag{5.11}$$

Both (5.10) and (5.11) are however assuming that the vehicle will be able to avoid entering an attraction set in just one time step i.e., assuming immediate actuation which is not a realistic assumption. Hence to make (5.11) accurately ensure safety, a consideration of the vehicle brake's actuation time t_B is required. Recall that t_B is comprised of the brake's pressure build up time as well as the triggering time of a braking intervention.

For the safety critical membership test to be properly defined, the vehicle state propagated $t_B + 1$ time steps from the current time k will need to be evaluated of belonging to an attraction set leading to a collision. The attraction sets included into the safety membership test will thus also need to be translated in accordance with the brake's actuation time. This can be defined as:

$$\mathbf{x}_{CA}(t_B+1) \in \mathcal{A}_{t_C+\tau-(t_B+1)}, \text{ for: } \tau = 0, ..., \Delta t_C \tag{5.12}$$

Hence the vehicle's propagated states at the time step of $t_B + 1$ is evaluated of belonging to any of the attraction sets from $\mathcal{A}_{t_C-(t_B+1)}$ to $\mathcal{A}_{t_C+\Delta t_C-(t_B+1)}$. For the sake of clearness, see footnote[2] for an example.

[2] If $t_B = 5$, $t_C = 10$ and $\Delta t_C = 4$ (note that all these values are given as number of time steps). Then (5.12) checks if the vehicle's states at time step 5+1=6 belongs to any of the *Attraction sets* between the time steps of 10-(5+1)=4 to 10+4-(5+1)=8 ($\mathcal{A}_4$ to $\mathcal{A}_8$)

5.2.4 Integration of uncertainties

Equation (5.12) denotes the condition of establishing if the current time k is the time step just before the PNR. If so, this should trigger an emergency braking intervention. However, as previously mentioned, false interventions are looked on with disapproval within the automotive industry, hence a correctly triggered intervention need to be assured. Thus, if measurement uncertainties are apparent, these will also need to be included into the safety membership test.

When using robust controllable sets as a tool, all the system uncertainties could be incorporated as a disturbance $\mathcal{W}$, described in Section 5.2.1. In the derivation of the *Attraction sets*, the properties of both the state and control input constraints are incorporated. Therefore, when altering the limits of these constraints, the size of each iterative attraction set changes accordingly. As an example, if the vehicle has the ability of braking with a higher deceleration, it will cause the sets to be more narrow hence having the possibility to avoid a collision later in time, as illustrated in Figure 5.5.

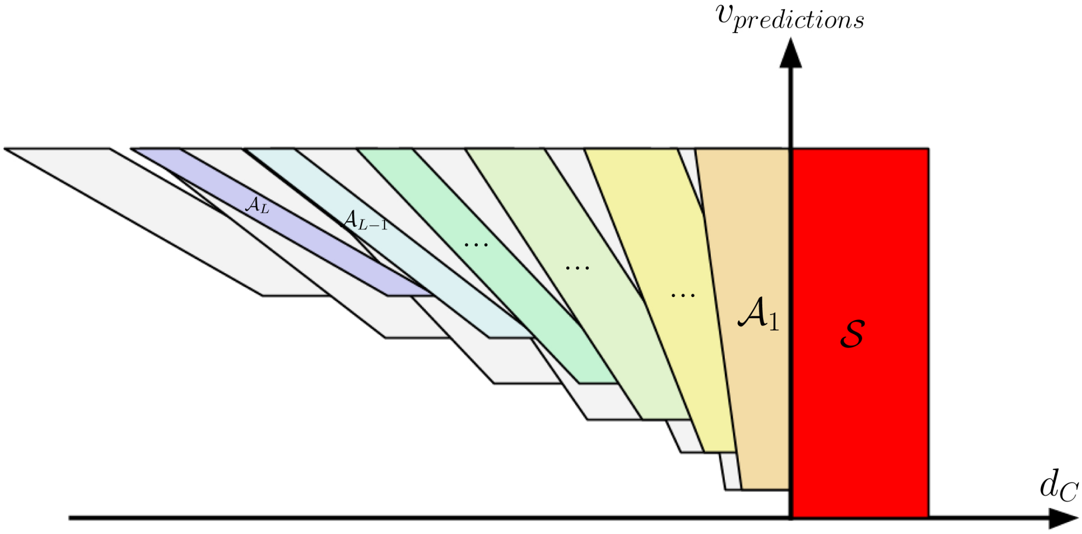


Figure 5.5: Illustration of the effect of altering the control input constraint's $\mathcal{U}$ limits in comparison to the spread of the *Attraction sets*. In this specific example, the coloured sets are derived for a higher deceleration.

Since the attraction sets, in some applications, could be too computational heavy to compute online they are instead required to be derived offline. An a priori estimate of the included disturbances is then needed. Including the uncertainties in DTC are difficult due to the transformation of the uncertainties in two dimensions, x and y, into the one dimensional space as seen in (4.28) to (4.35). This will therefore most likely result in a too conservative estimate of when to trigger an intervention. The uncertainties in DTC can instead be accounted for by stochastically incorporate these into the safety membership test. As given by (4.28) to (4.35), the estimated distance to collision is defined as:

$$\hat{d}_C \sim \mathcal{N}(d_C, \sigma^2_{d_C}) \tag{5.13}$$

The probability of belonging to a certain *Attraction set* $\mathcal{A}_{t_C+\tau-(t_B+1)}$ can then be formulated as:

$$\mathcal{P}_{Belong} = \int_{\mathcal{A}_{t_C+\tau-(t_B+1)}} \hat{d}_C \, \mathrm{d}\mathcal{A}_{t_C+\tau-(t_B+1)}, \text{ for: } \tau = 0, ..., \Delta t_C \tag{5.14}$$

where the integral can be numerically approximated by uniformly distributing p points along a 3σ-region around each DTC estimate as illustrated in Figure 5.6.

A vector spanning the uniformly distributed p points along the 3σ-region around each estimate can be defined as:

$$\vec{D}_{i_p} = \alpha_D + (i_p - 1) \cdot \epsilon \tag{5.15}$$

where $\alpha_D = -3\sigma_{d_C}$, $\epsilon = \frac{6\sigma_{d_C}}{p-1}$ and $i_p = 1, ..., p$. Evaluating a point along the 3σ-region to belong to a certain *Attraction set* $\mathcal{A}_{t_C+\tau-(t_B+1)}$ in the region of $\tau =$ 0,...,Δt_C, can then be done by:

$$\vec{M}_{i_p} = \begin{cases} 1, & \text{if } \vec{D}_{i_p} \in \mathcal{A}_{t_C+\tau-(t_B+1)}, \text{ for } i_p = 1, ..., p \\ 0, & \text{otherwise} \end{cases} \tag{5.16}$$

which results in a vector containing either a one or a zero at each index i_p. The vector of ones and zeros thereafter need to be compared to a vector $\vec{R}$ describing the probability of randomly picking a value from the standard normal distribution. The normal distribution can be divided into segments describing the approximate percentage of the area lying under the curve between the standard deviations as illustrated in Figure 5.7.

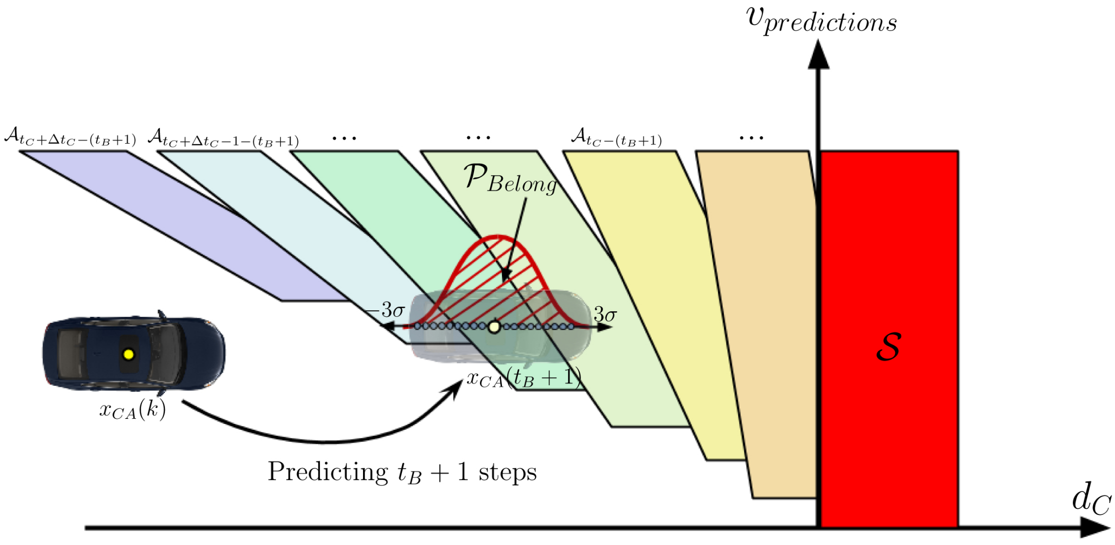


Figure 5.6: Illustration of how the safety membership is made for the predicted EGO's states $t_B + 1$ time steps into the future. The uncertainties is here incorporated into the membership tests of the *Attraction sets* $\mathcal{A}_{t_C+\Delta t_C-(t_B+1)}$ to $\mathcal{A}_{t_C-(t_B+1)}$, by spanning up the 3σ-region for a normal distribution centered at the estimated DTC. If the probability given by $\mathcal{P}$ is higher than the predefined threshold T_{CA}, then the intervention is triggered.

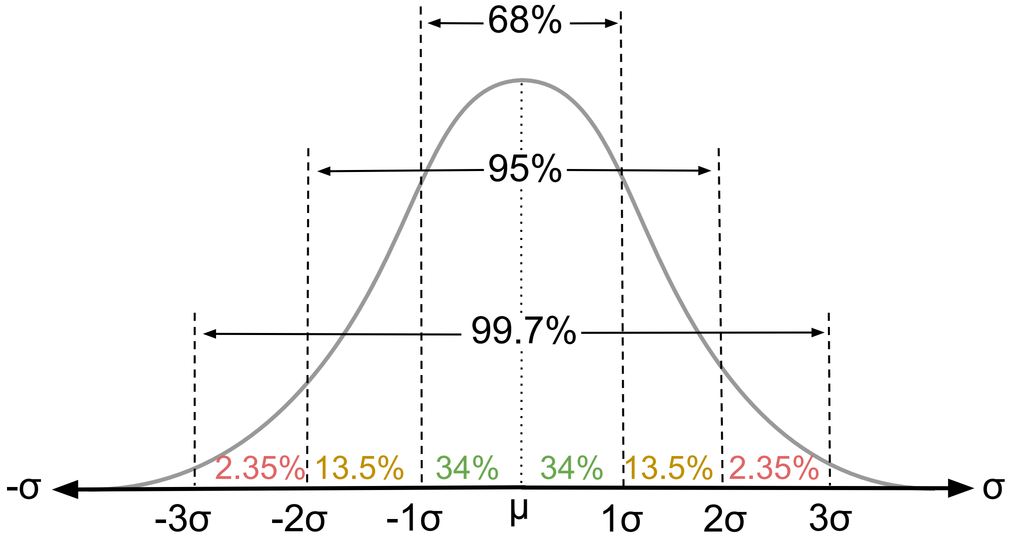


Figure 5.7: Illustration of the normal distribution divided by the 3σ-region describing the approximate percentage of the area lying under the curve between the standard deviations.

The vector $\vec{R}$, should therefore span the probability along the normal distribution, in the same fashion as (5.15), according to:

$$\vec{R}_{i_p} = \beta_R + (i_p - 1) \cdot \gamma \tag{5.17}$$

with $\beta_R = -3\sigma$ and $\gamma = \frac{6\sigma}{p-1}$ where $i_p = 1, ..., p$.

The membership tests giving the probability of belonging to a certain *Attraction set* i.e. calculating the numerical approximation of the integral, is then given as:

$$\mathcal{P}_{Belong} = \vec{M} \cdot \vec{R} \tag{5.18}$$

This probability will thereafter need to be compared to a predefined threshold, T_{CA}, to determine the reliability of the timing for an intervention, before making the actual intervention. The threshold will therefore control how certain the CA system should be before applying the emergency braking intervention and should be set high enough to avoid unnecessary false interventions.

5.3 Collision Avoidance Procedure

The *Collision Avoidance* subsystem requires the d_C, with estimated uncertainties $\sigma^2_{d_C}$ as well as the t_C and the predicted velocities for the EGO along the whole prediction horizon $v_{predictions}$ as inputs given from the *Collision Detection* subsystem. The CA system also requires the prediction time step ΔT.

The inputs will be used to make a back propagation from the *Target set* $\mathcal{S}$, through the *Attraction sets*, $\mathcal{A}$. The aim of the back propagation is to find the PNR, hence gaining information of at which time step the intervention should be triggered. The identification of the PNR incorporates both a vehicle brake's presumed pressure build up phase as well as the triggering time of the brakes. Thus, when the brakes are applied, the vehicle's future behaviour is anticipated and incorporated into the time step of the triggered intervention.

5.3.1 Collision Avoidance Algorithm

The algorithm described in this section concerns the procedure explained in the introductory text to this section as well as the mathematical formulations described in Section 5.2. As mentioned in Section 5.2.2, the *Attraction sets* is to be calculated offline prior to the given pseudo code with the toolbox *Multi-Parametric Toolbox* for Matlab.

Algorithm 6 Collision avoidance algorithm

```
**Input:** $t_C, \Delta t_C, d_C, \sigma^2_{d_C}, \Delta T, v_{predictions}$
**Output:** $Brake$

**Initialise:**
$t_L$ = Prediction horizon considered as the limit
$t_B$ = Brakes initiation time
$\mathcal{A}$ = All *Attraction Sets*
$T_{CA}$ = Membership probability threshold
**for** $\tau = 0 : \Delta t_C$ **do**
    $\mathcal{P}_{Belong}$ = MembershipTest$(t_C, d_C, \sigma^2_{d_C}, v_{predictions}, \mathcal{A}_{t_C+\tau-(t_B+1)}, t_B)$
    **Calculated according to** (5.18)

    **if** $\mathcal{P}_{Belong} > T_{CA}$ **then**
        $Brake = true$
        $break$
    **else**
        $Brake = false$
    **end if**
**end for**
**return** $Brake$
```

6
Results and analysis

The focus of the developed system has been on gaining a robust collision avoidance system able to cope with accident prone environments as in urban intersections. The main objective during the development was to answer the stated problems described in Section 2.1, with limitations as defined in Section 2.2.

The evaluation of the resulting three-block system will be described in this Chapter, where an overview of the complete system's functionality will firstly be given. Thereafter will the three subsystems *Long-term Path Prediction*, *Collision Detection* and *Collision Avoidance* be evaluated more closely. Alongside the evaluation of the systems ability to cope with the desired properties, a robustness analysis will also be presented.

6.1 System parameters

The defined system parameters in this section will be set static throughout the collision avoidance (CA) system, if not stated differently for a specific scenario. Some parameters are set irrespectively of external influences with focus on gaining a general and robust solution, whilst some parameters are directly dependent on which intended vehicle the system is to be implemented in.

The system parameters have been divided into:

- Time parameters e.g. the time step, ΔT and the prediction horizon, $T_{predictions}$.
- Thresholds for the collision detection as well as the collision avoidance subsystem, T_{CD} and T_{CA}, described in Chapters 4 and 5.
- Vehicle parameters giving maximum accounted velocity, v_{max} and possible deceleration, $-a_{max}$ as well as the vehicle's possible stop time, t_{stop}, defined as the reaction time combined with the braking time.
- Uncertainty- and disturbance parameters given by the process noise covariance matrix, Q, and measurement noise covariance matrix, R, described in Chapter 3.

The parameter values and further descriptions is given in Appendix A.

6.2 System performance

The performance of the system was evaluated in a series of test scenarios as described in Section 2.5. The aim of the test scenarios were to validate the system's ability to detect and avoid, or at least mitigate, an imminent collision. The robustness of the system was also evaluated in the sense of how well it could handle uncertainties and disturbances.

The evaluated test scenarios have been divided into two categories, non-evasive and evasive manoeuvres. The non-evasive manoeuvres demonstrate the system's behaviour in scenarios where it is not supposed to intervene. The evasive manoeuvres on the contrary, show the performance of the system when it is supposed to intervene. In the following subsections, a number of test scenarios will be explained and evaluated to show the flexibility of the system. For the sake of brevity, some of the scenarios are presented in Appendix B.

The evaluation was made by looking at how the BN and IMM behaved for the EGO and POV respectively, how the probability of collision was changing over time and how well the system could cope with an oncoming collision. To evaluate the quality of the collision avoidance subsystem, each scenario was compared to an *optimal braking* as seen in Figure 6.1.

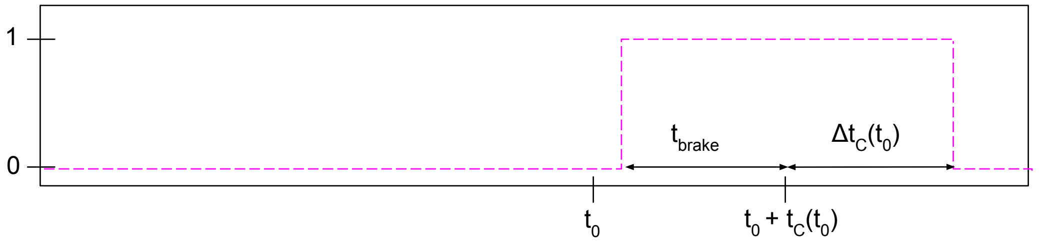


Figure 6.1: The optimal braking profiles were created by looking at the TTC, $t_C(t_0)$ in the time step before the braking occurs, t_0. The optimal braking starts t_{brake} seconds before and finishes Δt_C seconds after $t_C(t_0)$.

The optimal braking for each scenario was assumed to be most accurately calculated based on the TTC, $t_C(t_0)$, in the time step before the braking was initiated, t_0. The braking should optimally start t_{brake} seconds before and finish ΔTTC, $\Delta t_C(t_0)$, seconds after the $t_C(t_0)$.

It is also worth noting that the TTC and DTC in the *colliding trajectory*, of the TTC and DTC plots over time example seen in Figure 6.2, are the calculated ones when the threat assessment function is switched off. Therefore can the DTC and TTC increase or decrease from one time step to another even though a braking intervention is not performed. The reason for this change is due to that the estimations can change when the vehicles for example are turning where in most cases the vehicles also are decelerating. The result of this can be a curve that does not look smooth over time.

An overview of the full system, which will be evaluated, can be seen in Figure 6.3.

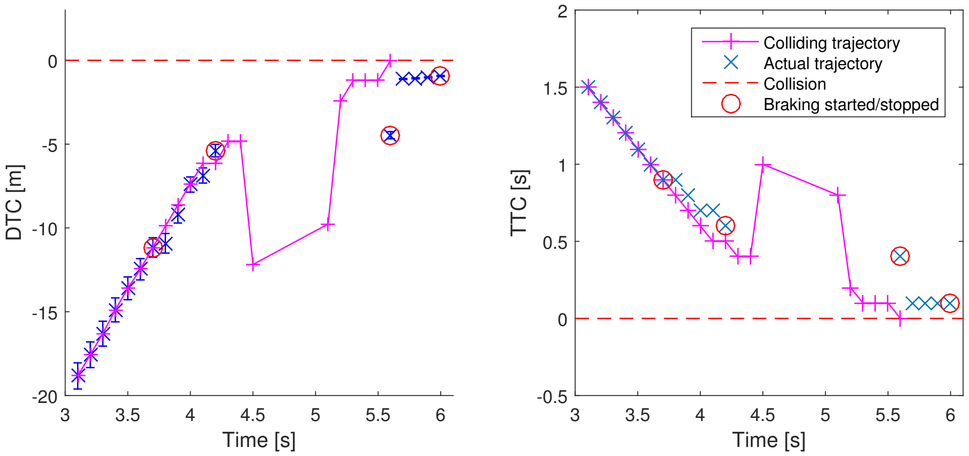


Figure 6.2: Example showing the behaviour of the *colliding trajectory* in the DTC and TTC plots over time

.

The flow through the system could be described as starting by acquiring the measurements for processing. These measurements will then flow through both the BN and IMM to give a probability of driver intent. Thereafter will the vehicle states be propagated using an UKF with the model describing the most probable driving path. The predicted states of each vehicle is then compared to evaluate an imminent threat and if so, return the time and distance to collision as well as the collision time interval. The final check evaluates if the current time is the time step just before PNR and if so, activates the emergency brake system.

Along the evaluation process, the nodes and default values stated below is the default BN setup, if nothing else is stated. To be able to have a BN being the most beneficial, the nodes used for the EGO were:

- turn signal (default: not set)
- road restrictions (default: possible to go straight, left and right)
- acceleration change (measured by the vehicle's internal sensors)
- lateral alignment (measured by a lane marker sensor)
- traffic rules (default: legally possible to go straight, left and right)
- angular rate (measured by the vehicles internal sensors)

The POV will however not be able to have the same nodes as the EGO, where the turn signal and lateral alignment would be indistinguishable, hence the nodes being used in the POV's BN were instead set as:

- road restrictions (default: possible to go straight, left and right)
- acceleration change (measured by the EGO's radar sensor)
- traffic rules (default: legally possible to go straight, left and right)
- angular rate (measured by the EGO's radar sensor)

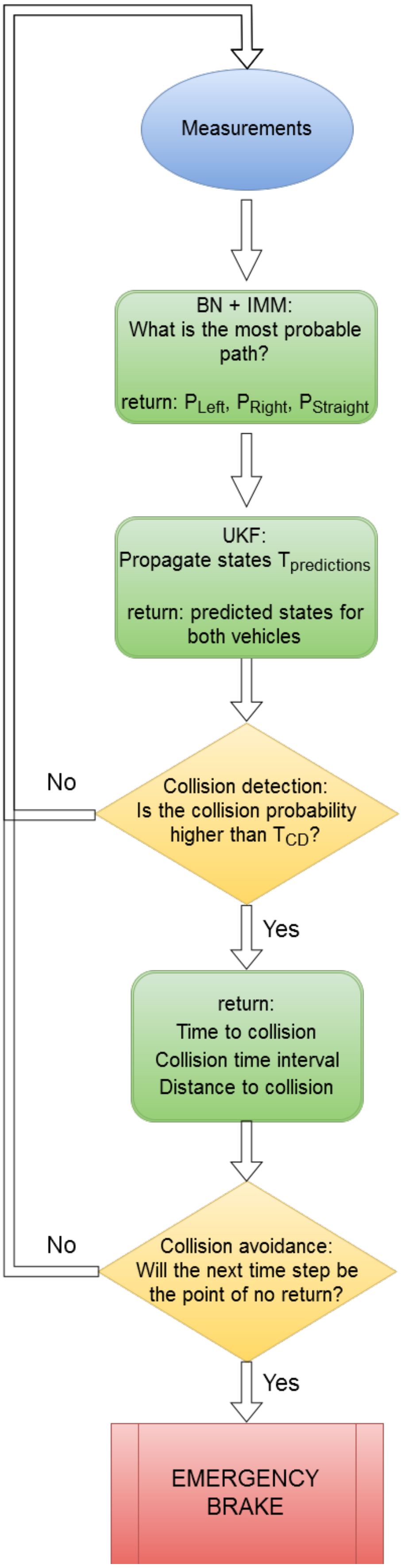

Measurements
BN + IMM:
What is the most probable path?
return: P_{Left}, P_{Right}, $P_{Straight}$
UKF:
Propagate states $T_{predictions}$
return: predicted states for both vehicles
Collision detection:
Is the collision probability higher than T_{CD}?
No
Yes
return:
Time to collision
Collision time interval
Distance to collision
Collision avoidance:
Will the next time step be the point of no return?
No
Yes
EMERGENCY BRAKE

Figure 6.3: Flowchart of the system. The flow through the system could be described as starting by acquiring the measurements for processing. These measurements will then flow through both the BN and IMM to give a probability of driver intent. Thereafter will the vehicle states be propagated using an UKF with the model describing the most probable driving path. The predicted states of each vehicle is then compared to evaluate an imminent threat, and if so then return the time and distance to collision as well as the collision time interval. The final check evaluates if the current time is the time step just before PNR, and if so activates the emergency brake system.

6.2.1 Non-evasive manoeuvres

For the non-evasive manoeuvres, the expected and desired outcome is to have a fully reliable system, not giving any unwanted interventions. This is evaluated using two scenarios denoted as *Abandoned turn* and *Avoidance manoeuvre*, where only the results for the Abandoned turn will be given in this section. The Avoidance manoeuvre can instead be seen in Section B.1 of Appendix, where it is illustrated that the scenario is carried without any false interventions, due to the probability of collision is kept below the collision detection threshold.

Abandoned turn:

Test scenario: Both the EGO and POV is travelling on a straight path. The POV then initiates a turn but gets to a full stop before crossing the EGO's lane, as can be seen in the illustration given by Figure 6.4.

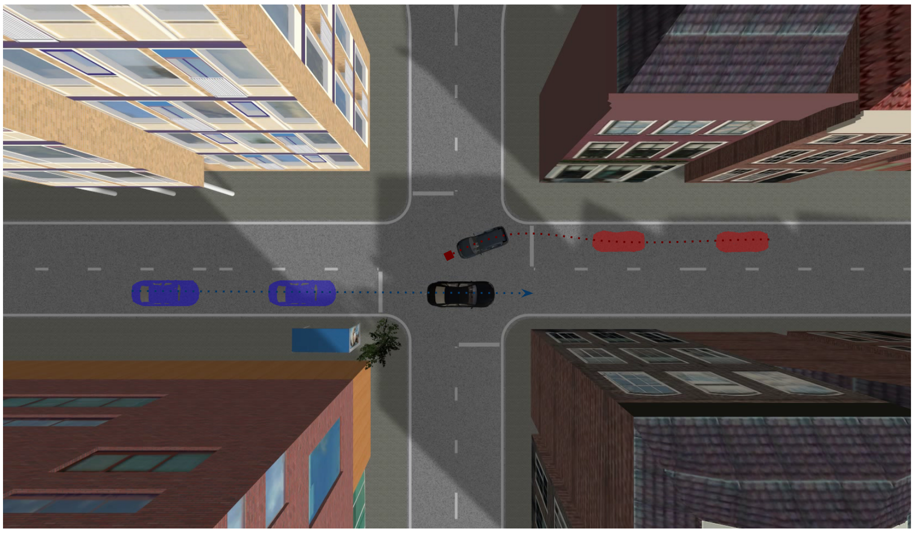
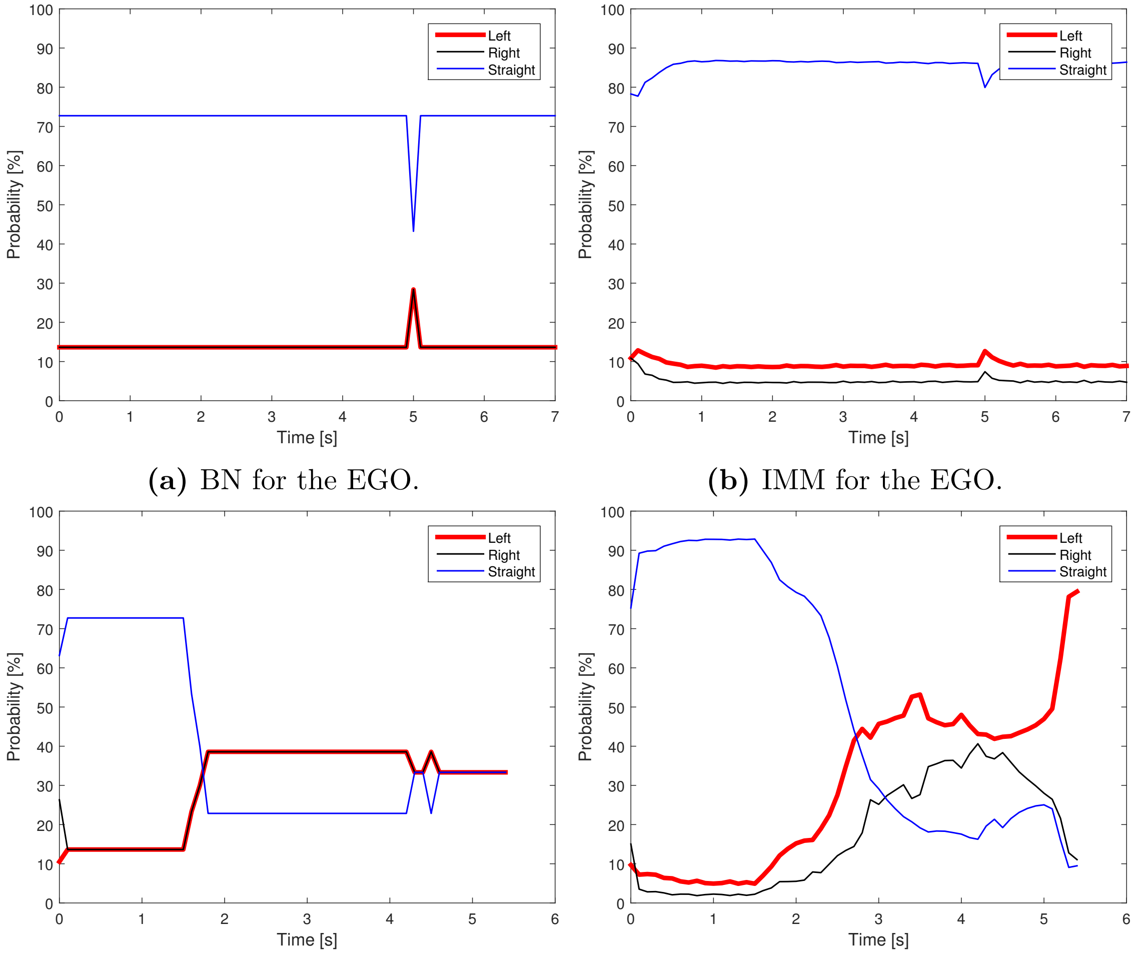


Figure 6.4: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.

In Figure 6.5 is the BN and IMM for both the EGO and POV illustrated, where the path with the highest probability for the EGO can be seen to be straight, Figure 6.5a and 6.5b, which corresponds to reality. Note that the highest probability in the BN for the EGO, Figure 6.5a, changes around 4.9 seconds but is corrected by the IMM filter, Figure 6.5b. This change is due to the information gained by the lane marker sensor, which in the centre of the intersection cannot detect any lane markers. Furthermore is the POV's BN seen in Figure 6.5c, where the BN cannot differentiate between a left or right turn. The IMM on the other hand have the ability to differentiate between them, as seen in Figure 6.5d. This demonstrates the BN's incapability of evaluating the driver intent on itself, hence shows the need of cooperation between the IMM and the BN.



(a) BN for the EGO.

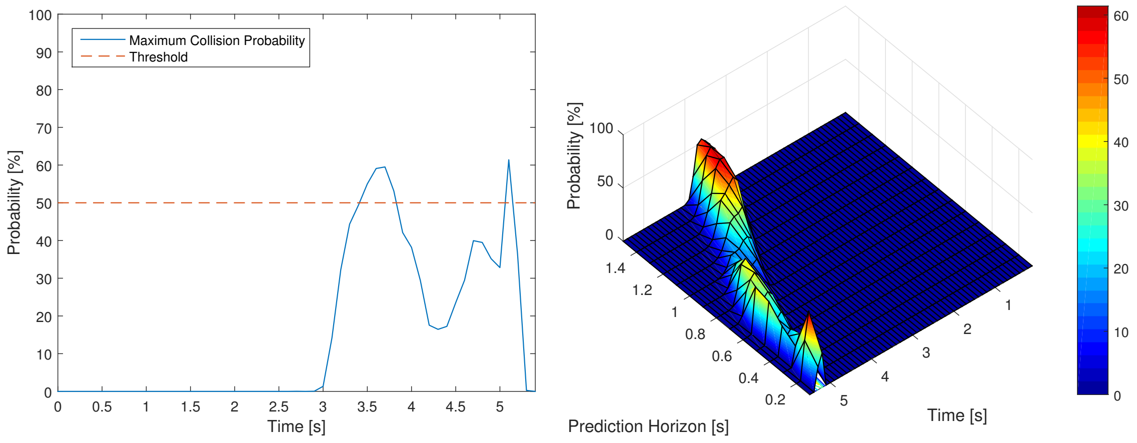
(b) IMM for the EGO.

(c) BN for the POV.

(d) IMM for the POV.

Figure 6.5: The BN probabilities as well as the IMM weights for each of the three models are here shown for both the EGO and the POV. The three different models are represented by red for left turn, black for right turn and blue for going straight.

Figure 6.6 illustrates the collision probability, where there is a risk of collision detected at around 3.5 seconds, but since the TTC has not reached PNR, a decision of braking is not made as seen in Figure 6.7. The other peak around 5 seconds is though both triggering a *Danger* signal and initiating the braking sequence. But in the end no braking is performed since the danger signal only is active for one time step.



(a) Maximum probability of collision of the prediction horizon, for each time step.

(b) Probability of collision in each prediction step for every time step.

Figure 6.6: Probability of collision according to the collision detection subsystem, where Figure 6.6a also illustrates the threshold that triggers the *Danger* signal.

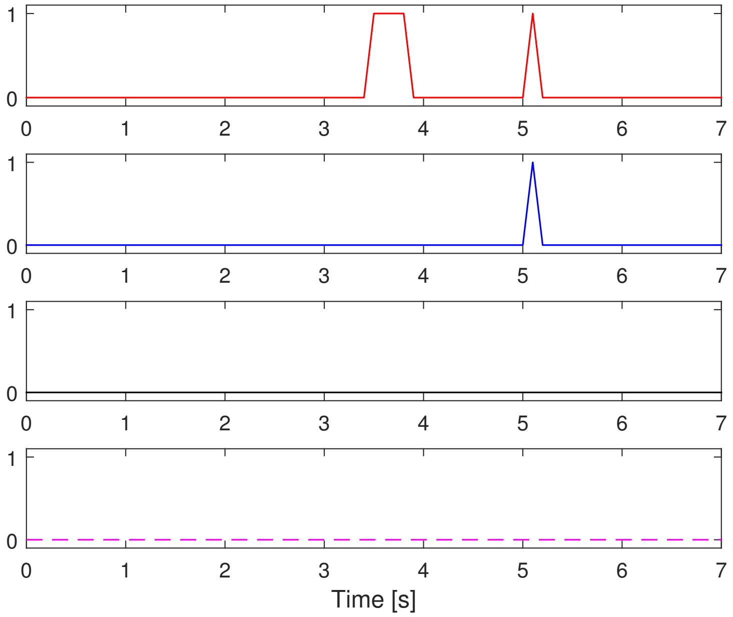


Figure 6.7: Illustration of the intervention timing from when a threat is detected to when a braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).

6.2.2 Evasive manoeuvres

The evasive manoeuvres have the demonstration effect of showing the systems ability to make a least restrictive, but maximally evasive intervention decision. The scenarios used to demonstrate this are a collision with a stationary obstacle, during two vehicles' crossing paths or when attempting a turning manoeuvre across or into the oncoming vehicles intended path. For the sake of brevity, only the most interesting results are presented in this chapter. More scenarios were nevertheless evaluated and is presented in Section B.2 of Appendix.

The scenarios found in appendix has a mixed outcome, a summary is given below but the reader is directed to Appendix B.2 for further details. The first scenario to be presented in the appendix is given by a stationary obstacle scenario. Here, a prevention of a collision was accurately performed ending up with the EGO getting to a full stop close to the POV. The second scenario instead evaluates an aggressive left turn by the POV across the EGO's path, where a safe braking procedure was performed ending up with a high DTC and TTC. In this scenario were the EGO able to get to a full stop at a safe distance to the POV. Three scenarios is thereafter conducted and presented where the EGO and the POV attempts a turn into the same lane. Two of the scenarios were close to a collision, but nevertheless avoided. The last scenario however were not fully avoided, but mitigated to a large degree. The last presented scenario is a road merging scenario, where two lanes merges into one, conducted to evaluate a scenario performed on other road sections than intersections. A collision was here avoided, thus instead initiating an intervention in a too early time step due to the scenario layout.

A Crossing paths:

Test scenario: Both the EGO and the POV is travelling on a straight path leading to a crossing intersection, an illustration can be seen in Figure 6.8.

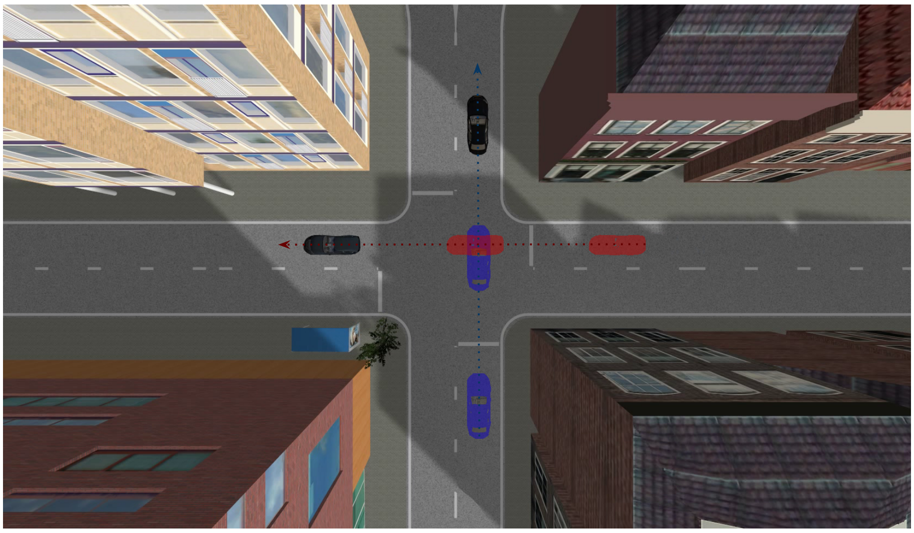
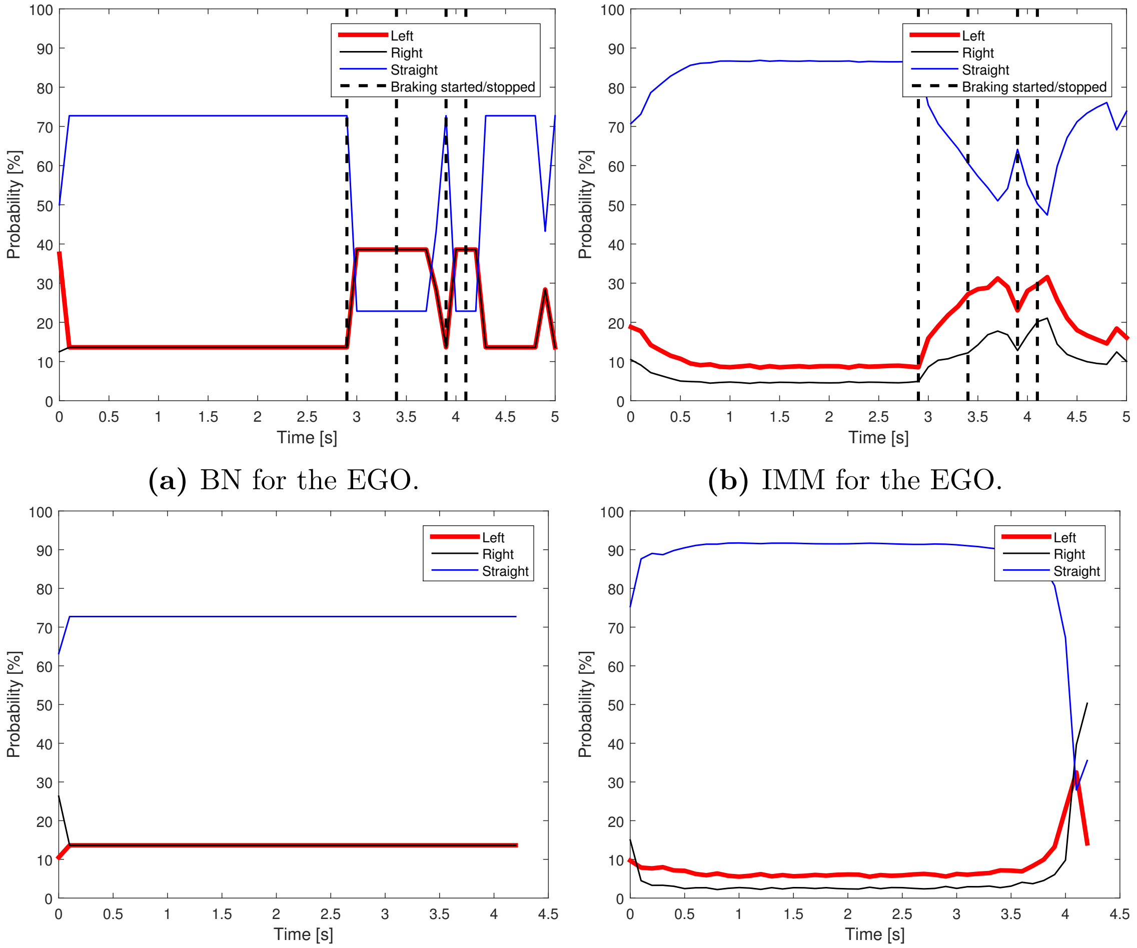


Figure 6.8: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.



(a) BN for the EGO.

(b) IMM for the EGO.

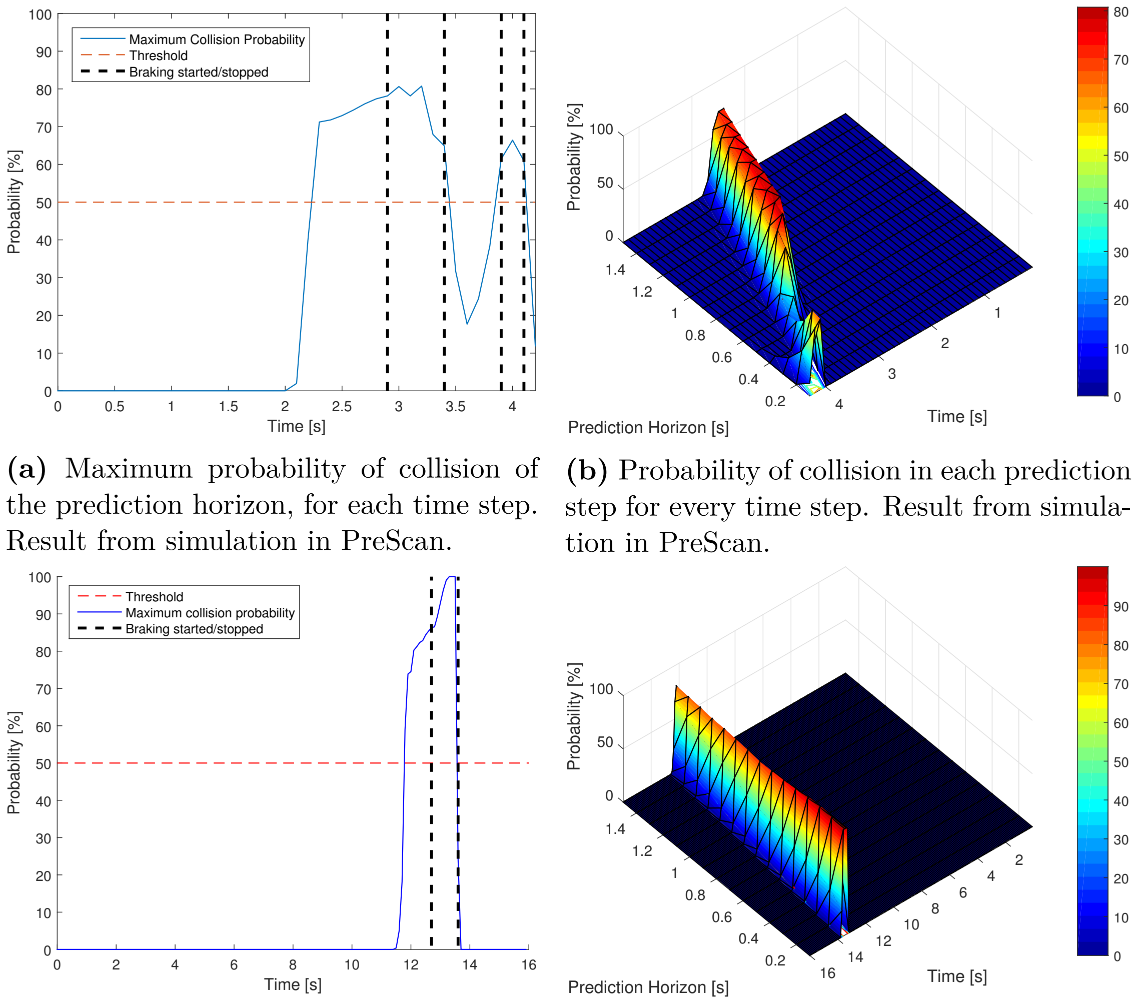
(c) BN for the POV.

(d) IMM for the POV.

Figure 6.9: The BN probabilities as well as the IMM weights for each of the three models in the simulation are here shown for both the EGO and the POV. The three different models are represented by red for left turn, black for right turn and blue for going straight. Additionally is the influence of the braking included for the EGO (black, dotted).

In Figure 6.9a and 6.9b are the BN and IMM probabilities illustrated for the EGO's three different models. The model having the highest probability in the BN changes from straight to turn model which can be explained by the braking graph seen in Figure 6.11a since deceleration is monitored by one of the information nodes in the BN. The IMM is though correcting this, hence the straight model is chosen as desired. The BN and IMM for the POV can be seen in Figure 6.9c and 6.9d, which show a high probability for the straight model as also desired.

The probability of collision in the simulation, seen in Figure 6.10a and 6.10b, has a probability above the threshold at around 2 seconds, but goes below when the EGO starts to brake. It finally increases again at around 4 seconds when a new threat has been detected. This behaviour follows from that the vehicle in the simulation case is braking and shortly thereafter accelerates, which most likely would not occur in reality.



(a) Maximum probability of collision of the prediction horizon, for each time step. Result from simulation in PreScan.

(b) Probability of collision in each prediction step for every time step. Result from simulation in PreScan.

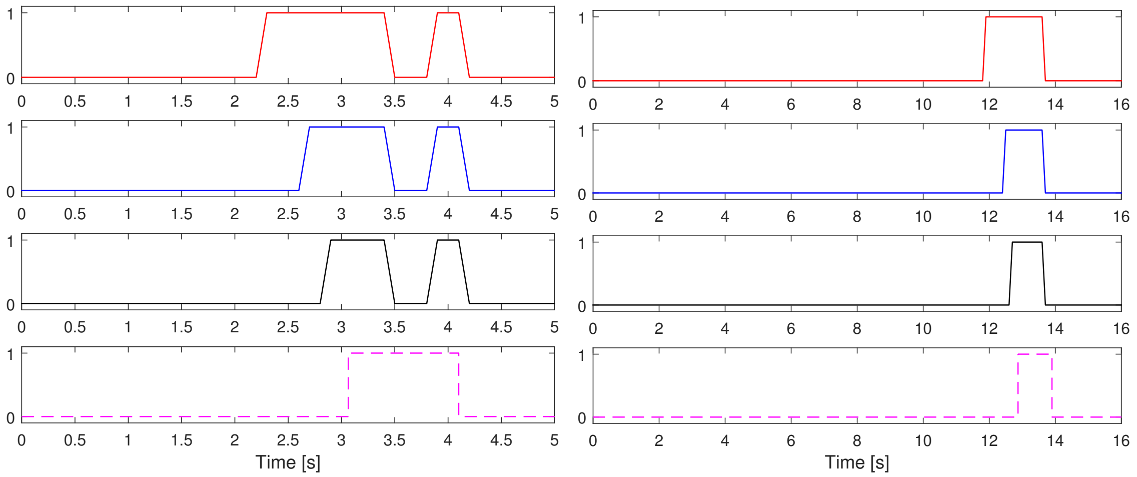
(c) Maximum probability of collision of the prediction horizon, for each time step. Result from in-vehicle test.

(d) Probability of collision in each prediction step for every time step. Result from in-vehicle test.

Figure 6.10: Probability of collision according to the collision detection subsystem, where Figure 6.10a and 6.10c also illustrates the threshold that triggers the *Danger* signal.

Figure 6.10c and 6.10d is instead illustrating the results from the real tests, which as seen is not behaving in the same fashion as in the simulation case. The in-vehicle tests was, as previously described in the *Stationary obstacle* scenario, however not performing any braking interventions which affects the length of the collision detection interval. The main purpose of the in-vehicle results is instead to demonstrate the system's threat assessment and supposed intervention ability, as also illustrated in Figure 6.11b.

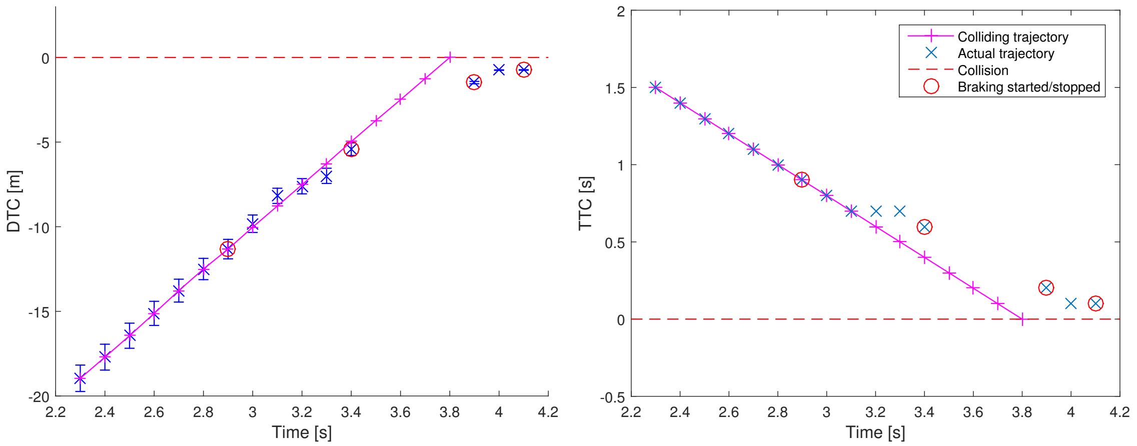
The probability of collision is triggering a *Danger* signal when it rises above the specified threshold. The collaboration between the *Collision detection* and *Collision avoidance* subsystems is thus illustrated in Figure 6.11a, where a threat is detected early but the brakes are however not applied until around 0.4 seconds later.



(a) Intervention timing, result from simulation in PreScan.

(b) Intervention timing, result from in-vehicle test.

Figure 6.11: Illustration of the intervention timing from when a threat is detected to when a braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).



(a) The evolvement of DTC over time.

(b) The evolvement of TTC over time.

Figure 6.12: Here are the DTC and TTC over time seen for the simulation. Blue represents the estimated DTC/TTC in each time step when the EGO executes an intervention. The red circles represent when the brakes are triggered and later released. Magenta represents the estimated DTC/TTC for a vehicle without a CA system. Figure 6.12a also illustrates the distance uncertainty with vertical bars. The idea is that the vehicle should never end up in a DTC or TTC equal to 0, since this is when a collision occurs.

The braking profile for the simulation, the DTC and the TTC as a factor of time, can be seen in Figure 6.12. This illustrates that the DTC and TTC ends up close to collision, but since the POV is travelling at a constant speed of 13 m/s and the EGO is standing still, in the end no collision occurs.

B Left turn by the POV across the EGO's path:

Test scenario: The EGO is travelling on a straight path whilst the POV makes a left turn across the EGO's lane, an illustration can be seen in Figure 6.13.

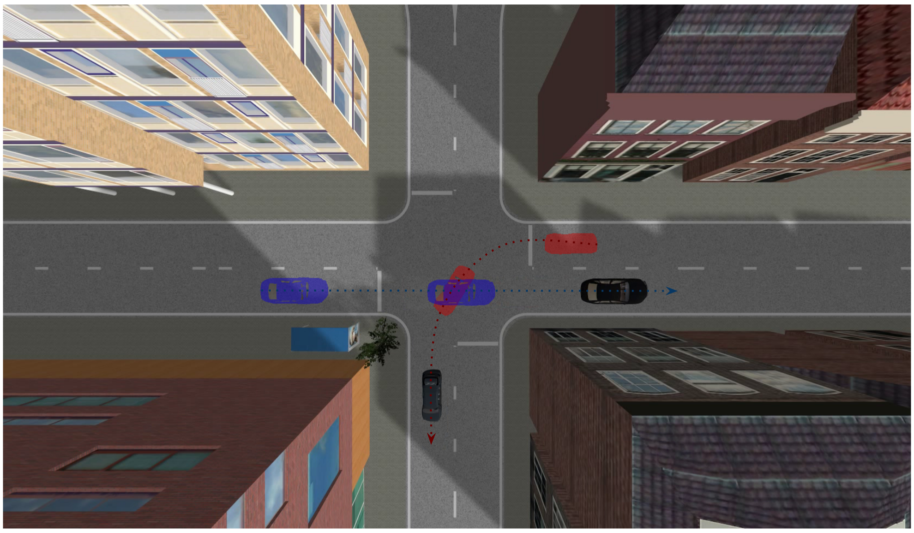
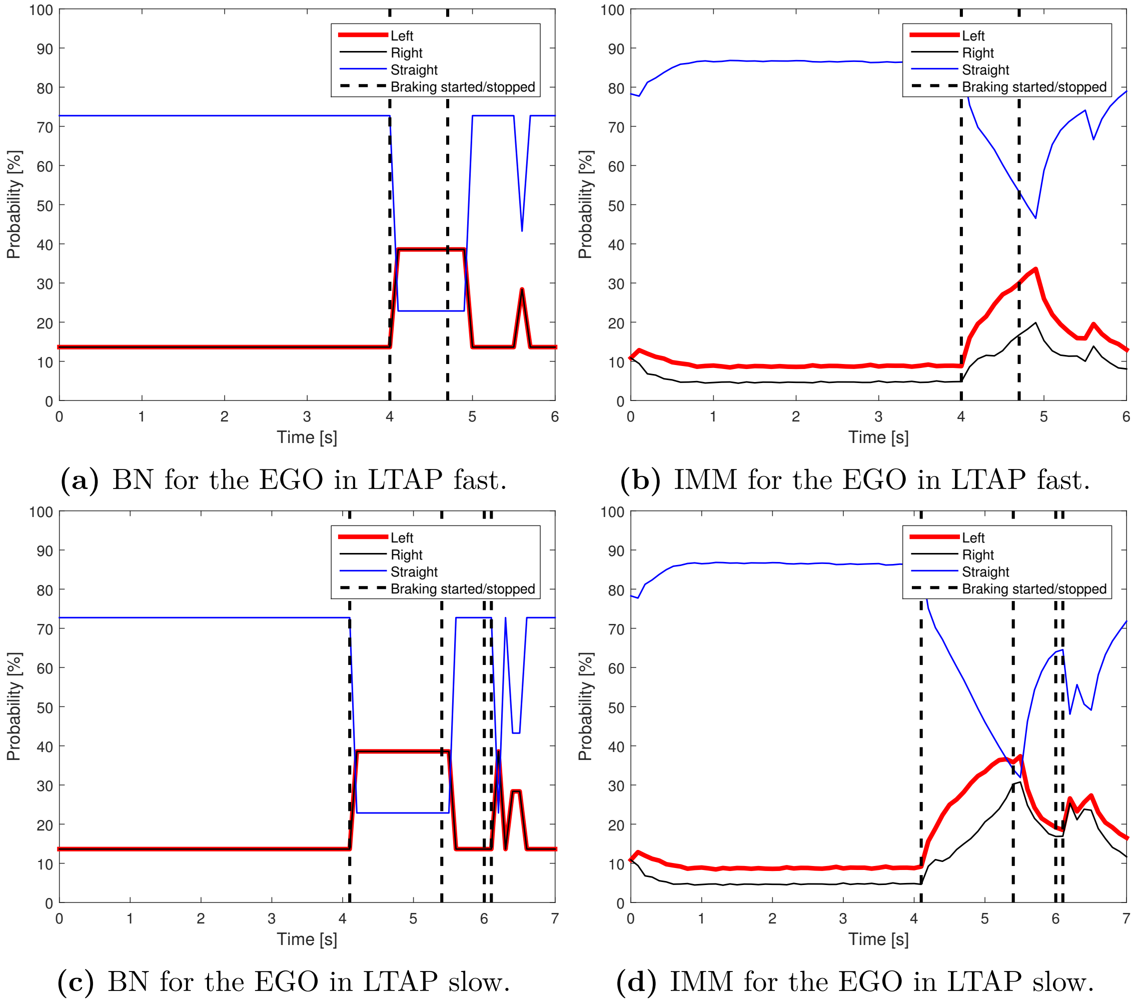


Figure 6.13: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.

Two different scenarios have been evaluated for the LTAP by the POV. In both scenarios is the POV slowing down before the turn, where in one of the scenarios the turn is taken in a high speed, hereafter referred to as *LTAP fast*, and in the other is it taken with a low speed, whereas the turn is performed during a longer time period, hereafter referred to as *LTAP slow*. Both of these scenarios were evaluated using simulations, whereas results with data recorded from real test can be seen in Appendix B.

Similar to before, the EGO is going straight and brakes when there is an imminent risk of collision, which results in the BN and IMM seen in Figure 6.14. In the BN and IMM for the *LTAP slow*, Figure 6.14c and 6.14d, the slower speed can be seen to result in a longer braking phase. The main difference is illustrated when comparing the IMM result for the slow and fast case, Figure 6.14b and 6.14d. The BN has during this brake phase altered its perception of likely driver intent, from straight to a turning motion. This affects the IMM for a longer time period in the *slow* speed case, hence it consequently counteracts the IMM's belief of straight driving direction in such a degree that the left turning motion has the highest probability at around 5.5 seconds.

Figure 6.15 illustrates the BN and IMM for the two different turns made by the POV. As can be seen, the duration of the turn is longer in the *LTAP slow*, Figure 6.15c and 6.15d, resulting in a more certain IMM when the turn actually occurs since the IMM-filter has more time to adjust. Note that the axis is longer for the *LTAP slow* than the *LTAP fast*.



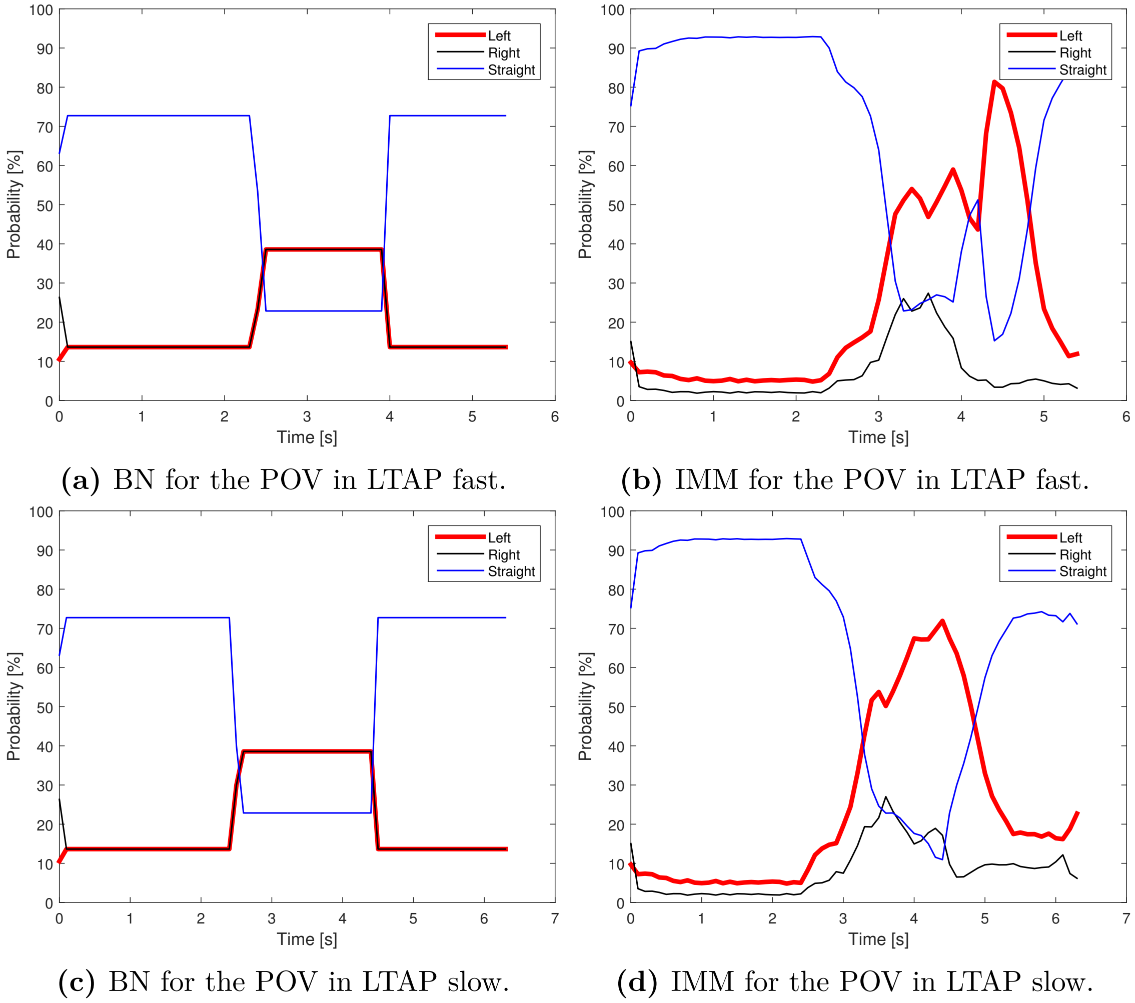
(a) BN for the EGO in LTAP fast. **(b)** IMM for the EGO in LTAP fast.

(c) BN for the EGO in LTAP slow. **(d)** IMM for the EGO in LTAP slow.

Figure 6.14: The BN probabilities as well as the IMM weights for each of the three models are here shown for the EGO when the POV makes a fast and a slow turn. The three different models are represented by red for left turn, black for right turn and blue for going straight. Additionally is the influence of the braking included for the EGO (black, dotted).

Figure 6.16 depicts the probability for the *LTAP fast* scenario, Figure 6.16a and 6.16b, and the *LTAP slow* scenario in Figure 6.16c and 6.16d. The probability of collision is generally higher and has a longer duration above the threshold for the slow turn case. The main difference is that the *LTAP slow* has a decreased probability of collision with a subsequent increase. The decrease of probability in Figure 6.17b can be explained by a braking intervention being performed in the time period between 4 and 5.5 seconds. The EGO is here accelerating which results in the probability to increase, hence the second peak would not be reasonable in reality.

The braking can be seen to have a longer duration and also a second peak for the *LTAP slow*, due to the already mentioned reasons, as seen in Figure 6.17b. The differences between the TTC and DTC for the *LTAP fast* and the *LTAP slow* can be seen in Figure 6.18. Since the POV passes by the EGO's intended path faster in the *LTAP fast*, the braking is released at a safe DTC and TTC.



(a) BN for the POV in LTAP fast.

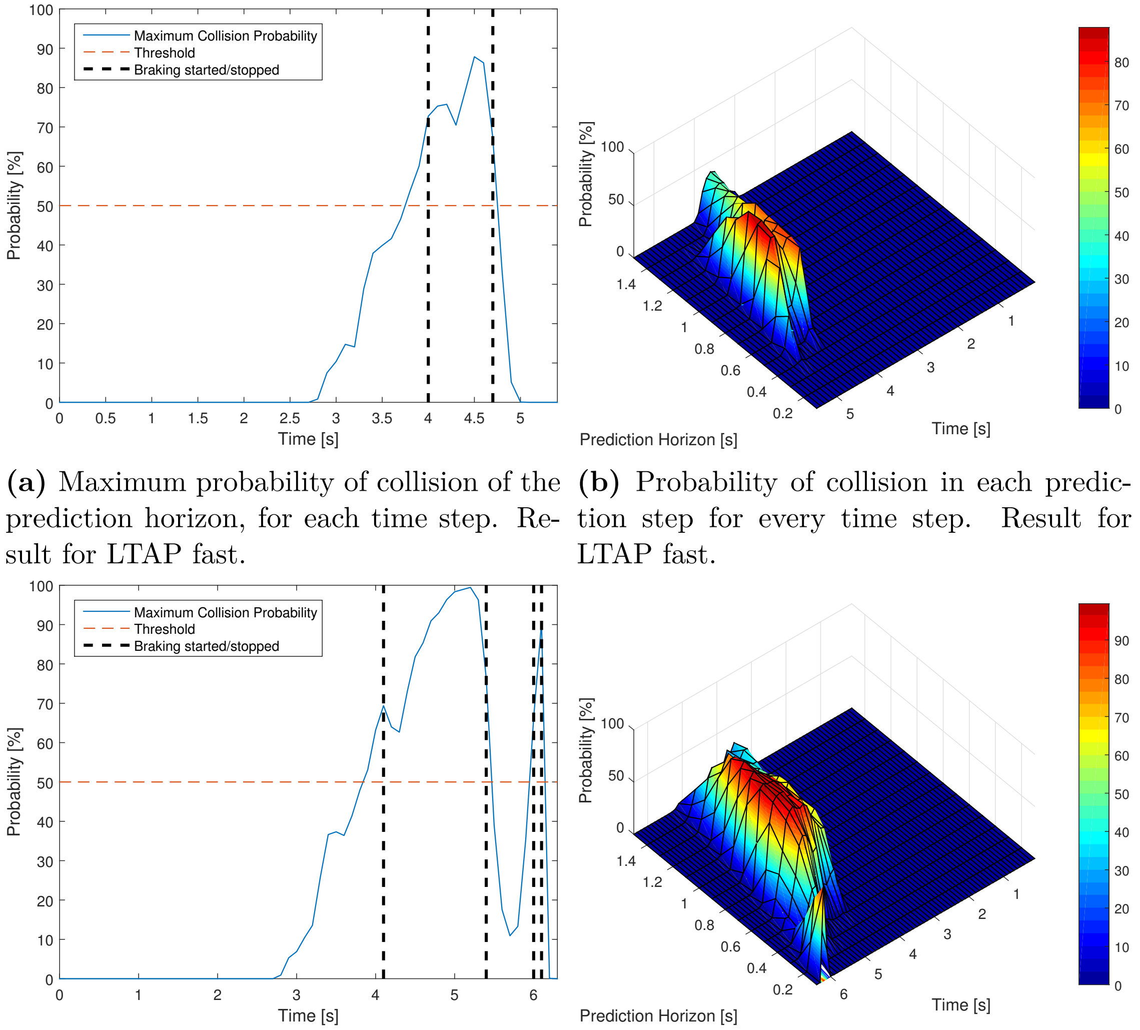
(b) IMM for the POV in LTAP fast.

(c) BN for the POV in LTAP slow.

(d) IMM for the POV in LTAP slow.

Figure 6.15: The BN probabilities as well as the IMM weights for each of the three models are here shown for the POV. The three different models are represented by red for left turn, black for right turn and blue for going straight.

For the *LTAP slow* scenario is the braking released first when being at a safe DTC and TTC, but since the EGO starts to accelerate, the risk of collision increases accordingly. Because of the EGO vehicle's low speed, it is though still possible to avoid a collision.



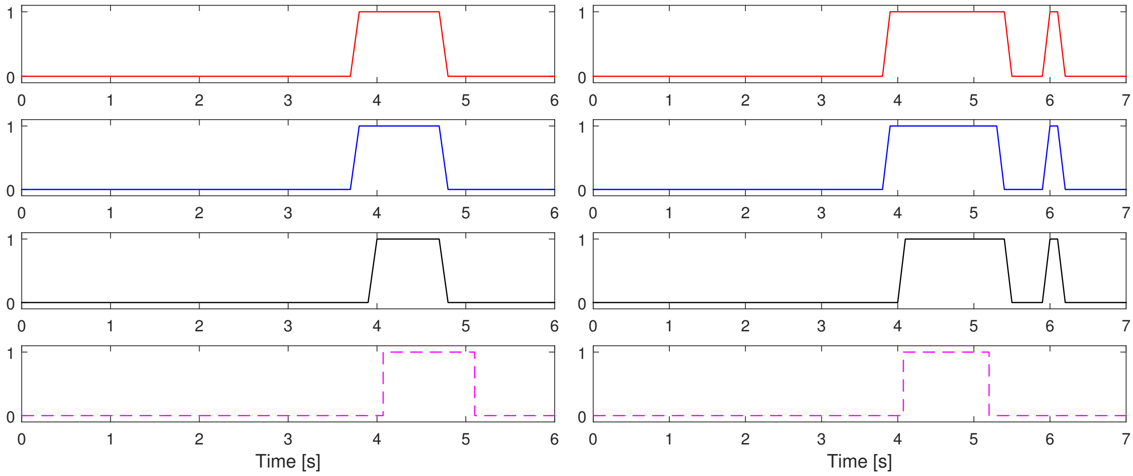
(a) Maximum probability of collision of the prediction horizon, for each time step. Result for LTAP fast.

(b) Probability of collision in each prediction step for every time step. Result for LTAP fast.

(c) Maximum probability of collision of the prediction horizon, for each time step. Result for LTAP slow.

(d) Probability of collision in each prediction step for every time step. Result for LTAP slow.

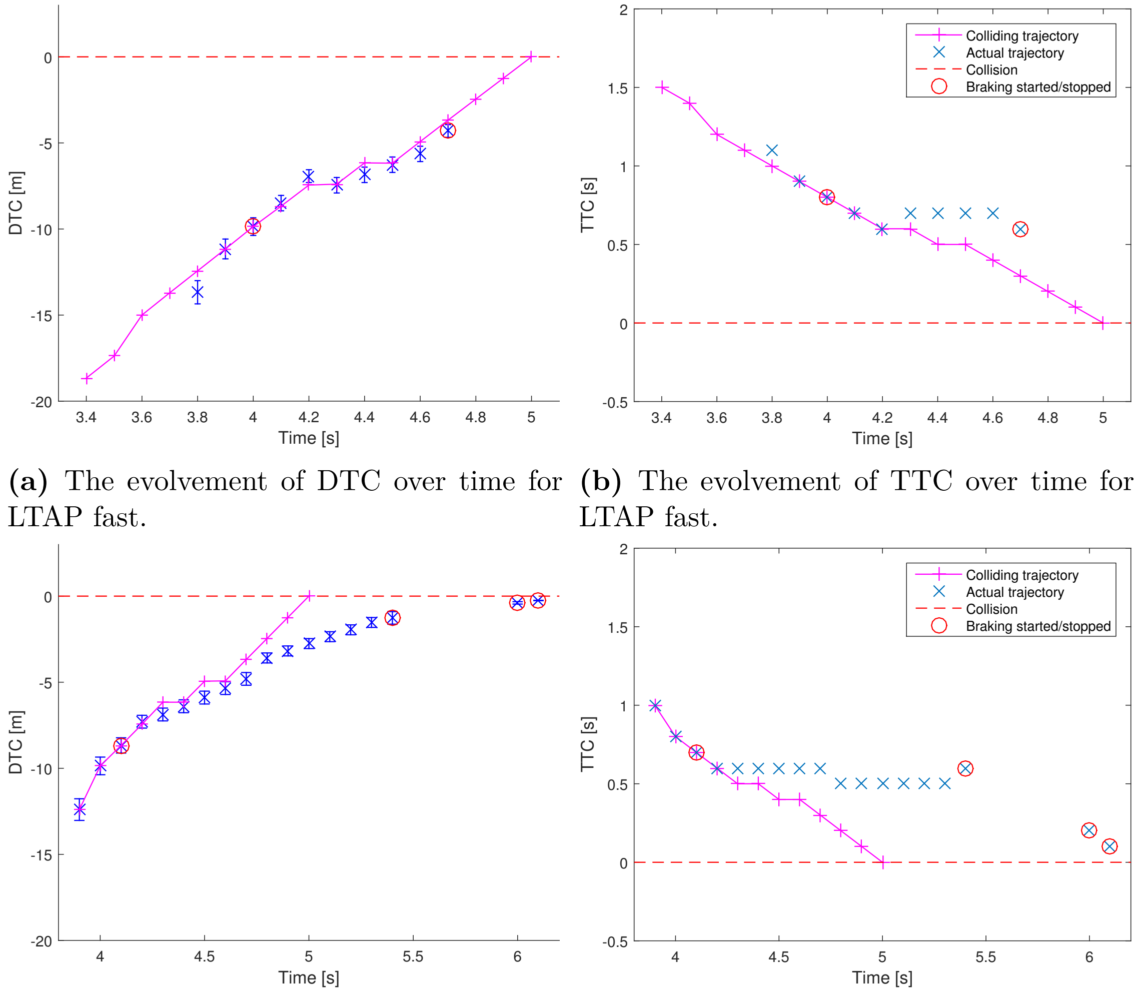
Figure 6.16: Probability of collision according to the collision detection subsystem, where Figure 6.16a and 6.16c also illustrates the threshold that triggers the *Danger* signal.



(a) Intervention timing, result for LTAP fast.

(b) Intervention timing, result for LTAP slow.

Figure 6.17: Illustration of the intervention timing from when a threat is detected to when a braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).



(a) The evolvement of DTC over time for LTAP fast.

(b) The evolvement of TTC over time for LTAP fast.

(c) The evolvement of DTC over time for LTAP slow.

(d) The evolvement of TTC over time for LTAP slow.

Figure 6.18: Here are the DTC and TTC over time seen for the two simulations. Blue represents the estimated DTC/TTC in each time step when the EGO executes an intervention. The red circles represent when the brakes are triggered and later released. Magenta represents the estimated DTC/TTC for a vehicle without a CA system. Figure 6.18c and 6.18a also illustrates the distance uncertainty with vertical bars. The idea is that the vehicle should never end up in a DTC or TTC equal to 0, since this is when a collision occurs.

C Left turn by the EGO across the POV's path:

Test scenario: The POV is travelling on a straight path whilst the EGO makes a left turn across the POV's path. The EGO's left turn signal is in this scenario activated close to the intersection which leads to a faster detection of the turn, an illustration can be seen in Figure 6.19.

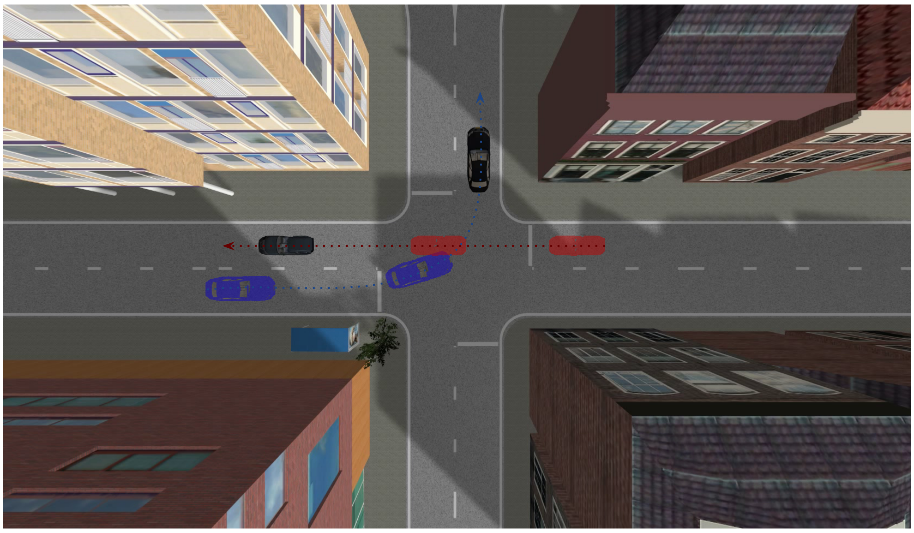
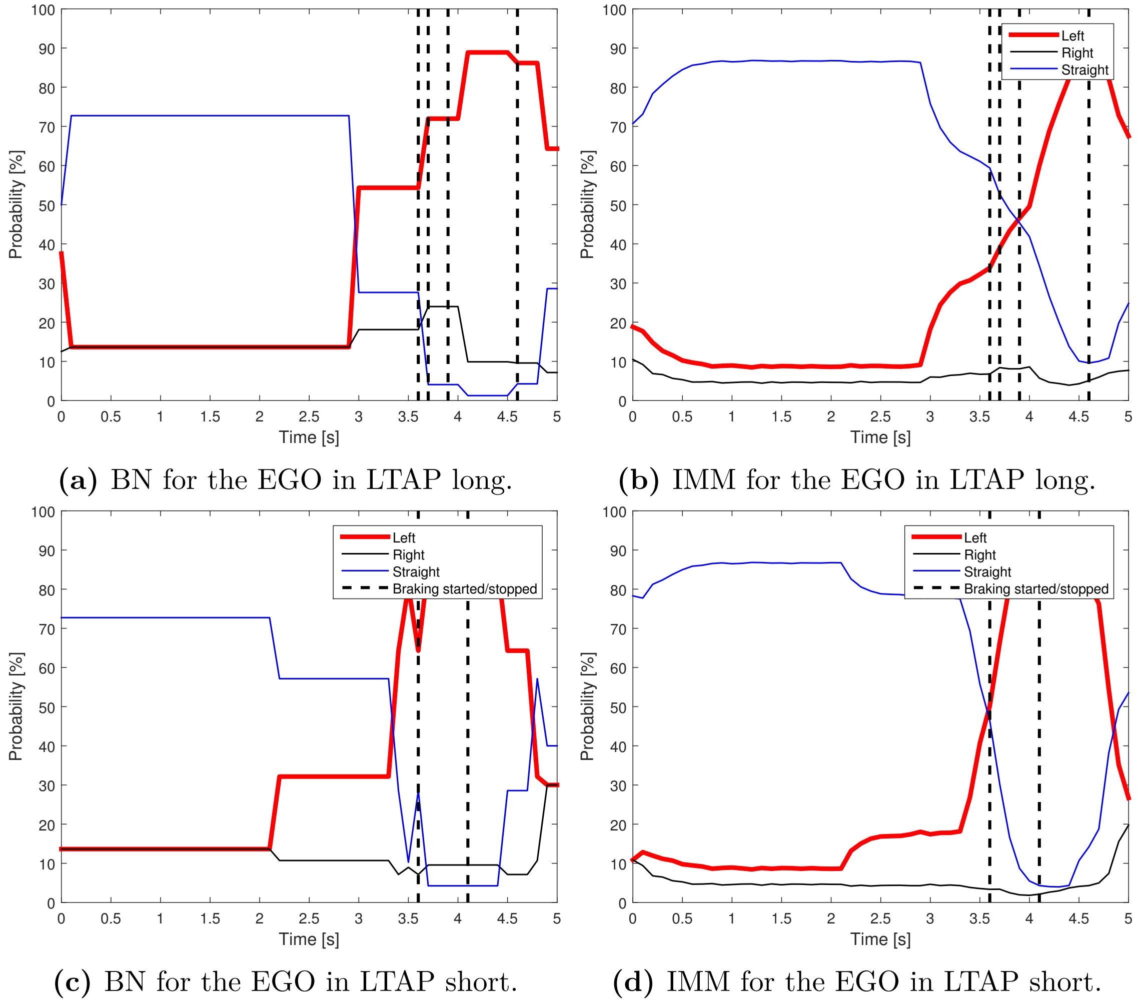


Figure 6.19: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.

Two different types of LTAP by the EGO across the POV's path are compared in this subsection, one where the EGO begins the turning manoeuvre early and therefore performs a longer turn, hereafter referred to as *LTAP long*, and the other where the turn is performed later, hereafter referred to as *LTAP short*. Both of these scenarios were evaluated using simulations, but additional results with data recorded from real test can be seen in Appendix B.

Figure 6.20 illustrates the EGO's BN and IMM probabilities for the two different turns, both of them behaving similar, as desired. As demonstrated by the figures, the EGO's initiated left turn is clearly detected. The BN and IMM for the POV is similar to what was seen in Figure 6.9c and 6.9d, the crossing path scenario, since the POV is travelling on a straight path.

In Figure 6.21 can the probability be seen for the different turns. Figure 6.21a and 6.21b illustrates the *LTAP long* scenario, where the probability of collision changes drastically at around 4 seconds due to the change of model in this interval as seen in the IMM for the EGO. This demonstrates the *Collision detection* subsystems vulnerability towards the choice of threshold, where an evaluation to find the optimal threshold can be seen in Section 6.3.2. Figure 6.21c and 6.21d illustrates the probability of collision for the *LTAP short* scenario, where the probability of collision is growing drastically and becomes very high hence reveals information of a collision to occur.



(a) BN for the EGO in LTAP long.

(b) IMM for the EGO in LTAP long.

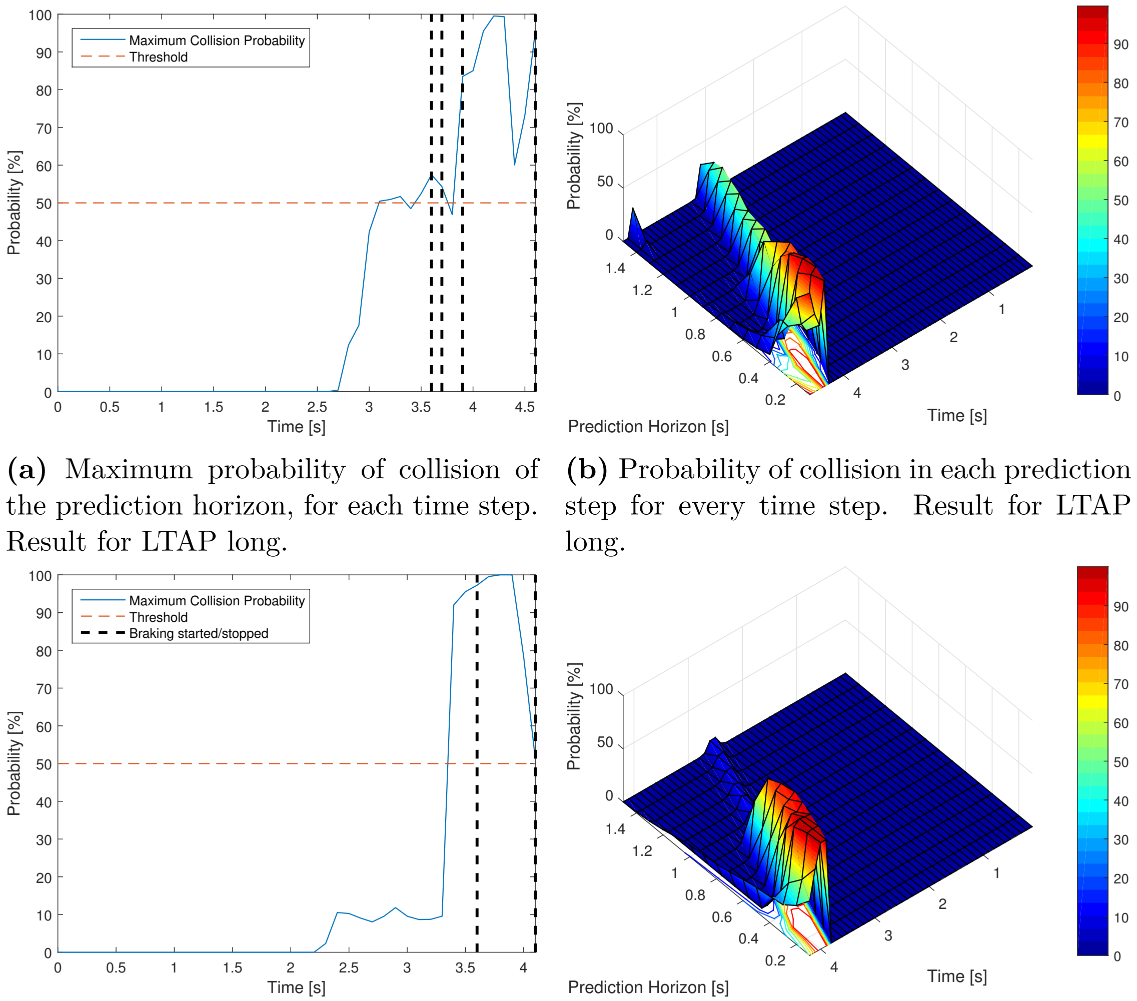
(c) BN for the EGO in LTAP short.

(d) IMM for the EGO in LTAP short.

Figure 6.20: The BN probabilities as well as the IMM weights for each of the three models are here shown for the EGO. The three different models are represented by red for left turn, black for right turn and blue for going straight. Additionally is the influence of the braking included for the EGO (black, dotted).

The intervention timing is seen in Figure 6.22, where the timing for the *LTAP long* is shown Figure 6.22a, which is fluctuating because of the change of the *Danger* signal. The reason for the *Danger* signal to alter between high and low, is due to the change of decided motion model to use for the predictions during the turn. At the beginning of the turn, a straight motion model is used hence detecting an imminent collision. Thus with the transition to the left turning motion model, a collision is not deemed likely at first but is at 3.8 seconds set as being an imminent collision.

A braking intervention is triggered first 0.5 seconds after the first received *Danger* signal, however not kept sustained until the time of 3.8 seconds as illustrated in Figure 6.22b. The collision is thus mitigated due to the initial braking at 3.5 seconds, but not entirely avoided. As compared to the optimal braking, the braking phase should ideally be set as being sustained earlier for the ability to completely avoid the collision.



(a) Maximum probability of collision of the prediction horizon, for each time step. Result for LTAP long.

(b) Probability of collision in each prediction step for every time step. Result for LTAP long.

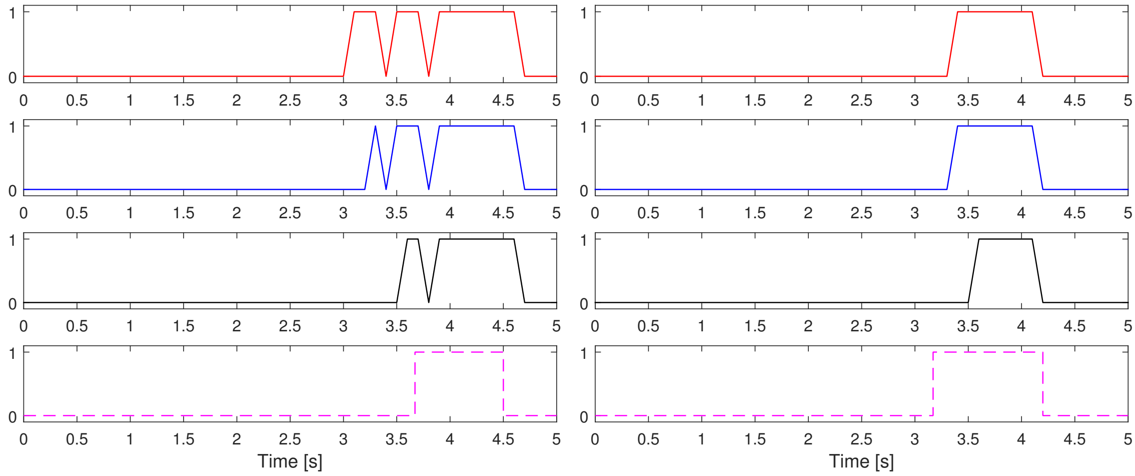
(c) Maximum probability of collision of the prediction horizon, for each time step. Result for LTAP short.

(d) Probability of collision in each prediction step for every time step. Result for LTAP short.

Figure 6.21: Probability of collision according to the collision detection subsystem, where Figure 6.21a and 6.21a also illustrates the threshold that triggers the *Danger* signal.

The DTC and TTC as a function of time can be seen in Figure 6.23. Figure 6.23a and 6.23b, for the *LTAP long*, shows that even though the EGO is braking, it is really close to a collision.

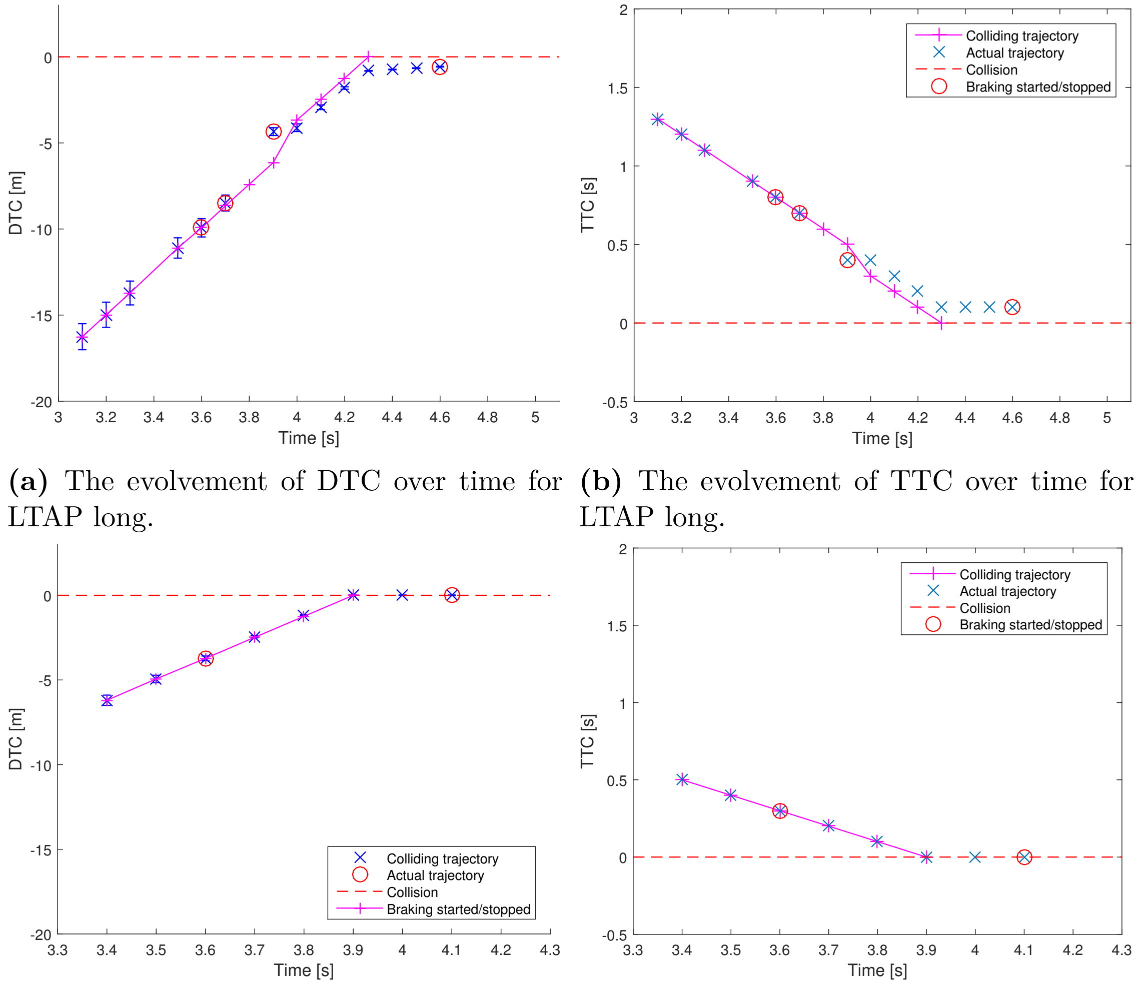
Figure 6.23c and 6.23d, for the *LTAP short*, shows that the collision time interval is really narrow and consequently the collision is seen really late. When the braking starts, the TTC is 0.3 seconds which means that the oncoming collision is unavoidable.



(a) Intervention timing, result for LTAP long.

(b) Intervention timing, result for LTAP short.

Figure 6.22: Illustration of the intervention timing from when a threat is detected to when a braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).



(a) The evolvement of DTC over time for LTAP long.

(b) The evolvement of TTC over time for LTAP long.

(c) The evolvement of DTC over time for LTAP short.

(d) The evolvement of TTC over time for LTAP short.

Figure 6.23: Here are the DTC and TTC over time seen for the two simulations. Blue represents the estimated DTC/TTC in each time step when the EGO executes an intervention. The red circles represent when the brakes are triggered and later released. Magenta represents the estimated DTC/TTC for a vehicle without a CA system. Figure 6.23c and 6.23a also illustrate the distance uncertainty with vertical bars. The idea is that the vehicle should never end up in a DTC or TTC equal to 0, since this is when a collision occurs.

6.3 Robustness analysis

The overall system performance has been presented through evaluation of different scenarios as seen in Section 6.2.1 and 6.2.2. These scenarios were evaluated with, for this system, parameters giving the best performance as presented in Section 6.1 and 6.2. Although the system works optimally for the given vehicle setup with the stated parameter values, other sensors with better or worse capacity could change the presented system's ability to perform.

To be able to determine the systems performance and capacity, a robustness analysis is needed to expose the critical limits. Section 6.3.1 presents an analysis made in regard to critical velocity levels and in Section 6.3.2 is an analysis made for critical noise levels. In Section 6.3.3, 6.3.4 and 6.3.5 is instead the performance and critical parameters of each individual subsystem *Long-term path prediction*, *Collision detection* or *Collision avoidance* presented.

6.3.1 Critical velocity limits

The purpose of stressing the system with different velocities, is to determine which critical velocity limits that gives unwanted system performance. Such an unwanted system performance could be an undesired intervention as seen to the driver's expectation, hence could lead to a disastrous outcome. The awareness of the critical velocity limits could thus be used to narrow down the usage of the presented system to some particular scenarios or sections of the road network. Moreover could it point out which future work needing more attendance.

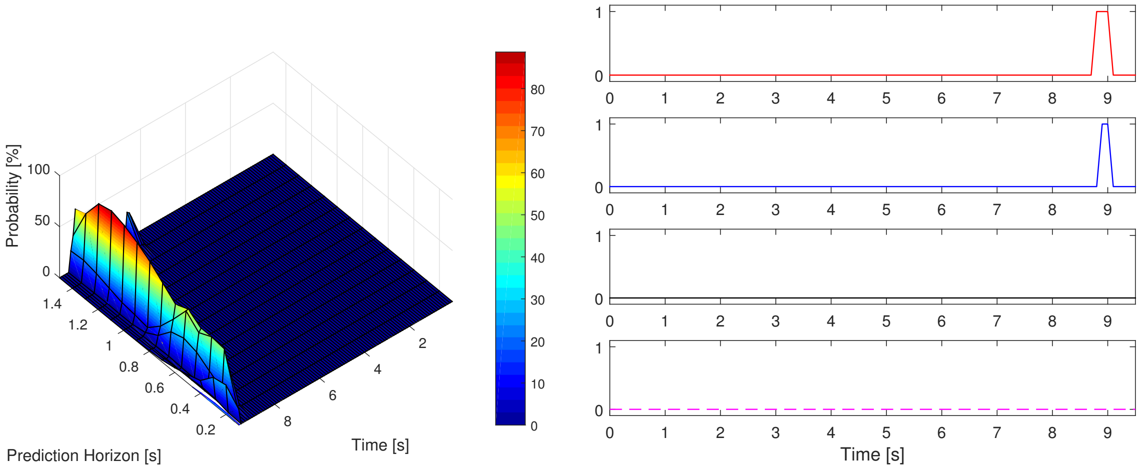
To determine the critical velocity limits, the four scenarios *Abandoned turn*, *Crossing intersection*, *Left turn across path made by the EGO* and *Left turn across path made by the POV* were used for evaluation. The different scenarios demonstrate one case where no braking should be applied, one where both vehicles are heading straight at each other and two cases where either the EGO or the POV makes a turn across the others intended path.

The velocities determined to be the system's critical limits are categorised as the *lower velocity limit* at 6 [m/s] (21.6 [km/h]), *regular velocity limit in urban intersections* at 13 [m/s] (46.8 [km/h]) and the *upper velocity limit* at 16 [m/s] (57.6 [km/h]) where the result using these critical limits is presented in each section representing a specific scenario.

A Abandoned turn

Performing the abandoned turn scenario should result in a non-evasive manoeuvre, which is accomplished for all velocity levels as seen in Figure 6.24b, 6.25b and 6.26b.

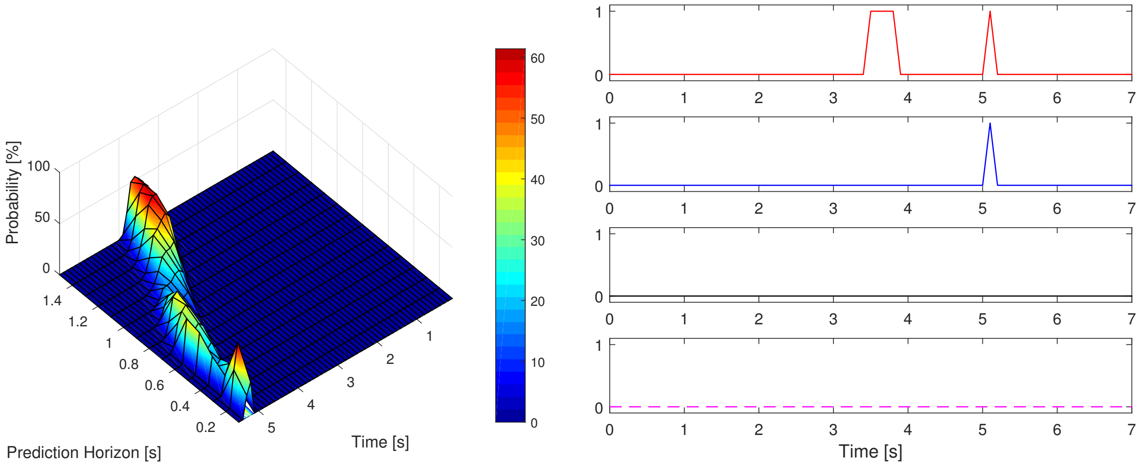
Figure 6.24a reveals that the collision detection rate at low speeds is high but concentrated to approximately one time step. On the contrary, the detection rate at high speeds, shown in Figure 6.26a, becomes low as well as scattered across more time steps. It can thus be explained by the fact that at low speeds, the prediction horizon will not extend as far as for higher speeds. An intersection of two oncoming vehicles with either low or high speed will therefore be detected either later or earlier accordingly. Figure 6.25 illustrates the perception for the regular speed setup where an imminent collision is detected, but without an intervention being activated as illustrated in Figure 6.25b. This figure is thus also showing that a braking intervention is recommended, blue curve, but not applied due to a too short collision detection time span.



(a) 3D-probability

(b) Threat/brake assessment

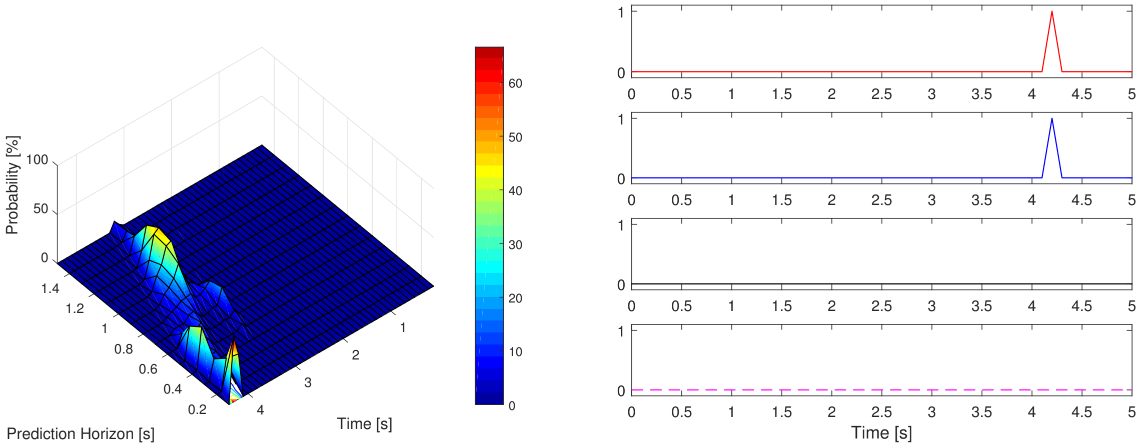
Figure 6.24: Abandoned turn scenario with low speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).



(a) 3D-probability

(b) Threat/brake assessment

Figure 6.25: Abandoned turn scenario with regular speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).

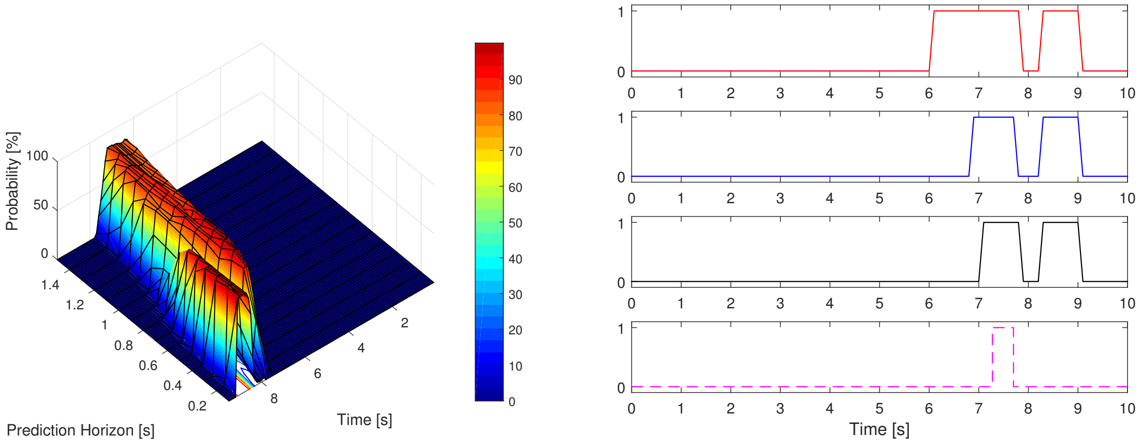


(a) 3D-probability **(b)** Threat/brake assessment

Figure 6.26: Abandoned turn scenario with high speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).

B Crossing intersection

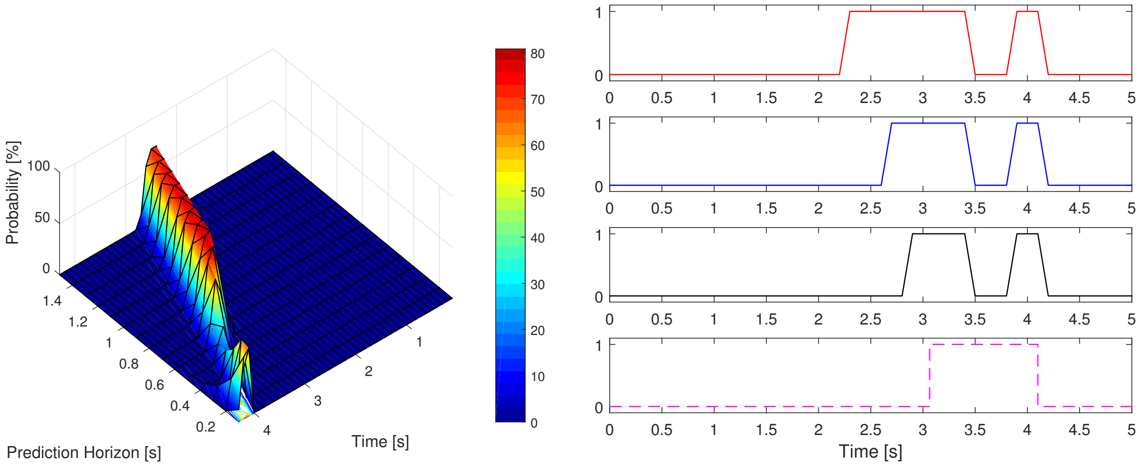
The crossing intersection scenario is classified as a forgiving scenario in the sense that the long-term path predictions could be done sufficiently accurate, even for vehicles with high speed, since both vehicles travel in a straight direction. This is illustrated for low speed in Figure 6.27, same speed as the regular setup in 6.28 and at high speed seen in Figure 6.29.



(a) 3D-probability **(b)** Threat/brake assessment

Figure 6.27: Crossing intersection scenario with low speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).

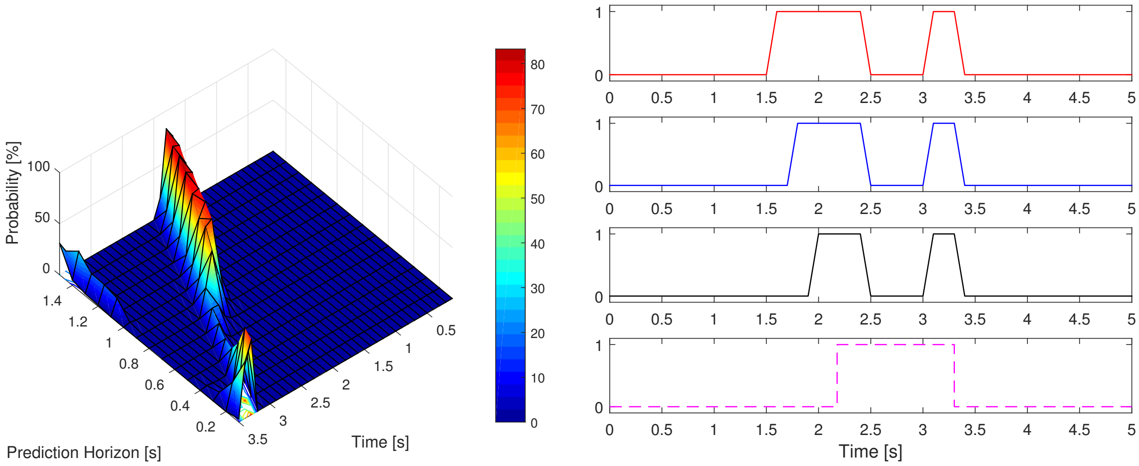
The main difference is at what detection rate the EGO can identify the imminent risk of collision. The low speed case is illustrated in Figure 6.27, where 6.27b demonstrates that a threat is detected later in time and 6.27a shows that the collision probability will be concentrated to just a few time steps. For the regular and high speed case, the imminent danger could be detected earlier as illustrated in both Figure 6.28 and 6.29.



(a) 3D-probability

(b) Threat/brake assessment

Figure 6.28: Crossing intersection scenario with regular speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).



(a) 3D-probability

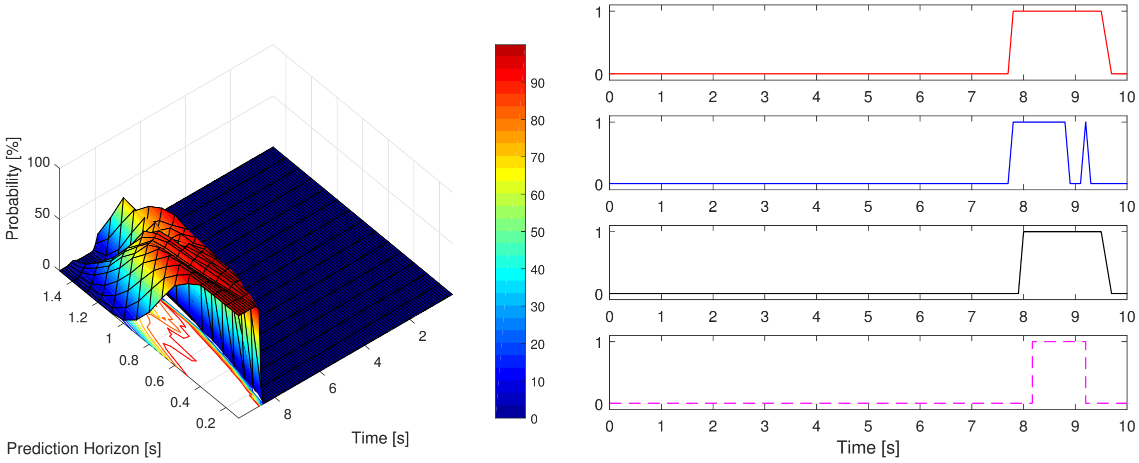
(b) Threat/brake assessment

Figure 6.29: Crossing intersection scenario with high speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).

C Left turn across path made by the EGO

A left turn by the EGO across the POV's path is illustrated for the three speed levels with low speed in Figure 6.30, same speed as the regular setup in 6.31 and high speed seen in 6.32.

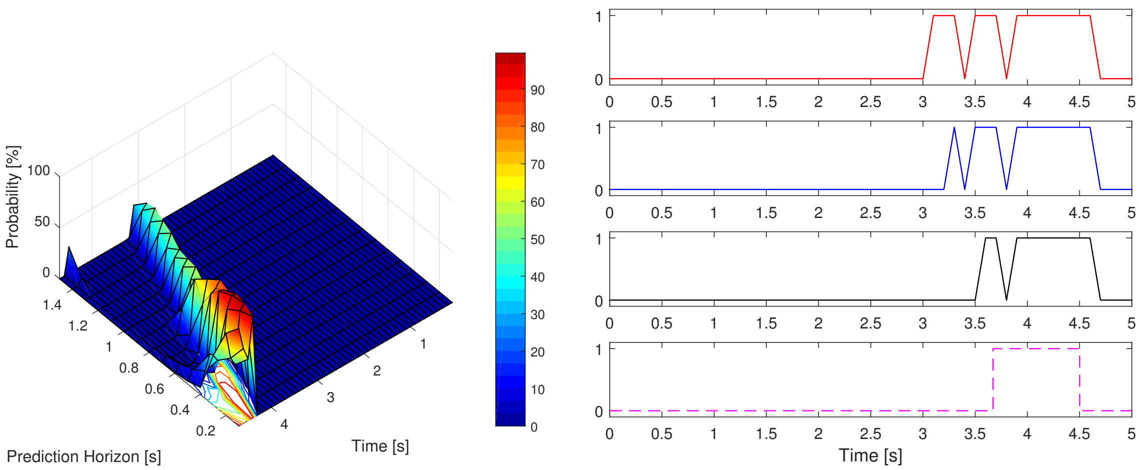
The low and regular speed cases are shown to be manageable by the system, whereas at the high speed level, a full collision avoidance could not be accomplished due to too late detection time as shown in Figure 6.32b. Figure 6.32b also demonstrates a comparison between the actual braking and the referenced need of braking, magenta dashed line, where the braking intervention at latest should be made at the reference start. Even though the collision is not avoided, it is nonetheless mitigated.



(a) 3D-probability

(b) Threat/brake assessment

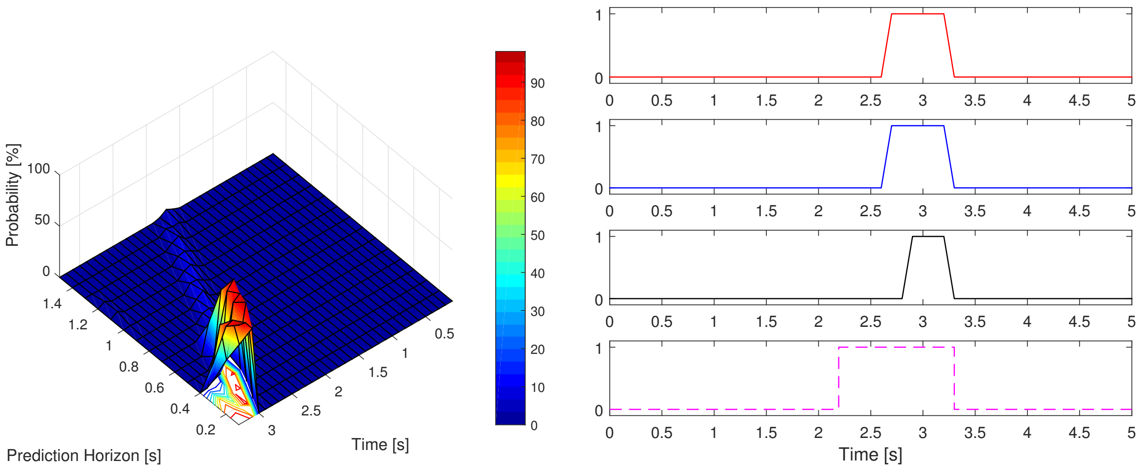
Figure 6.30: Left turn across path by the EGO scenario with low speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).



(a) 3D-probability

(b) Threat/brake assessment

Figure 6.31: Left turn across path by the EGO scenario with regular speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).



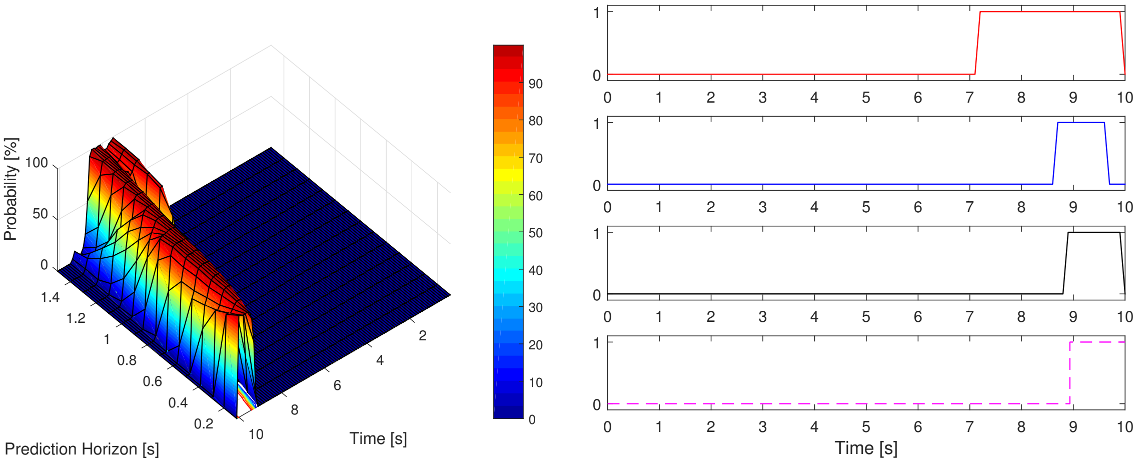
(a) 3D-probability

(b) Threat/brake assessment

Figure 6.32: Left turn across path by the EGO scenario with high speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).

D Left turn across path made by the POV

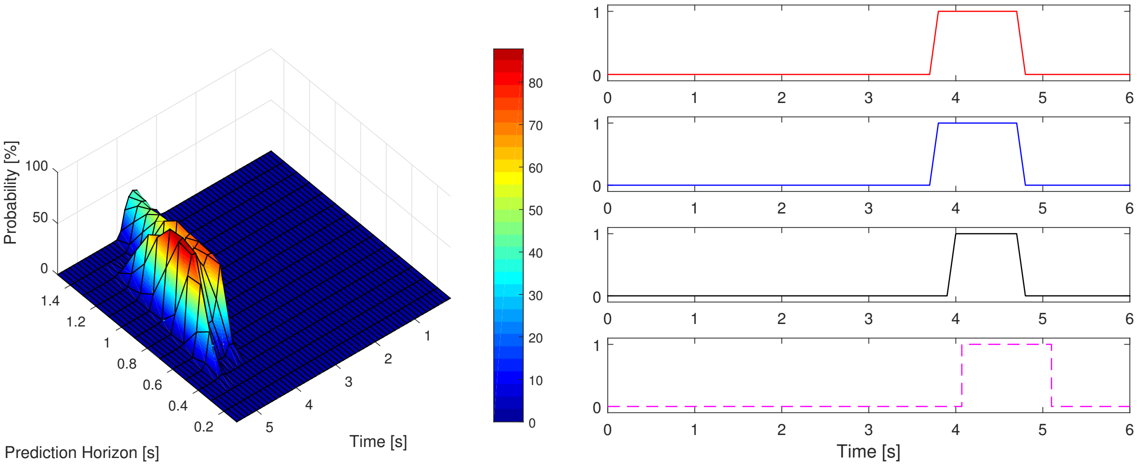
A left turn by the POV across the EGO's path, is in the same fashion as previously demonstrated results illustrated for the three speed levels with low speed in Figure 6.33, regular speed in 6.34 and high speed seen in 6.35. The low and regular speed cases are again shown to be manageable by the system. The low speed case is also demonstrating the system's ability to disregard a threat until the time step just before the PNR. The turn at high speed is thus detected too late hence the collision could not be avoided, but is nonetheless mitigated.



(a) 3D-probability

(b) Threat/brake assessment

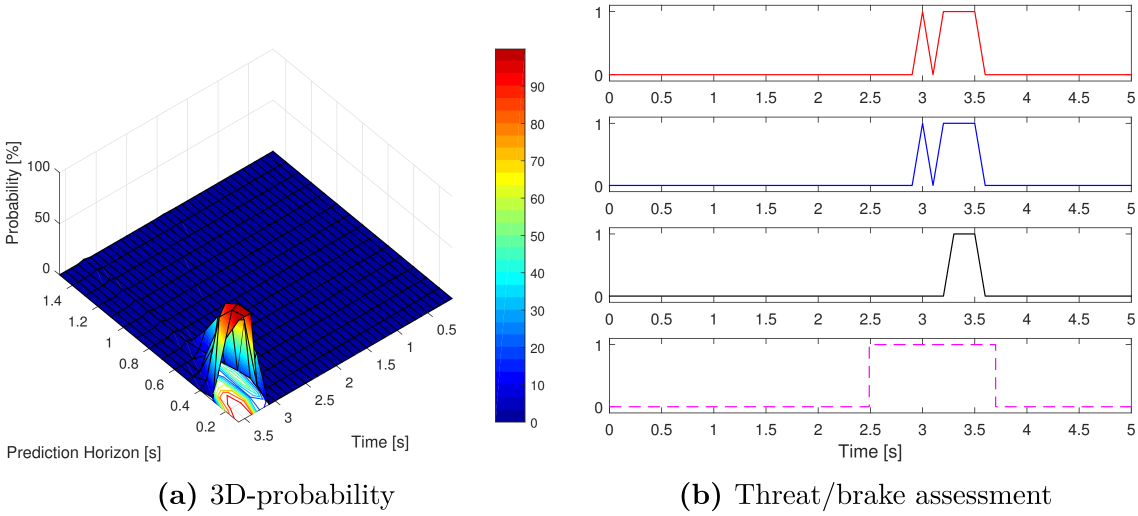
Figure 6.33: Left turn across path by the POV scenario with low speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).



(a) 3D-probability

(b) Threat/brake assessment

Figure 6.34: Left turn across path by the POV scenario with regular speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).



(a) 3D-probability **(b)** Threat/brake assessment

Figure 6.35: Left turn across path by the POV scenario with high speed setup. Illustration of the probability of collision in each prediction step for each consecutive time step seen in (a) and the steps from an indicated threat to an actual braking intervention in (b).

6.3.2 Noise robustness

During the evaluation of the system's robustness to noise, it was noticed that the main parameter being sensitive to added noise was the position. The systems robustness has therefore been evaluated by low, intermediate and high noise on the position with noise values as given by Table 6.1.

Table 6.1: Noise values given as *Low*, *Medium*, *Medium/High* and *High* noise levels given on the x- and y-position of both the EGO and POV. Units given in meters.

Parameters	Low noise	Medium noise	Medium/High noise	High noise	Units
$\sigma_{x,EGO}$	0.25	0.50	0.75	1	meters
$\sigma_{y,EGO}$	0.25	0.50	0.75	1	meters
$\sigma_{x,POV}$	0.25	0.50	0.75	1	meters
$\sigma_{y,POV}$	0.25	0.50	0.75	1	meters

Noise added to the position will in the proposed solution have an impact on the BN as well as the IMM since both of them get the observed measurements as input. With a noisy position, the information nodes such as the *Lane alignment* node for the BN could give a false perception of the reality. The IMM on the other hand could have difficulties in deciding which motion model to best represent the driver intent, or consequently choose a faulty model. With both the BN and IMM being vulnerable to added noise, the model chosen for the predictions could be wrong hence the UKF is directly affected accordingly. This could lead to propagations of two colliding trajectories which in fact are not an imminent collision, thus both the *Collision detection* and most likely the *Collision avoidance* is affected.

To be able to foresee and prevent the CA system to behave incorrect, a noise robustness analysis is needed. Here, the analysis is presented by the use of a *Receiver operating characteristic* (ROC)-curve. This type of analysis method is mostly used to illustrate a system's performance for different threshold-levels, but could at the same time illustrate the performance with different noise levels. The analysis is made by evaluating a certain scenario with both a *positive case* (scenario leading to a collision) and a *negative case* (scenario with no collision). The scenario used for evaluation is the *left turn by POV across the EGO's path*, with two negative cases where one is the POV making the turn just before the EGO passes and one when the POV initiates a turn but stops to let the EGO pass.

The noise robustness evaluation was made with the collision detection threshold swept from (0% to 100%) for the four different noise levels presented in Table 6.1. The collision detection threshold was selected since it was found out that it had a bigger influence on the over all system performance, compared to the collision avoidance threshold. The chosen representation of the ROC-curve is thereafter shown as the number of accurately made braking interventions, denoted as the *True positive rate* (TP), in comparison to the number of falsely intervened negative cases, denoted as a *False positive rate* (FP). The ROC-curve for the given scenario is illustrated in Figure 6.36.

The reason for the chosen scenario to be evaluated with the different threshold levels is to determine the best fitted threshold giving a high TP whilst having a low

FP. The dotted line in the figure illustrates the reference to be above for the system to be better than a Collision Avoidance (CA) system randomly deciding if there is an imminent collision or not. The desired result is a system being minimally restrictive whilst being maximally evasive, hence not making any unwanted interventions. An optimal result of such a system is when having a 100% true positive rate whilst having 0% false positive rate.

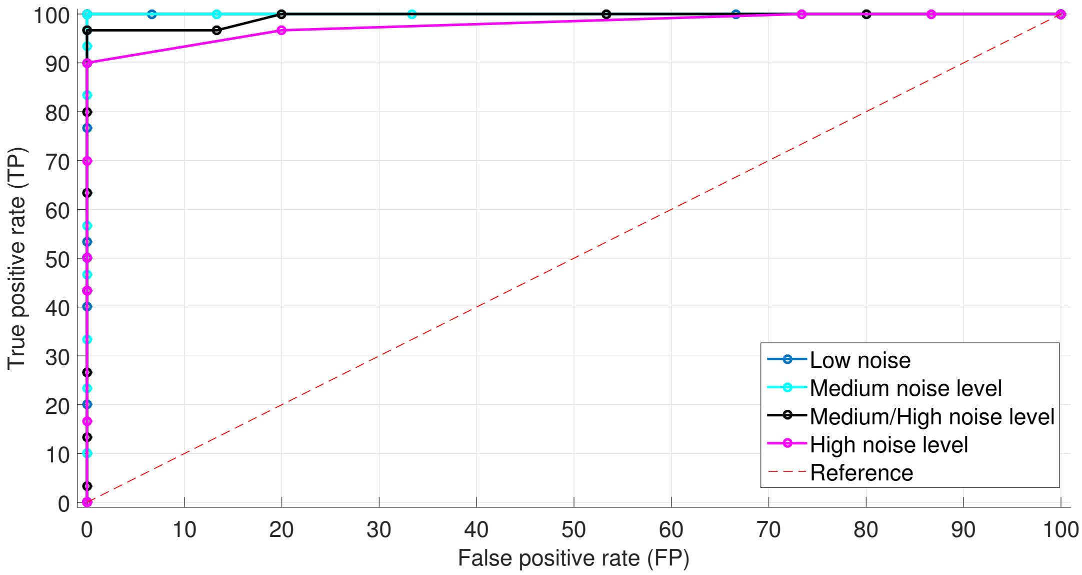


Figure 6.36: ROC-curve for the negative case when the POV performs a left turn across the EGO's path, thus passing just before the EGO. The illustration demonstrates the system's collision avoidance ability for a sweeping threshold level analysis with four different noise levels. A desirable system gives a TP/FP result above the dotted reference line.

Figure 6.36 and 6.37 demonstrates the system's noise robustness for the four different noise levels stated in Table 6.1, using the two different negative case scenarios. The threshold level is here zero at the upper right corner, TP and FP equal to 100, and decreases towards the lower left corner, TP and FP equal to 0. The threshold was swept from 0% to 100%, with a 2% interval hence equal to 0%, 2%, ..., 100%. This should generate 50 points along each curve, but as can be seen only around 20 values are illustrated, since some of them are overlapping.

Figure 6.36 illustrates that all noise levels are well above the reference line, where a collision detection threshold T_{CD} level above 25% generates a TP/FP at the upper left corner of the plot. Figure 6.37 is however revealing the systems limitations and difficulties with the negative case. The *Low* and *Medium* noise level is well above the reference line where the threshold level of 40% is indicated with the red squares in the figure. The system is thus having problems handling the scenario for the *Medium/High* and *High* noise levels. The *Medium/High* noise level is here kept around or slightly below the reference, whereas the *High* noise level is well below. Hence both of these noise levels generates an undesirable result.

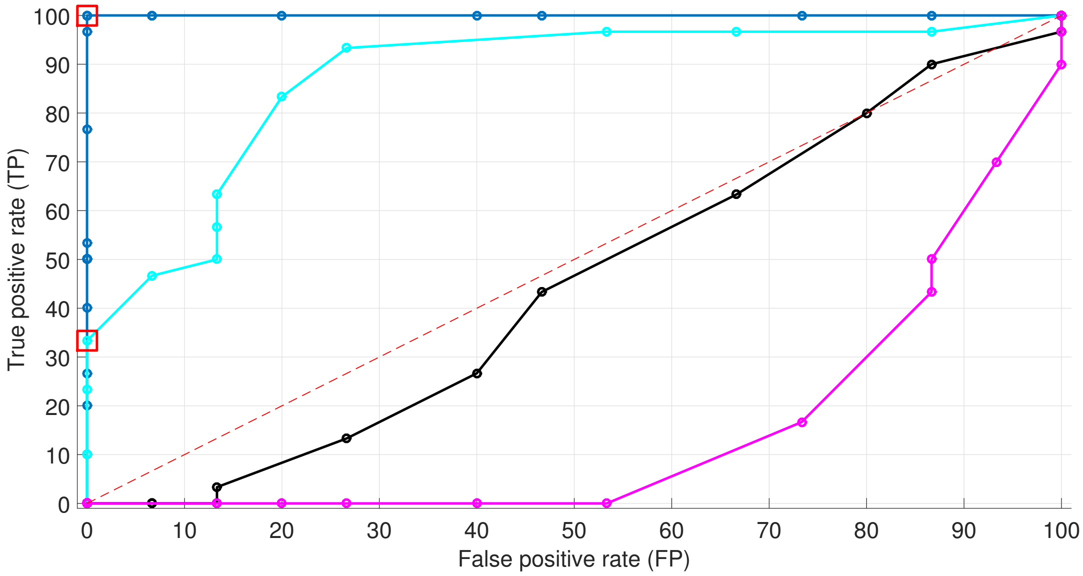


Figure 6.37: ROC-curve for the negative case when the POV attempts a left turn across the EGO's path, but stops before the EGO passes. The illustration demonstrates the system's collision avoidance ability for a sweeping threshold level analysis with four different noise levels. A desirable system gives a TP/FP result above the dotted reference line.

The explanation on why the first negative case is handled better than the second, is due to the presented solution's problem of handling approaching vehicles doing a rapid stop when aiming for a turn. The system will in this scenario still propagate a turn trajectory for some time steps hence both trajectories intersect, as explained in the *Abandoned turn* scenario of Section 6.2.1. This is mainly because the proposed IMM approach does not include a model that covers a performed deceleration motion.

The ROC-curves described in Figure 6.36 and the *Low* as well as *Medium* noise level ROC-curves in Figure 6.37 concludes that the desired threshold should be set above 40%. The collision detection threshold used for the proposed solution, T_{CD} was however set to 50%, due to this threshold being able to keep the false interventions to a minimum while keeping the correct braking interventions high.

Both the *Medium/High* and *High* noise levels in Figure 6.37 concludes the limits of the presented approach. The system could in this case not manage to push any of these noise levels toward a high percentage of correct braking interventions without introducing a high unwanted intervention percentage.

6.3.3 Long-term path prediction

In the subsystem of *Long-term path prediction* mainly two parts influence the performance of the overall system's ability to detect a collision, namely the *Bayesian Network* (BN) and the *Interacting Multiple Models* (IMM)-filter. Both of these subsystems receive measurements which are compared to computed predictions of what left, right and straight driving direction are set to be.

This section will explain the benefit of each part of the subsystem and their affecting parameters. The evaluation will be made with a performance comparison between the ordinary system setup, the BN turned off and the IMM turned off (using only a Constant Velocity (CV) model) as seen in Subsection A.

An evaluation has also been made of the BN's as well as the IMM's internal performance, presented in Subsections B and C respectively.

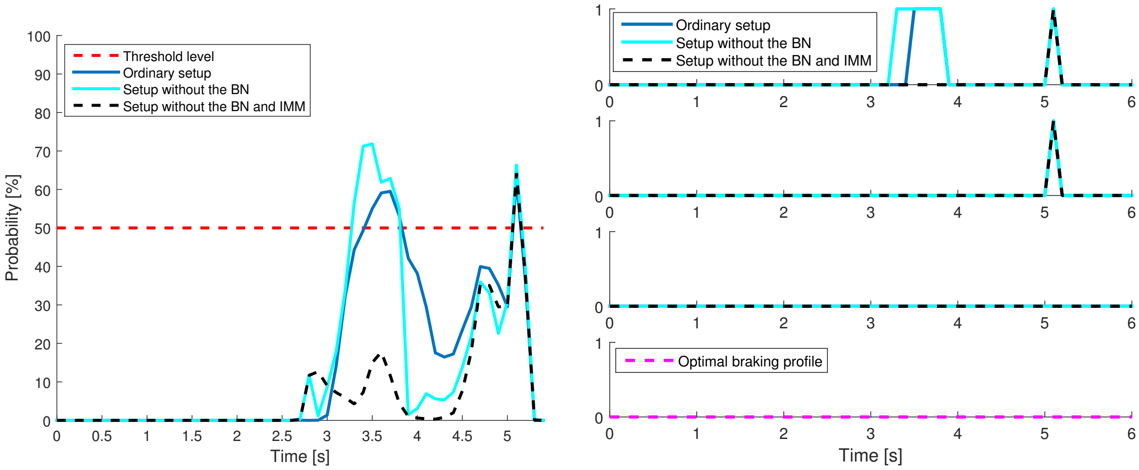
A Performance comparison

A performance comparison has been made using the scenarios *Abandoned turn*, *Crossing intersection* and *Left turn across path made by either the EGO or POV*. The comparison was made in regard to the advantage of using:

- The BN (evaluated by turning this part off, but keeping the IMM active)
- The IMM-filter (evaluated by only using one of the filter models such as the CV-model, hence both the BN and IMM are deactivated)

Abandoned turn

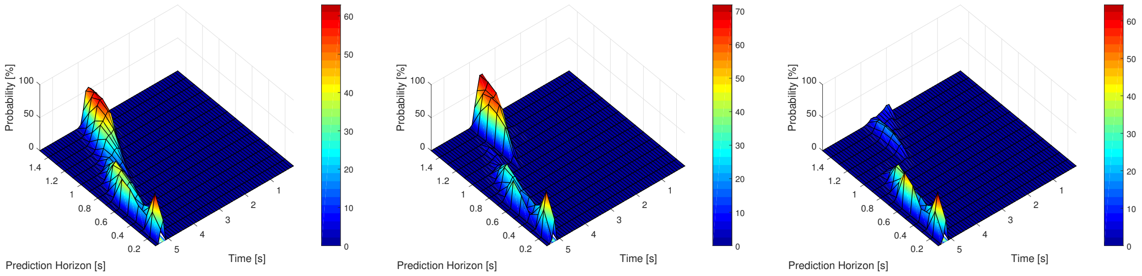
As previously mentioned, during the *Abandoned turn* scenario, the EGO vehicle should not perform an evasive manoeuvre such as a braking intervention, which according to Figure 6.38b did not occur.



(a) Collision probability

(b) Threat/brake assessment

Figure 6.38: Illustration of how each part influences the overall subsystems behaviour for the abandoned turn scenario, where (a) shows the probability of collision and (b) the threat/brake assessment for the three different setups.



(a) Ordinary setup **(b)** Setup without the BN **(c)** Setup without the IMM

Figure 6.39: A probability graph showing the detection rate of a probable collision for the abandoned turn scenario, where (a) illustrates the ordinary setup, (b) without the BN and (c) shows the affect of not using the IMM as well.

Figure 6.39 shows the collision detection rate along the whole prediction horizon for the three different cases where Figure 6.39a illustrates the ordinary case, Figure 6.39b shows the case without the BN and Figure 6.39c shows the case without the BN as well as the IMM.

When using the ordinary system setup, the information nodes associated with the POV could not distinguish a left from a right turn, hence both being equally likely at the beginning of the turn. This will therefore affect the overall perception of the likely manoeuvre giving a slightly lower likelihood of actually making a left turn. When disconnecting the affect of the BN, the subsystem fully relies on the IMM hence making a more, in this case, reliable perception of the likely course of action. Needless to say, this will give a higher likelihood of collision which actually is something not wanted, therefore the result from the ordinary case is more advantageous.

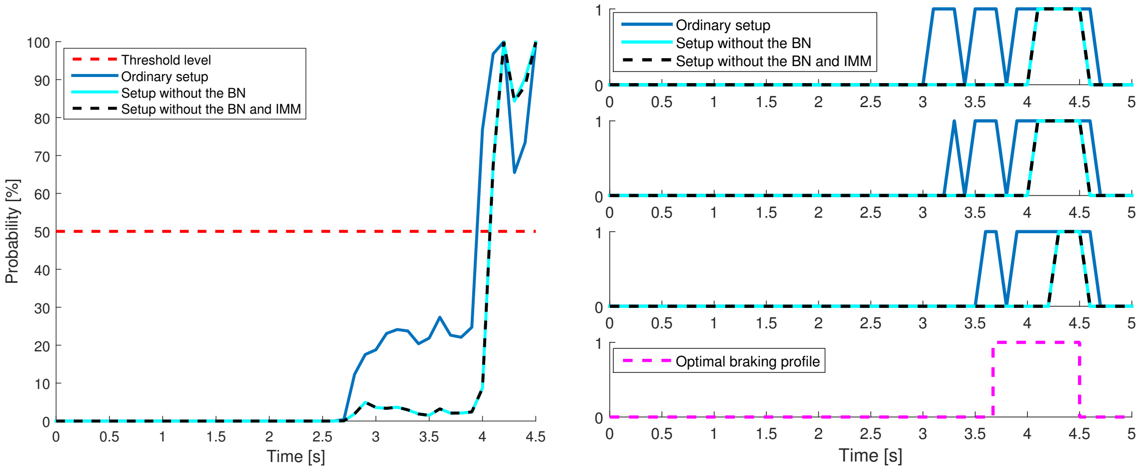
As illustrated by Figure 6.38 and 6.39, it is however evidently an even greater advantage of having both the BN and IMM deactivated in this particular case. This is explained by the presented system's difficulty in regarding a decelerating motion. Thus, when initiating a turning motion and then brakes to a full stop, it will be falsely modelled by both the BN and IMM as continuing the turn for some time steps. In this case will therefore the POV's predictions slightly intersect with the EGO's predicted trajectory and the collision probability goes above the threshold. As Figure 6.38b illustrates, it will however not lead to a braking intervention as this point in time is not evaluated to be the time step just before the PNR.

Crossing intersection

The evaluation of the crossing intersection scenario resulted in the system being equally good at detecting and evade a collision irrespective of if either the BN or IMM was active. This is explained by that the intended manoeuvre in the crossing intersection scenario is a straight path hence even without both BN and IMM, the system is being able to evaluate this scenario to be an imminent risk of collision. Therefore no further analysis will be conducted and the results can be seen in Appendix C.

Left turn across path made by the EGO

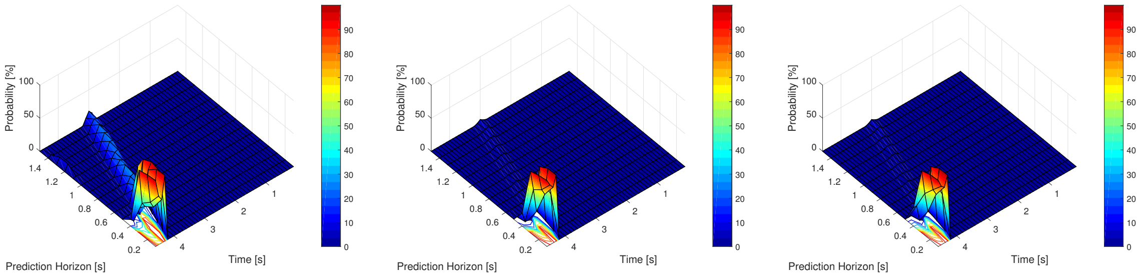
When the EGO makes a left turn across the POV's path, the long-term path prediction has a crucial part in the ability to detect an imminent collision. Figure 6.40 and 6.41 shows the difference in ability to detect a collision when using the ordinary system setup in comparison when not using the BN or IMM. Figure 6.40 illustrates the importance of all the parts in the subsystem which, with the ordinary system setup, has the ability to predict a left turn at an early stage making it possible to prevent a collision. Without either BN or IMM, a collision will occur, but nonetheless be mitigated as shown in Figure 6.40b.



(a) Collision probability

(b) Threat/brake assessment

Figure 6.40: Illustration of how each part influences the overall subsystems behaviour for the left turn across path by EGO scenario, where (a) shows the probability of collision and (b) the threat/brake assessment for the three different setups.



(a) Ordinary setup

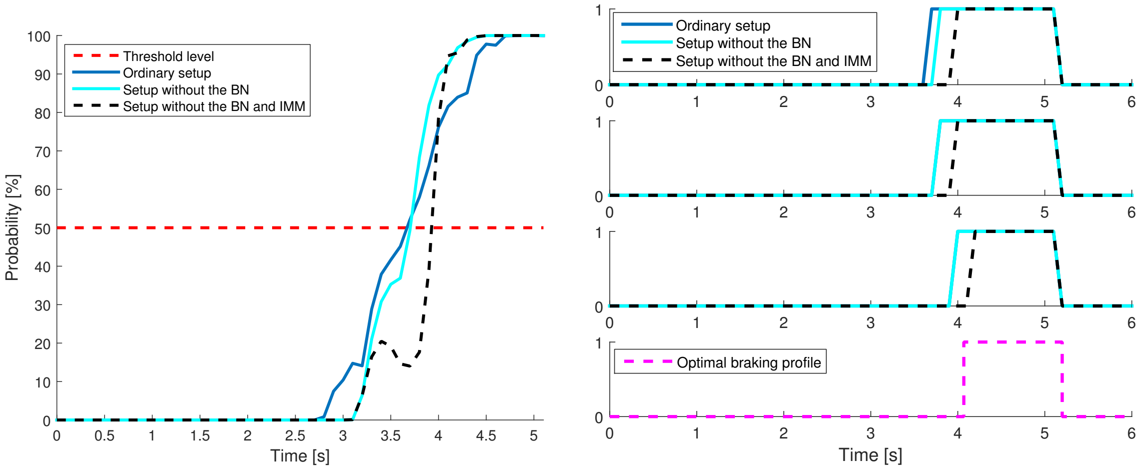
(b) Setup without the BN

(c) Setup without the IMM

Figure 6.41: A probability graph showing the detection rate of a probable collision for the left turn across path by EGO scenario, where (a) illustrates the ordinary setup, (b) without the BN and (c) shows the affect of not using the IMM as well.

Left turn across path made by the POV

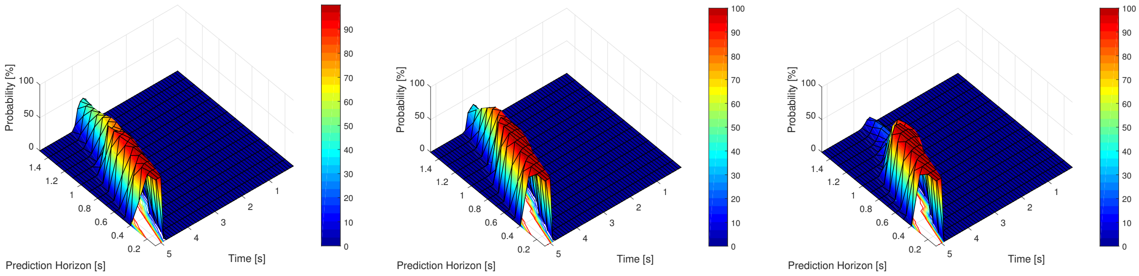
In the scenario where the POV makes a left turn across the EGO's path, the difference in the comparison of the different parts of the subsystem is less divergent as seen in Figure 6.42 and 6.43. The setup without the BN does not contribute to any difference due to the lack of evidences given from the POV in the original scenario setup. The main advantage is given in the comparison of using IMM (*Ordinary setup*) and only using a CV model (*Setup without the IMM*), which gives a slight difference in the ability to detect a collision and thereafter avoiding it. The setup without the IMM will nonetheless be able to mitigate the collision in such an extent that the collision is nearly avoided.



(a) Collision probability

(b) Threat/brake assessment

Figure 6.42: Illustration of how each part influences the overall subsystems behaviour for the left turn across path by POV scenario, where (a) shows the probability of collision and (b) the threat/brake assessment for the three different setups.



(a) Ordinary setup

(b) Setup without the BN

(c) Setup without the IMM

Figure 6.43: A probability graph showing the detection rate of a probable collision for the left turn across path by POV scenario, where (a) illustrates the ordinary setup, (b) without the BN and (c) shows the affect of not using the IMM as well.

B The BN's affect on system performance

The information nodes which the BN is built up by demands correctly given data to be able to make accurate long-term path predictions. However, if the data to the BN would be wrongfully given (such as a turn signal giving the wrong direction than the intended or wrong map information given) the system should be robust enough to not be affected solely by this.

An evaluation of the BN's affect on the whole systems performance is made through the scenario left turn by the EGO across the POV's path. A comparison has been conducted in regard to the ordinary system setup, a wrongfully given map information (stating an absence of a road to the left) and with a wrongfully given turn signal (stating to go right instead of left) as well. The result is illustrated in Figure 6.45 and 6.44.

A conclusion from the results can be made in regard to how crucial the nodes are to correctly compute the long-term path prediction. The results on the other hand also illustrates that even though the BN presents an inaccurate decision, the outcome is not solely affected by this decision. This is explained by the IMM's ability to counteract a decision by the BN with the IMM's computation of the near-term path prediction. A collision could therefore, in this case, not be fully avoided with the wrongfully given information to the nodes but nonetheless be mitigated. The main conclusion is therefore that the outcome from the BN does not solely decide how well the long-term path predictions of each vehicle will be.

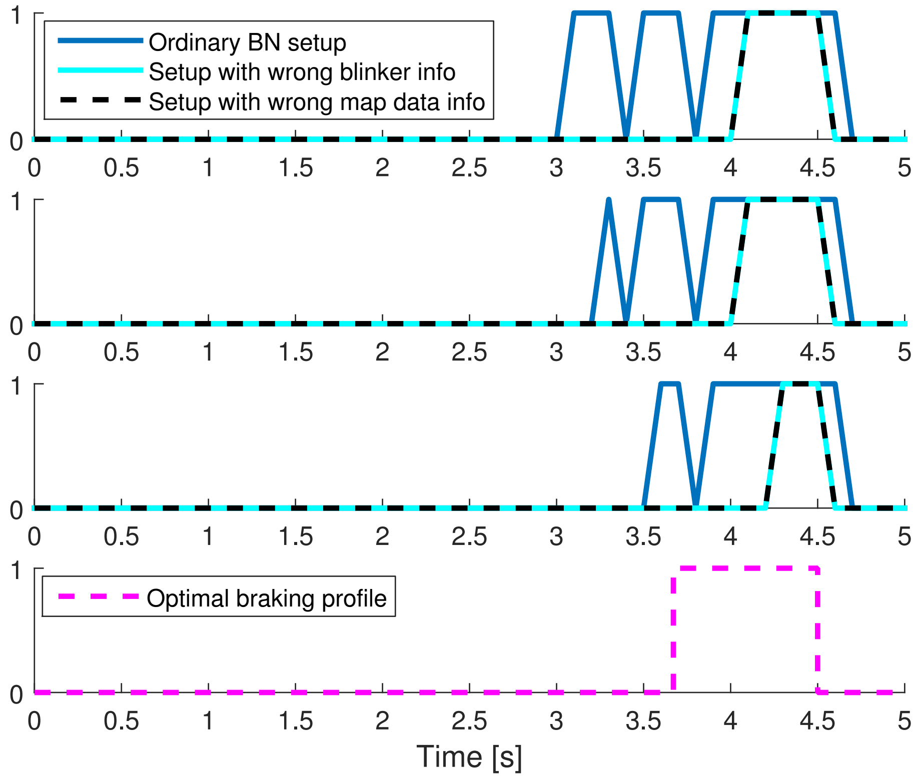
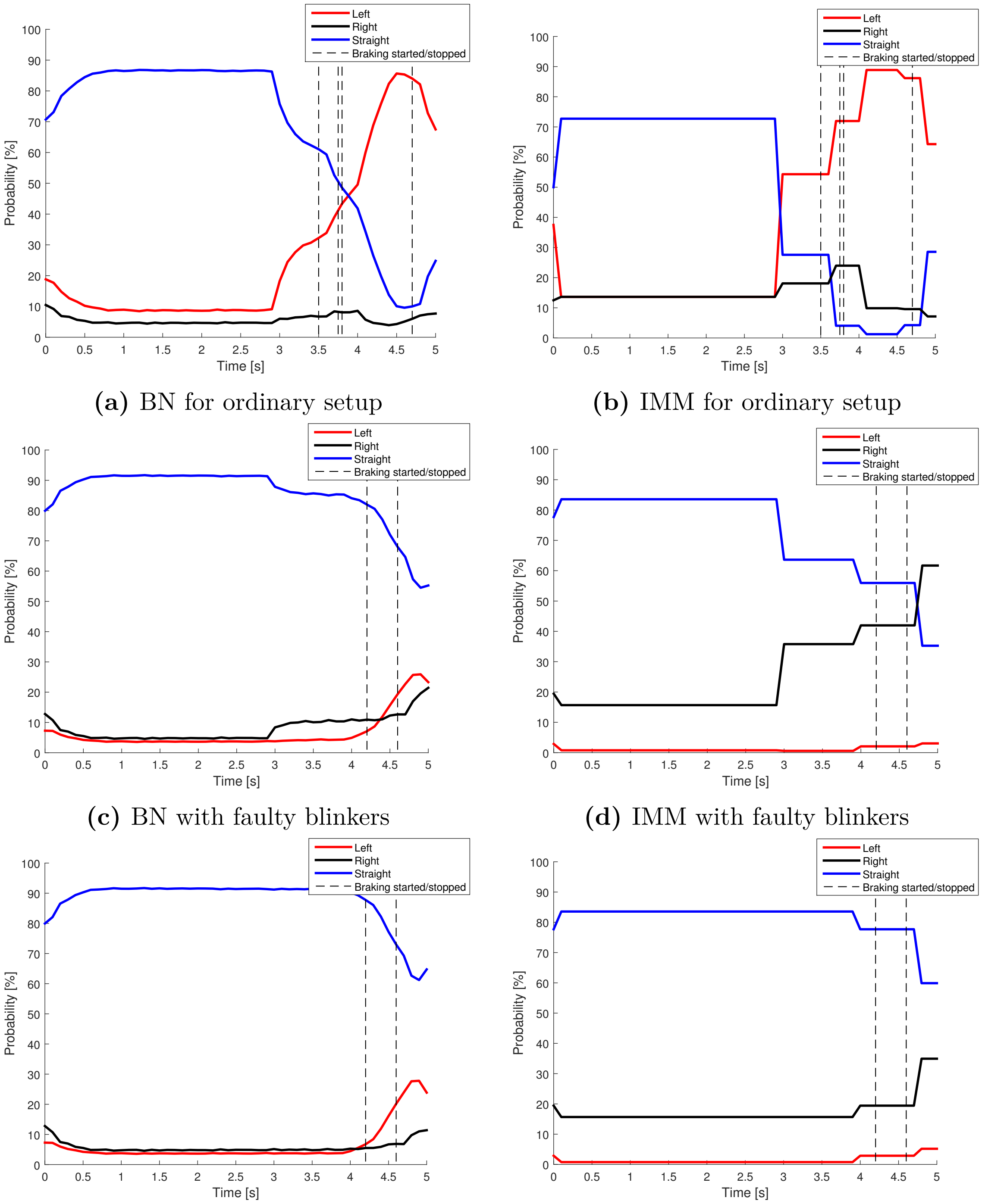


Figure 6.44: The figure illustrates the difference in collision detection and at last the ability to brake in time for the ordinary system setup, with wrongfully given turn signal and with wrongfully given map data.



(a) BN for ordinary setup

(b) IMM for ordinary setup

(c) BN with faulty blinkers

(d) IMM with faulty blinkers

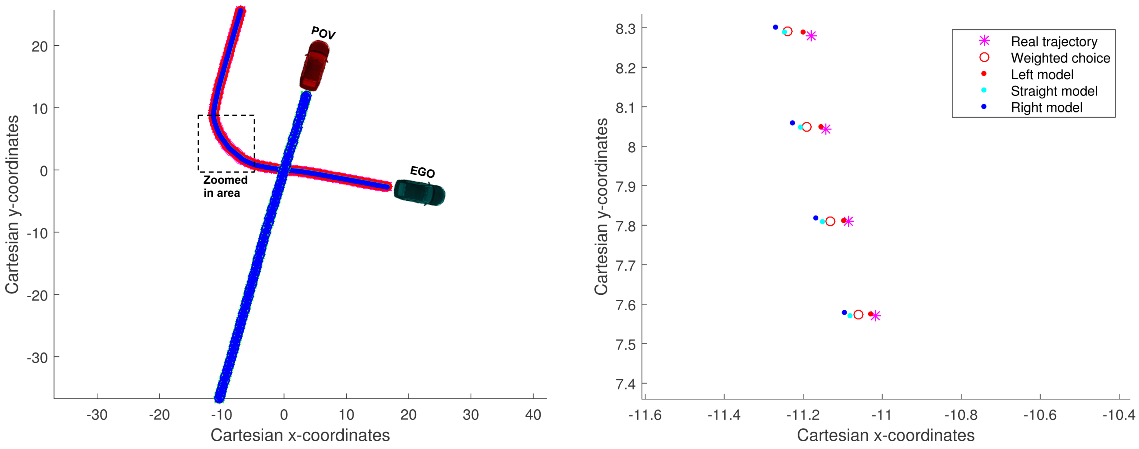
(e) BN with faulty map info

(f) IMM with faulty map info

Figure 6.45: The plots illustrates the BN and IMM weights for a left turn by the EGO across the POV's path. Plot (a) and (b) shows the ordinary system setup, (c) and (d) wrongfully given turn signal (depicted a right instead of left turn to be made) and (e) and (f) demonstrates when having wrongfully given map data (denoting an absence of a road to the left).

C The IMM's internal performance

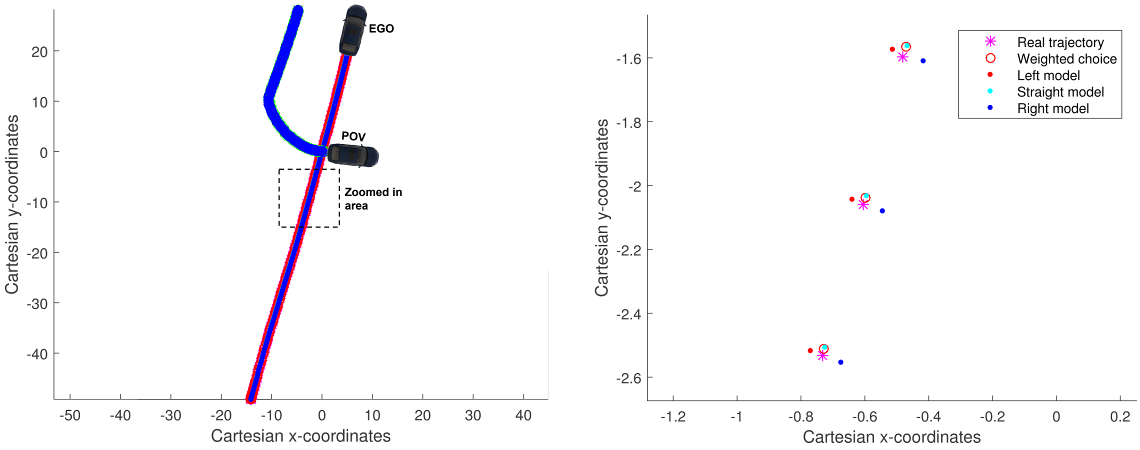
The IMM's ability to follow either a left or a straight going trajectory is for the EGO shown in Figure 6.46 and 6.47 respectively.



(a) Trajectory overview

(b) Zoomed part of the trajectory

Figure 6.46: Illustration of how the models(left, straight and right) in the IMM is spread along a left going trajectory, where the measurements and weighted choice from the three models ability to reflect the measurement. An overview of the trajectory is shown in (a), whereas (b) shows a zoomed part of the marked area on the trajectory



(a) Trajectory overview

(b) Zoomed part of the trajectory

Figure 6.47: Illustration of how the models (left, straight and right) in the IMM is spread along a straight trajectory, where the measurements and weighted choice from the three models ability to reflect the measurement. An overview of the trajectory is shown in (a), whereas (b) shows a zoomed part of the marked area on the trajectory

Figure 6.46 demonstrates the IMM's ability to change from a straight going path to a left turn, where the scenario with the trajectory is illustrated in 6.46a. The top of Figure 6.46b shows that the measurement as well as the weighted choice is aligned near the model for straight path (green). Along the trajectory, both the measurements as well as the weighted choice will move towards the model for left going path (and eventually changing the model for the prediction calculations from straight to left model).

Figure 6.47 on the other hand illustrates the IMM's ability to choose the straight going model as the most appropriate model to represent the vehicles driving direction. Figure 6.47b demonstrates the weighted choice to more or less align with the straight going models filtered output.

6.3.4 Collision detection

The main influence on the collision detection subsystem is the number of points used for representation of each vehicle's area. The number of points used in the collision detection algorithm has a big influence on the computational complexity, hence an evaluation was made to see how much the number of points can be decreased to still give a reliable result.

Table 6.2 presents the results where 847 points are used as a reference meaning that the points are spread out with approximately 10 cm in between them. As seen in the table, there is not a big difference in probabilities with a higher number of points. The lower limit can be seen to be at 3x5 points (3 on the width and 5 on the length) points, thus 3x3 points will still not differ too much from the reference values. The biggest difference can be seen when an even number of points is used, i.e. when the centre of the vehicle's width and/or length is not represented by a point. It can be seen in the table (with the help of Appendix D), that the scenarios giving the biggest difference in probability, are the configurations where the vehicles overlap in any kind of way. The reason is that in those cases, the points with the highest probability are the points inside the overlapping vehicle, hence results in a lower probability if these points are not considered.

Table 6.2: This table shows how the reliability of the collision detection subsystem is decreasing with the number of points used for each vehicle. A reference with 847 points was used, which should correspond to infinity (marked in grey). Width, length is the number of points spread out evenly across the width and length of the vehicles; total no points, is the total number of points spread out, i.e. equal width*length; distance points, is the approximate distance between the points. All the probabilities are given in percent where the differences below 0.01 percent is marked with "-", cells of differences above 5% are marked in yellow and percentages above 10% are additionally marked with red text. For each different number of points is the same variance used for each scenario, further details about the scenarios are given in Appendix D.

Width, length			9, 23	21, 5	5, 13	5, 9	3, 9	3, 7	3, 5	2, 6	3, 3	2, 2	1
Total no. points		**847**	**207**	**105**	**65**	**45**	**27**	**21**	**15**	**12**	**9**	**4**	**1**
Distance points		0,1	0,2	0,3	0,4	0,5	0,6	0,7		0,8			
Scenario		**Ref.**	**Differences in probability**										
Apart	P1	2,67E-06	-	-	-	-	-	-	-	-	-	-	-
	P2	0,04	-	-	-	-	-	-	-	-	-	-	0,04
	P3	0,08	-	-	-	-	-	-	-	-	-	0,02	0,06
	P4	0,08	-	-	-	-	-	-	-	-	-	0,04	0,04
	P5	0,08	-	-	-	-	-	-	-	-	-	-	-
Next to eachother	P1	5,95E-06	-	-	-	-	-	-	-	-	-	-	-
	P2	24,60	-	-	-	-	-	-	-	-	-	-	24,18
	P3	49,04	-	-	0,02	-	-	0,02	0,40	0,03	0,58	10,27	38,63
	P4	49,04	-	-	-	-	-	-	-	0,28	-	24,45	24,45
	P5	30,58	-	-	-	-	-	-	-	-	-	-	-
Overlapping	P1	38,51	-	-	-	-	-	-	-	13,91	-	13,91	13,91
	P2	68,04	-	-	-	-	-	-	-	24,59	-	24,59	24,59
	P3	76,34	-	-	-	-	-	-	-	27,59	-	27,59	27,59
	P4	76,79	-	0,03	0,01	0,02	0,02	0,18	0,44	28,03	0,44	28,03	28,03
	P5	76,79	0,01	-	0,04	0,09	0,09	0,04	0,09	28,03	2,61	33,33	33,33
	P6	76,79	-	-	-	-	-	-	-	28,03	-	52,19	52,19
EGO into POV	P1	49,04	-	-	-	-	-	-	-	1,70	-	1,70	1,70
	P2	68,80	-	-	-	-	-	-	-	2,38	-	2,38	2,38
	P3	76,79	-	-	-	-	-	-	-	2,66	-	2,66	2,66
	P4	76,70	-0,01	0,61	0,16	0,40	0,40	2,45	2,45	10,08	2,45	10,28	10,28
	P5	76,79	0,31	0,14	0,42	0,42	0,42	0,42	0,42	2,66	0,42	29,45	29,45
	P6	76,78	-0,01	0,02	-0,01	-0,01	0,35	0,41	0,41	2,66	0,41	64,99	73,07
POV into EGO	P1	76,79	0,31	0,14	0,42	0,42	0,42	0,42	0,42	2,66	0,42	29,45	74,92
	P2	76,70	-0,01	0,61	0,16	0,40	0,40	2,45	2,45	10,08	2,45	10,28	75,67
	P3	76,79	-	-	-	-	-	-	-	2,66	-	2,66	76,36
	P4	68,80	-	-	-	-	-	-	-	2,38	-	2,38	68,67
	P5	49,04	-	-	-	-	-	-	-	1,70	-	1,70	49,01
	P6	0,81	-	-	-	-	-	-	-	0,03	-	0,03	0,81
EGO turn, POV straight	P1	1,17E-08	-	-	-	-	-	-	-	-	-	-	-
	P2	2,56	-	-	-	-	-	-	-	-	-	-	-
	P3	76,70	-	0,60	1,18	1,26	1,26	1,26	1,26	1,26	1,26	1,26	14,93
	P4	76,79	0,01	0,06	0,91	0,11	0,25	1,05	5,10	2,40	5,10	14,44	14,44
EGO and POV turn	P1	4,29E-07	-	-	-	-	-	-	-	-	-	-	-
	P2	34,75	-	-	-	-	-	-	-	-	-	-	26,22
	P3	76,77	0,03	-0,01	0,45	0,45	0,45	1,08	0,45	1,24	2,97	29,12	29,12
	P4	2,36	-	-	-	-	-	-	-	-	-	-	-
	P5	2,74E-12	-	-	-	-	-	-	-	-	-	-	-

6.3.5 Collision avoidance

The *Collision Avoidance* subsystem was previously developed by [17] as a modular block to be used in future research works. During the development of the presented general and effective long-term path prediction as well as the collision detection algorithm, the aim of the thesis work was to utilise the previously developed modular *Collision avoidance* subsystem into the whole system. The modularity as well as performance was therefore evaluated alongside the overall systems performance.

The result is a subsystem having predefined *Attraction sets* in regard to the EGO's performance parameters. These attraction sets were built up around the EGO's ability to avoid a collision by emergency braking (as stated before with a maximum deceleration of 8.5 m/s^2). The *Point-of-No-Return* (PNR) was thereafter found through a membership test of the attraction sets using the vehicle's current velocity, DTC as well as reaction time of 0.2 seconds.

During the presentation of the overall system performance result in Chapter 6, an evaluation of the *Collision avoidance* subsystem was also conducted. The main objective of this subsystem is to make the overall system maximally evasive whilst being minimally restrictive. It will therefore need to evaluate each probability of collision being above the given collision detection threshold. This is done for the possibility to find the moment right before going into the PNR. The Collision Avoidance (CA) system then had the ability to intervene at the latest time possible hence being able to avoid or at least mitigate the oncoming collision. This has been demonstrated alongside the presented results as the ability to disregard a signal of danger only being present for a short period of time, and in most cases being able to intervene at the latest time possible.

7
Discussion and concluding remarks

The general objectives for the thesis work were stated in Section 1.3 as:

- Proposing a method to merge pertinent information of the surrounding environment in order to accurately predict the behaviour of oncoming traffic.
- Developing a stochastic solution evaluating the risk of collision using the previous mentioned predictions.
- Defining a formal, robust decision making procedure for least-invasive braking interventions.
- Evaluating the proposed Collision Avoidance (CA) system with both simulations as well as with real-time implementation.

The aim of this chapter is to discuss the results presented in Chapter 6 and draw conclusions of the system's ability to cope with the formulated problems, stated in Section 2.1.

This chapter provides a profound analysis of the performance, stability and robustness of the proposed system through four sections. Firstly, Section 7.1 addresses the development of the three individual modular blocks. Secondly, Section 7.2 discusses how the evaluation of the whole system was made through different application methods. Furthermore, suggestions of future work will be given in Section 7.3. Finally, in Section 7.4 can a presentation of some concluding remarks be seen.

7.1 System design

This section has been divided into the three parts *Long-term path prediction* in Subsection 7.1.1, *Collision detection* Subsection 7.1.2 and *Collision avoidance* in Subsection 7.1.3 as each of them reflect the first three general objectives in Section 1.3. The performance of the full system relies on the individual performance of each part of the overall system, hence the design of the three different parts will be discussed separately.

7.1.1 Long-term path prediction

The long-term path prediction block can subsequently be divided into two system operators, namely a long-term and a near-term prediction. The ability of predicting along a far horizon lies within the properties of the *Bayesian Network* (BN) whereas the near future is better predicted by the *Interacting Multiple Model* (IMM) filter. Hence by fusing the two sub blocks' predictions, the possibility to identify the driver's intent is given.

For improvement of the results, both the BN and IMM rely on specific factors for each calculated outcome. The BN is dependent on the observations it can make, or more specifically on how many unique information nodes it can use in the long-term prediction and the accuracy of the information given by these. The performance and robustness of this part are improved by the number of observations, i.e. the number of information nodes that can be used. If only one observation is available, such as for example a *turn signal*, this will have the full power of decision for the BN. The system's performance could therefore be vulnerable to this sole observation. If instead several different observations are used, a false observation from one node could be counteracted by the other nodes' observations. The performance can then be concluded to be better with an increasing number of accurately observable nodes, but this would also generate a higher computational heaviness.

In the scenarios used for evaluation, the road geometry was given in advance, often stating a four-way intersection where a left, right and/or straight driving direction was possible. To already have information of the road geometry was considered as a valid assumption by the authors due to the extent of the development made within digital maps. The other observations were constrained by the performance of the considered on-vehicle sensors such as the Inertial Measurement Unit (IMU) sensor, wheel speed sensors and the radar. The quality of the long-term predictions for the POV is therefore consequently constrained by these sensors' capability of detecting the vehicle at an early stage. With the available sensors for the scenario setup, the predictions instead acted more as a near-term observer than a long-term. If instead additional sensors could be added on the EGO, for example giving information of the POV's lane alignment or activated turn signal, a better performed long-term prediction could be computed.

The IMM on the other hand, is built up by a number of motion models representing possible manoeuvres to be made by a driver. A model could thus describe a turning motion either to the right or left as well as with different turning angles. A model could also describe a braking (deceleration) or acceleration motion. However, with an increasing number of models, the more computational heavy the overall system will become. This since the models are run in parallel. In the presented solution, the IMM has a restrained possibility to only choose between a left, right or straight model to best represent the driver intent, which has been noted to not always give the correct prediction. This was evident in the scenario where the POV was initiating a turn, but stopping to let the EGO pass i.e. the *Abandoned turn* scenario. Here, before the POV would get to a full stop, the chosen motion model was either a right or left turning model due to both an initiated turn and deceleration. This did in some cases infer that the driver was still turning, when instead the vehicle was braking to a full stop hence inevitably ended up in a danger of collision. Including one motion model describing constant acceleration and one describing a constant deceleration might yield a better perception of the reality. On the contrary, in comparison to a motion model describing only a single motion such as the Constant Velocity (CV) model, the IMM approach is more reliable as shown in the analysis from Section 3. This is primarily due to the IMM's possibility to match the driver intent from a set of pre-defined motion models instead of only being able to account for one models ability to follow the probable pathway.

A common key factor for both the BN and IMM is the sensors reliability. In both simulations using PreScan and by the use of the data collected from real-tests, the input data was very accurate (position variance up to maximally 10 cm), hence was considered to give sufficiently reliable data. As the data for the POV was considered to be collected using a radar, this data was regarded to be more inaccurate.

Since the IMM considers both the input data and estimated affecting noise to be of equal importance, it is crucial that the noise is correctly estimated. If the noise were to be estimated wrongly, being either too large or too low than the reality, this could consequently have negative affects on the filtering performance. The IMM itself requires some noise on the input (as it most often has) for the ability to differentiate between the different models set up by the designer. Moreover is also the sampling frequency a key factor in the ability to differentiate between the models, as described in Section A. For instance with a too high sampling frequency (higher than 10 hertz), the models will be updated too often hence the predictions from the filter seem to be at the same point as the previous time step, for each model. A lower sampling frequency will thus generate more noticeable changes between each consecutive update, hence the IMM becomes more reliable.

The fused predictions from both the BN and IMM result in a motion model giving a good representation of the driver intent, which in the last part of the long-term path prediction will be used to compute the predicted path some time steps into the horizon. This is, as described in Section 3.4, accomplished with the use of the prediction part of the *Unscented Kalman Filter* (UKF).

Even if the process is easily computed, there are however three main factors affecting the final outcome which are the noise covariance matrix, the weight coefficients and lastly the number of prediction time steps. Both the noise covariance and the weight coefficients are in the presented solution chosen to be least restrictive, described in Section 3.4, hence could not be optimised to any further extent. The third factor, the number of prediction time steps, is instead the main affecting parameter for the ability to perform accurate perception of future time steps. At a velocity of around 50 km/h, as studied in this thesis, a full stop using an Autonomous Emergency Braking (AEB) system could be obtained within around 1.1 seconds [42]. The number of prediction time steps would therefore need to cover at least 1 second into the future. However, the prediction time could not be set to cover a too large horizon due to the fact that with a longer prediction horizon, it will consequently give a higher sensitivity to have a wrongfully chosen model. This design challenge is illustrated in Figure 7.1. The prediction horizon of 15 time steps, equal of covering a future time span of 1.5 seconds, is chosen for the ability to asses and intervene an oncoming collision whilst still having low probability of false interventions.

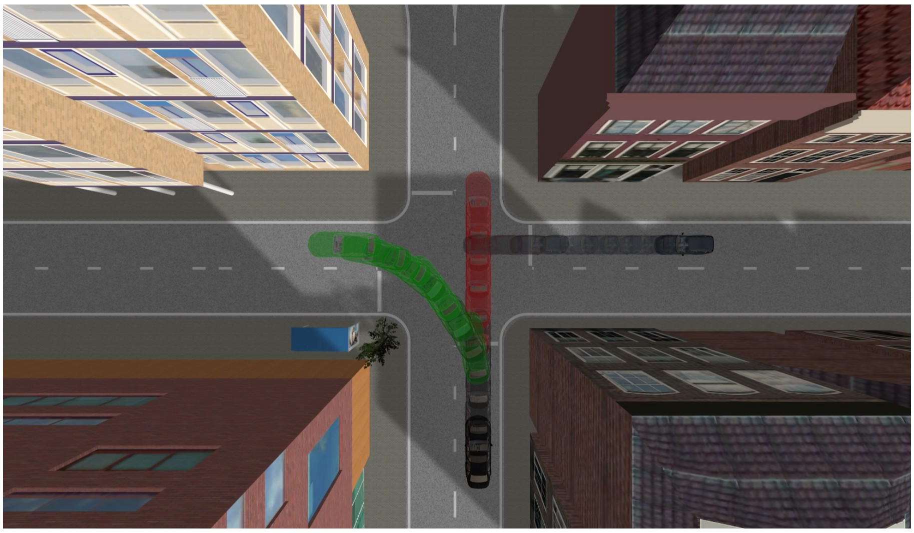


Figure 7.1: Illustration of a too long prediction horizon giving an unwanted result. The green predictions represents the meant path whereas the red represents the wrongfully predicted path.

7.1.2 Collision detection

The collision detection subsystem has been seen to be able to detect a vast variety of collision configurations. Since the system relies on the quality of the predictions, consequently the reliability of the system is decreased as the uncertainties increase. As the variance defines the uncertainty, the variances will also increase along the prediction horizon. Too large variances will thus lead to lack of interventions since the probabilities of collision never reach the threshold. The threshold could moreover be lowered, but would then most likely result in a higer number of false interventions as illustrated by the ROC-curve described in Section 6.3.

Since the collision detection algorithm is dependent on knowledge of the vehicle's size, some kind of classification algorithm determining a vehicle's area will be crucial in future developments. This knowledge is needed to be able to evaluate the risk of collision as well as calculating the collision time interval correctly. A classification algorithm like this could moreover lead the collision detection algorithm to be useful with road participants other than cars.

The biggest drawback of the collision detection subsystem is the computational complexity. For the evaluation made in real-time, problems occurred due to the computation time of each subsystem, in particular because of the collision detection subsystem's influence. A decrease in the number of points denoting a vehicle's area, will decrease the complexity but also the reliability of the system as described in Chapter 6.3.4. Research on how to find a numerically lighter approximation of the integral (computing the probability of collision) is therefore recommended by the authors. Another improvement could be to span a vehicle's area in a different way, e.g. place the points in some other configuration. Additionally, can future investigations of a two step method for the collision detection be made, similar to [31]. The first step is thus to generate a rough estimation of the collision risk. Secondly, if a threat is detected, a more accurate evaluation would be performed.

7.1.3 Collision avoidance

The CA system described in Chapter 5 was designed with the aim to be least restrictive whilst being maximally evasive, i.e. avoiding or mitigating a collision with few false interventions. This is done by keeping the EGO from travelling beyond the Point-Of-No-Return (PNR) with the aid of the emergency braking system (EBS).

The research and design towards the long-term prediction and collision detection subsystems as part of the presented CA system, was able to benefit from the tools developed by [17]. The modularity as well as effectiveness of the CA system has been evaluated through simulations to be fully executable within the urban intersection environment, as presented in Chapter 6.

The presented results is confirming the CA systems ability to disregard the collision risk given by the *Collision detection* subsystem having a time interval of maximally 1 time step. It also demonstrates the systems ability to evaluate an accurate PNR, before triggering a braking intervention. A crucial part of the system is hence to be able to evaluate a detected danger to be an inevitable collision and at the same time only intervene when the collision is considered to be unavoidable by the driver.

7.2 System application

The system has been tested and evaluated through simulations mostly in MATLAB/Simulink but later also using PreScan. Each block of the CA system has been evaluated independently of the others, hence keeping the modularity of the system. The aim of the thesis work has always been on real-time implementations and the preparations for this, using AutoBox [43] as a tool, was conducted during the last 3-4 weeks of the thesis. The ability to finalise the tests were unfortunately not possible, where the problem could either lie in the code generation from the Simulink to AutoBox model or by the computational performance of the in-vehicle computer.

Regarding the system's performance, there are some system properties generating worse or better results. The underlying properties of bad results could be:

- For the BN, there were too few observations mostly for the POV, but also for the EGO.
- Not enough manoeuvres to represent the probable directions.
- The choice of Extended Kalman Filter (EKF) as the filtering technique.

The first point could be explained by the fact that if more information about the POV is observable such as turn signal and lateral alignment, an earlier recognition of probable direction could be gained. More possible manoeuvres to choose from would be beneficial both for the BN as well as the IMM, since this could describe the probable direction with a higher accuracy. The extended set of different manoeuvres could include one model for acceleration and one for deceleration. The third point regards the choice of using EKF as filtering technique. An UKF could give higher precision, but will however consequently result in higher computational complexity.

Another dimension to the filtering would be to include a *smoothing* technique similar [30], which could help in the ability to differentiate between a turning and a straight going motion.

On the contrary, the factors that could generate better results than expected are given by:

- ”Perfect data” simulations
- Faulty trajectory predictions still detecting a threat in the vicinity of the driver intended trajectory

In both the simulations and in-vehicle tests, the given measurement data has been almost perfect in the aspect of only low noise added. The ability to perform tests with radar has been done through PreScan, but not in the in-vehicle tests. If radar is to be used to detect other objects along the path, the proposed solution will need to be evaluated through in-vehicle tests. This is due to PreScan operating in a too ”perfect environment”, hence giving measurements not representing possible sensor imperfections. A demonstration of the systems critical noise limits was presented in Section 6.3.2, which also gave information of the manageable sensor imperfection limits.

The second factor is given by the *Long-term path prediction* subsystems limits in accurate prediction performance. In scenarios similar to the illustration in Figure 7.2, a faulty predicted trajectory could still lead to a correct threat assessment as well as braking intervention. The braking intervention will thus be actuated earlier than it supposedly should as seen to the driver intended trajectory, hence could lead to an unacceptable intervention in the eyes of the consumer. This is therefore something that need more attendance in future work.

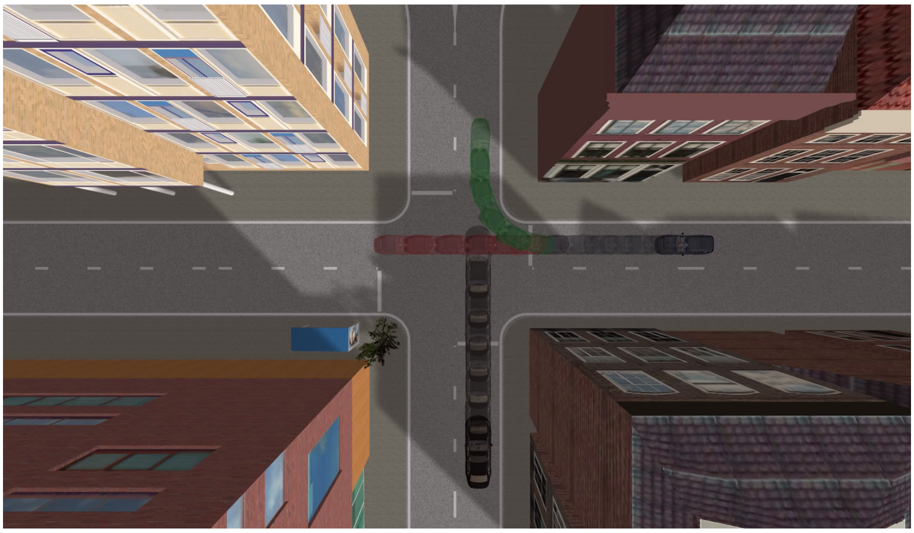


Figure 7.2: Illustration of a factor giving a well-performed system even with faulty predicted trajectory. The green predictions represents the meant path where as the red represents the wrongfully predicted path.

7.3 Future work

As a result of this thesis work, the presented CA system is designed to able to, in a robust and efficient way, detect and avoid or mitigate an oncoming collision. But even though the developed system manages to solve the stated problems, it still has room for improvements.
The major improvements recommended for future work in the aim of getting a commercially available system is to:

- Incorporate more statistical information into the BN-nodes. These nodes could thereafter be an underlying basis for a look-up library applicable for different types of sensors. This is because the presented BN is developed to comprehend information given by a sensor regardless of which sensors used. An inclusion of V2V or V2I is therefore a possible incorporable solution.
- Include more motion models into the IMM describing accelerating and decelerating trajectories.
- Obtain the road curvature for each specific road, to be used for both the IMM and UKF. As of now, the subsystems are given a constant curvature radius acting as a generally good estimate. The system would however improve the computed predictions with an accurate curvature.
- Constrain the trajectories of each vehicle according to the path restrictions given by a digital map. These restrictions could be limited to the drivable pathway along the roads.
- Include other possible intervention methods such as turning or acceleration, giving the possibility to at most times avoid a collision instead of only mitigating it. This could be solved using a Model Predictive Control (MPC) structure which gives an ”optimal trajectory follower”.

As a minor upgrade to the system design, with the prediction horizon of 1.5 seconds, there is a possibility to provide graphical and/or audio information of an oncoming collision to the driver. This information could be as a warning to enforce an intervention earlier than the human would predict it would need to do. An upgrade like this could hence give a more commercially acceptable solution.

The stated improvements would enhance and solve the major short comings of the system design. The developed system would thus foremost need to be further evaluated through real-vehicle tests. The system was developed for and implemented in real-vehicle tests, but due to unknown computational problems within the AutoBox environment, no successful results were obtained. Any extended development of the proposed system is therefore recommended to be put on studies on getting real-vehicle tests fully functioning.

7.4 Concluding remarks

The results and analysis in Chapter 6 show the systems robustness and ability to perform in the systems intended environment such as urban intersections. Even though the discussion in Section 7.1 and 7.2 highlights some discovered problems, the authors opinion is that the proposed system design fulfils the intended objective stated in the first paragraph of Chapter 7.

The first objective is handled by the long-term path prediction block, the second by the collision detection block and lastly the third objective is obtained through the collision avoidance part of the system. The fourth objective was partially fulfilled as the real-vehicle implementations was obstructed by software issues. To be a commercially acceptable product, some improvements are proposed as described in Section 7.3, but the authors recommend to first make sure that the system is implementable within a vehicle's computation system such as for example an AutoBox.

Bibliography

[1] European Commission, “Intelligent Transport Systems - Innovating for the transport of the future”, 2015. `http://ec.europa.eu/transport/themes/its/index_en.htm` [2015-10-20].

[2] Kamal, A.S. et al., 2014. “A Vehicle-Intersection Coordination Scheme for Smooth Flows of Traffic Without Using Traffic Lights”, pp. 1–12.

[3] Hafner, M. & Cunningham, D., 2011. “Automated vehicle-to-vehicle collision avoidance at intersections”.
Available at: `http://web.mit.edu/ddv/www/papers/ICA_Conf.pdf` [2015-02-02]

[4] Von Eichhorn, A., Zahn, P. & Schramm, D., 2013. “Automatic generation of intersection topologies using numerous GPS traces”, 2013 IEEE 5th International Symposium on Wireless Vehicular Communications, WiVeC 2013 - Proceedings.

[5] Basma, F., Tachwali, Y. & Refai, H.H., 2011. “Intersection collision avoidance system using infrastructure communication”, IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, pp. 422–427.

[6] Tunçer, O., et al., “Vision based lane keeping assistance control triggered by a driver inattention monitor”, IEEE International Conference on Systems Man and Cybernetics (SMC), Oct 2010, pp. 289-297.

[7] Benalie, N., et al., “Improvement of adaptive cruise control system based on speed characteristics and time headway”, IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2009, pp. 2403-2408.

[8] Birdsall, M., “Google and ITE: The Road Ahead for Self-Driving Cars”, Institute of Transportation Engineers. ITE Journal, 84(5), May 2014, pp. 36-39.

[9] European Commission; Directorate-General for Mobility & Transport: DaCoTA, 2012. “Traffic Safety Basic Facts 2012: Junctions”, pp. 1–20.

[10] European Commision; European Road Safety Observatory (ERSO), “Traffic Safety Basic Facts 2015: Junctions”, 2015.

[11] Transportstyrelsen; Sveriges officiella statistik, “Vägtrafikskador 2014”, 2015.

[12] Press release from Toshiba's webpage, "Toshiba's Image Recognition Processor Powers DENSO's Automotive Front-Camera-Based Active Safety System", 2015. https://www.toshiba.co.jp/about/press/2015_10/pr0601.htm [2015-10-29].

[13] Subaru Nordic's webpage, "Avancerad säkerhetsutrustning", 2015. http://www.subaru.se/modeller/levorg/active-safety/ [2015-10-29].

[14] ConceptCarz, "City safety by Volvo Cars - Outstanding crash prevention that is standard in the all-new XC90", 2015. http://www.conceptcarz.com/a9807/CITY-SAFETY-BY-VOLVO-CARS-%E2%80%93-OUTSTANDING-CRASH-PREVENTION-THAT-IS-STANDARD-IN-THE-ALL-NEW-XC90.aspx [2015-10-29].

[15] Schubert, R., Richter, E. & Wanielik, G., 2008. "Comparison and evaluation of advanced motion models for vehicle tracking." Information Fusion, 2008, pp. 730–735.
Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4632283 [2015-03-18].

[16] TASS International, "PreScan", 2015. https://www.tassinternational.com/prescan [2015-07-30]

[17] Runarsson F, A. H., Granum,F., 2014. "Collision Avoidance at intersections: A probabilistic threat-assessment and decision-making system for safety intervention".

[18] Simon, D. & Chia, T.L.I., 2002. "Kalman filtering with state equality constraints." IEEE Transactions on Aerospace and Electronic Systems, pp. 128–136.

[19] Popescu, V., Bace, M. & Nedevschi, S., 2011. "Lane identification and ego-vehicle accurate global positioning in intersections", IEEE Intelligent Vehicles Symposium, Proceedings, pp. 870–875.

[20] Ardeshiri, T. et al., 2006. "Offset Eliminative Map Matching Algorithm for Intersection Active Safety Applications." 2006 IEEE Intelligent Vehicles Symposium, pp. 82–88.

[21] Jiang, L. et al., 2012. Location-based cooperative vehicle collision avoidance for unsignalized intersections: A multi-sensor integration approach. 2012 International Conference on Connected Vehicles and Expo, ICCVE, pp. 246–251.

[22] Petrich, D. et al., 2013. "Collision Risk Prediction and Warning at Road Intersections Using an Object Oriented Bayesian Network", pp. 270–277.

[23] Lefèvre, S., Laugier, C. & Ibañez-Guzmán, J., 2011. "Exploiting map information for driver intention estimation at road intersections", IEEE Intelligent Vehicles Symposium, Proceedings, pp. 583–588.

[24] Petrich, D. et al., 2014. “Assessing Map-Based Maneuver Hypotheses using Probabilistic Methods and Evidence Theory.” 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), pp. 995–1002.

[25] Dyckmanns, H. et al., 2011. “Object tracking in urban intersections based on active use of a priori knowledge: Active interacting multi model filter”, IEEE Intelligent Vehicles Symposium, Proceedings, (Iv), pp. 625–630.

[26] Schubert, R. & Wanielik, G., 2009. “Unifying Bayesian networks and IMM filtering for improved multiple model estimation.” 2009 12th International Conference on Information Fusion, pp. 810–817.

[27] Naidu, VPS et al., 2007. “Interacting Multiple Model Extended Kalman Filter for Tracking Target Executing Evasive Maneuvers.” International Radar Symposium of India, Dec. 2007, Bangalore.

[28] Jensen, F.V. & Nielsen, T.D., 2007, “Bayesian networks and decision graphs”, Springer, New York.

[29] Bar-Shalom, Y., Li, X.-. & Kirubarajan, T. 2001, “Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software”, Interscience, US.

[30] Särkkä, S., “Bayesian filtering and smoothing.” Cambridge University Press, 2013.

[31] Ferguson, D. et al., 2008. “Detection, prediction, and avoidance of dynamic obstacles in urban environments”, IEEE Intelligent Vehicles Symposium, Proceedings, pp. 1149–1154.

[32] Drezner, Z., “Computation of the Bivariate Normal Integral.” Mathematics of Computation, 32(141), Jan 1978, pp. 277-279.

[33] Mukai, M. et al., “An automotive collision avoidance control based on a feasible set”, Iccas-Sice, 2009, pp. 2164–2168.

[34] Tomas-Gabarron, J.B., Egea-Lopez, E., Garcia-Haro, J., “Vehicular trajectory optimization for cooperative collision avoidance at high speeds”, IEEE Transactions on Intelligent Transportation Systems, 2013, pp. 1930–1941.

[35] Brännström, M., Coelingh, E. & Sjöberg, J., 2010. “Model-based threat assessment for avoiding arbitrary vehicle collisions”, IEEE Transactions on Intelligent Transportation Systems, pp. 658–669.

[36] Kaempchen, N., Schiele, B. & Dietmayer, K., 2009. “Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios”, IEEE Transactions on Intelligent Transportation Systems, pp. 678–687.

[37] Hillenbrand, J. & Kroschel, K., 2006. “A study on the performance of uncooperative collision mitigation systems at intersection-like traffic situations”, 2006 IEEE Conference on Cybernetics and Intelligent Systems.

[38] Campos, G.R, Falcone, P., Sjöberg, J., “Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing” Intelligent Transportation Systems (ITSC), 17th International IEEE Conference, Oct 2014.

[39] Althoff, M. & Dolan, J.M., “Online verification of automated road vehicles using reachability analysis”, IEEE Transactions on Robotics, 2014, pp. 903–918.

[40] Borelli, F., Bemporad, A., and Morari, M., “Predictive Control for linear and hybrid systems”, Jun 2015.

[41] Herceg, M., et al., “Multi-Parametric Toolbox 3.0”, Proc. of the European Control Conference, Zürich, Switzerland, July 2013, pp. 502-510
Available at: http://people.ee.ethz.ch/~mpt/3/ [2015-06-26].

[42] Advanced Forward-Looking Safety Systems – Working Group 3, “vFSS-WG3 Meeting”, Sindelfingen, May 5th, 2012.

[43] dSPACE Gmbh, “AutoBox”, 2015. https://www.dspace.com/en/pub/home/products/hw/accessories/autobox.cfm [2015-10-17].

A

System parameters

The system parameters used throughout the proposed approach is in this chapter defined and presented.

A.1 Time parameters

There are four parameters connected to time in the proposed algorithm. One denoted as the time step ΔT describing the overall collision avoidance system's sample time, which was set to 0.1 seconds. The sample time were chosen to be fast enough to account for possible important changes, but was however restricted by the IMM's sample frequency preferences. If the sample time would have been set as fast as the sensors allowed it to be, the IMM would struggle to differentiate between each consecutive time step's state information. A too fast sample time would hence lead the IMM to the belief that the vehicles were standing still from one time step to another.

A prediction horizon $T_{predictions}$ of 1.5 seconds, equal to 15 predictions ($n_{predictions}$ = 15), were chosen. A longer prediction horizon is not needed since at a velocity of around 50 km/h, as studied in this thesis, a full stop using an Autonomous Emergency Braking (AEB) system could be obtained within around 1.1 seconds according to [42]. A longer prediction horizon will also be more sensitive to false predictions, hence consequently could lead to more false interventions. Depending on how uncertain the measurements would be, a longer prediction time would not necessarily generate any additional information.

In the decision process of knowing when to brake, a reaction time, $t_{reaction}$, need to be defined. The reaction time will thus also include the time it takes for the braking system to be prepared for emergency braking. This time parameter was set to have a duration of two time steps i.e. 0.2 seconds. This was based on the evaluation of [17] since the same type of vehicles were used. Through this evaluation, the authors found that the ramp-up time were varying between 0.2 and 0.3 seconds.

In summary, the time parameters used throughout the CA system were set as:

$$\begin{aligned}
\Delta T &= & & & 0.1s \\
n_{predictions} &= & & & 15 \\
T_{predictions} &= & \Delta T \cdot n_{predictions} &= & 1.5s \\
t_{reaction} &= & 2\Delta T &= & 0.2s
\end{aligned}$$

A.2 Thresholds

There are two thresholds in the system, one for the collision detection subsystem, T_{CD}, and one for the collision avoidance subsystem, T_{CA}. The thresholds can be changed to regulate the behaviour of the system, where an increase of the thresholds would lead to a system having fewer false interventions, but with a threshold set too high could lead to a non-intervening system. If the thresholds instead would be set too low, an excessive degree of interventions would occur hence giving more false interventions.

A threshold of 50 percent was set for the collision detection since this would be the probability of collision if two vehicles were to be next to each other, with the variance being low. The threshold for the collision avoidance was also set to 50 percent since if a threat was determined by the collision detection subsystem, it is equally likely of being an imminent collision. The parameters were therefore set as:

$$T_{CD} = 50\%$$
$$T_{CA} = 50\%$$

As explained later in this chapter, the collision detection threshold has been seen to have a bigger influence on the behaviour of the system than the collision avoidance threshold.

A.3 Vehicle parameters

The vehicles' maximum velocity, v_{max}, was according to the made restrictions set to 13.9 m/s (equal to 50 km/h) where as the maximum deceleration, $-a_{max}$, was set as -8.5m/s^2 due to the vehicle properties based on [17].

The required time it takes for the vehicle to get to a full stop t_{brake} is defined as, $t_{brake} = (v_{end} - v(t))/(-a_{max})$, where v_{end} will be equal to zero at the end of the brake and $v(t)$ is the velocity in the first prediction in each time step. The vehicle's stopping time, t_{stop} is thereafter calculated as the sum of t_{brake} and $t_{reaction}$.

The knowledge of the width and length of each vehicle is important for the collision detection subsystem since the algorithm is dependent on the areas of both vehicles. The total number of η points describing each vehicle area were chosen as 27 (**3**x**9** = **width** x **length**), since this was identified in the evaluation as sufficient to give an accurate result. The evaluation of number points to describe a vehicle's area is later presented in Section 6.3.4.

A.4 Uncertainty- and disturbance parameters

To allow the IMM-filter to be able to account for all possible initial paths of each vehicle, the initial measurement noise covariance matrices need to capture a range of driving configurations that could possibly be executed, as illustrated in Figure A.1.

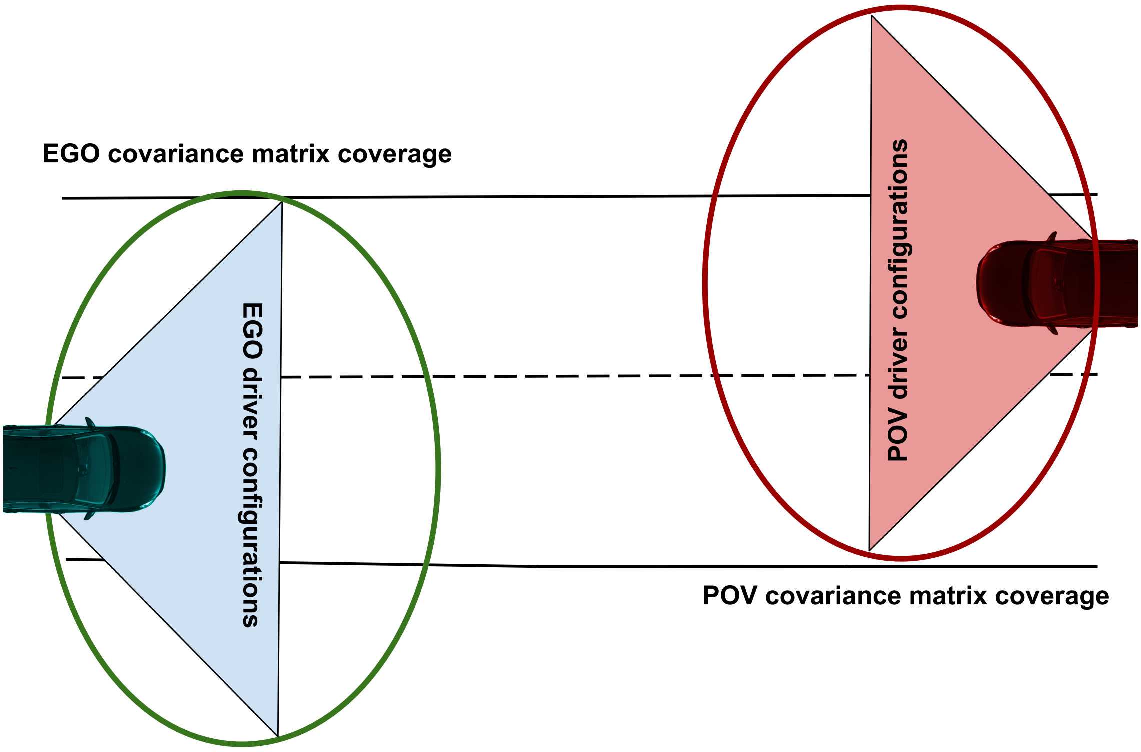

EGO covariance matrix coverage
EGO driver configurations
POV driver configurations
POV covariance matrix coverage

Figure A.1: Illustration of how the initial measurement noise covariance matrices captures a range of driving configurations that could possibly be executed. The covariance matrix covers not only manoeuvre configurations, but also sensor measurement uncertainties hence covering a larger area.

The initial measurement noise covariance matrices for the IMM-filter in Section 3.3 were therefore initialised as:

$$P_{CV}^{init} = diag([10, 10, 10, 10, \frac{50\pi}{180}]) \tag{A.1}$$

$$P_{CT}^{init} = diag([1, 1, 1, 1, \frac{5\pi}{180}]) \tag{A.2}$$

$$P_{CVP}^{init} = diag([10, 10, \frac{90\pi}{180}, 10, \frac{90\pi}{180}]) \tag{A.3}$$

$$P_{CTP}^{init} = diag(1, 1, \frac{5\pi}{180}, 1, \frac{5\pi}{180}]) \tag{A.4}$$

where *Constant Velocity Polar* (CVP) and *Constant Turn rate Polar* (CTP) interprets as the polar state space representation, and *Constant Velocity* (CV) and *Constant Turn rate* (CT) is interpreted as belonging to the spherical state space representation. Thus could, as previously described in Equation (2.1), the covariance matrices first two indexes denote the x- and y-cartesian coordinates, where as the fifth index denotes the angular velocity. The disparity between the two state representations is the third and fourth indexes, where it represents the heading (θ) and speed (v) for the spherical and velocity in x-direction (V_x) and y-direction (V_y) in the polar representation.

The values in the measurement noise covariance matrix for the straight going models have higher variance for the ability to tune in the initial predictions faster. The IMM-filter will therefore have the ability to avoid a long initial transient at an early stage of the vehicle's movement. The overall position, heading and velocity variance values are moreover kept high to be able to capture every initial configuration.

In comparison to the dynamic measurement noise covariance matrix, which will be updated every time step, the process noise covariance matrix will instead be set as static throughout the IMM computations. The process noise covariance matrix will therefore need to quickly capture divergent driver manoeuvres hence was initiated as:

$$Q_{CV/CT} = diag([0, 0, 2, 2, 0]) \tag{A.5}$$

$$Q_{CVP/CTP} = diag([0, 0, 0, 2, 0]) \tag{A.6}$$

This means that the process noise is only applied to the velocities included in the different motion models, hence being able to rapidly regulate on prompt velocity changes.

The process noise covariance matrices for the UKF however will not be set to capture as widely divergent changes, due to the predictions being calculated on already filtered states. These covariance matrices were therefore initiated as:

$$Q_{pred,CV} = diag([0, 0, 0.35, 0.35, \frac{0.5\pi}{180}]) \tag{A.7}$$

$$Q_{pred,CT} = diag([0, 0, 0.35, 0.35, \frac{30\pi}{180}]) \tag{A.8}$$

$$Q_{pred,CVP} = diag([0, 0, \frac{0.5\pi}{180}, 0.35, \frac{0.5\pi}{180}]) \tag{A.9}$$

$$Q_{pred,CTP} = diag([0, 0, \frac{10\pi}{180}, 0.35, \frac{10\pi}{180}]) \tag{A.10}$$

where the values are kept low on the velocities for the straight going models to keep the regulations on the calculated predictions to a minimum. The regulations for the heading and yaw rate for the turn models on the other hand are desired to have rapid regulation properties, hence being kept higher.

B

System performance

Further scenarios for both non-evasive and evasive manoeuvres are presented in this chapter.

B.1 Non-evasive manoeuvres

B.1.1 Avoidance manoeuvre:

Test scenario: Both the EGO and POV are travelling on a straight path, where the POV evades an obstacle in its own lane by making a turning manoeuvre into the EGO's lane, later on returning to the straight path in its own lane. An illustration can be seen in Figure B.1.

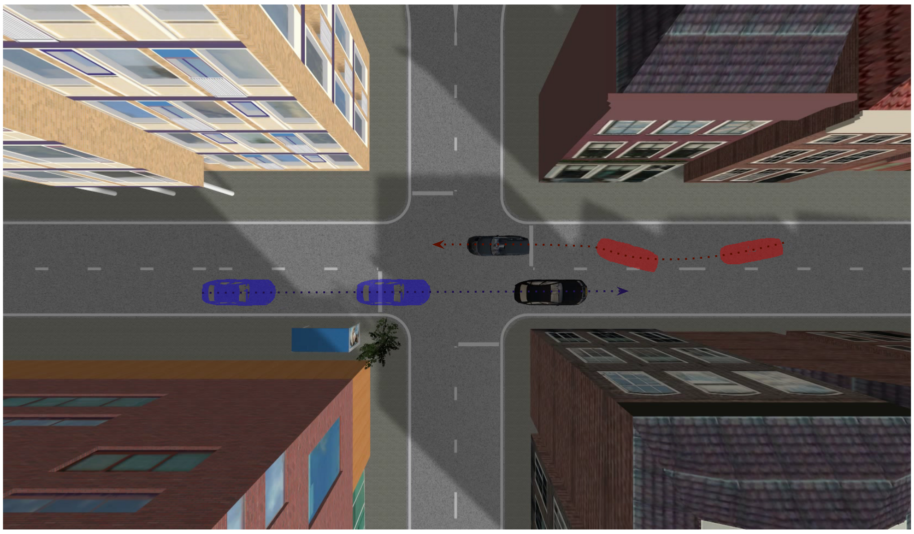
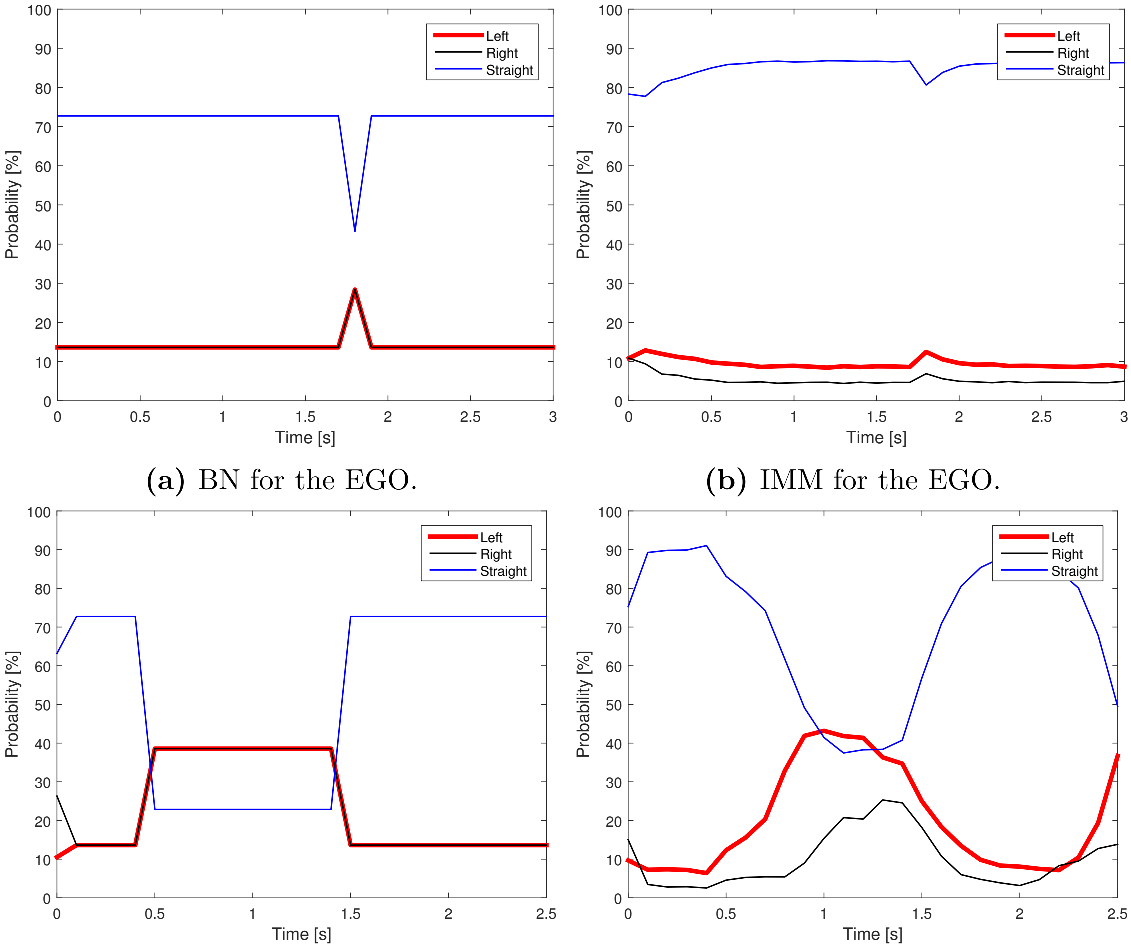


Figure B.1: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.

In Figure B.2 is the BN and IMM for both the EGO and POV illustrated, where the path with the highest probability for the EGO can be seen to be straight, Figure B.2a and B.2b, which corresponds to reality. Furthermore, in Figure B.2c and B.2d is the highest probability in the BN and IMM for the POV also representative of the reality since the POV can be seen to be turning.



(a) BN for the EGO.

(b) IMM for the EGO.

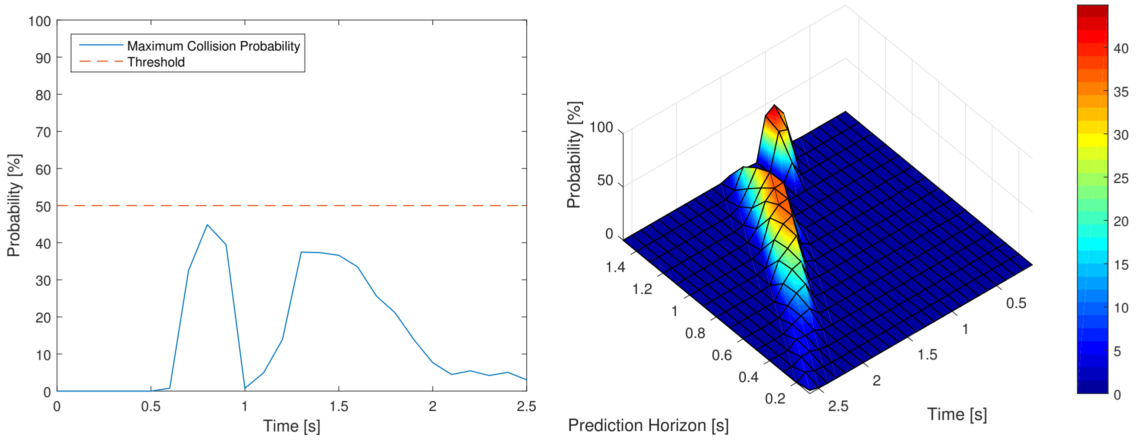
(c) BN for the POV.

(d) IMM for the POV.

Figure B.2: The BN probabilities as well as the IMM weights for each of the three models are here shown for both the EGO and the POV. The three different models are represented by red for left turn, black for right turn and blue for going straight.

Note that the highest probability in the BN for the EGO, Figure B.2a, changes around 1.7 seconds but is corrected by the IMM filter, Figure B.2b. This change is due to the information gained by the lane marker sensor, which in the centre of the intersection cannot detect any lane markers. In the BN for the POV, Figure B.2c, is the highest probability instead changing from straight to left model because of the performed avoidance manoeuvre, but similarly in this case it is thus corrected by the IMM filter as seen in Figure B.2d.

Figure B.3 shows that there is never an imminent risk of collision since the probability of collision is always below the threshold, hence neither a *Danger* nor a *Brake* signal is set.



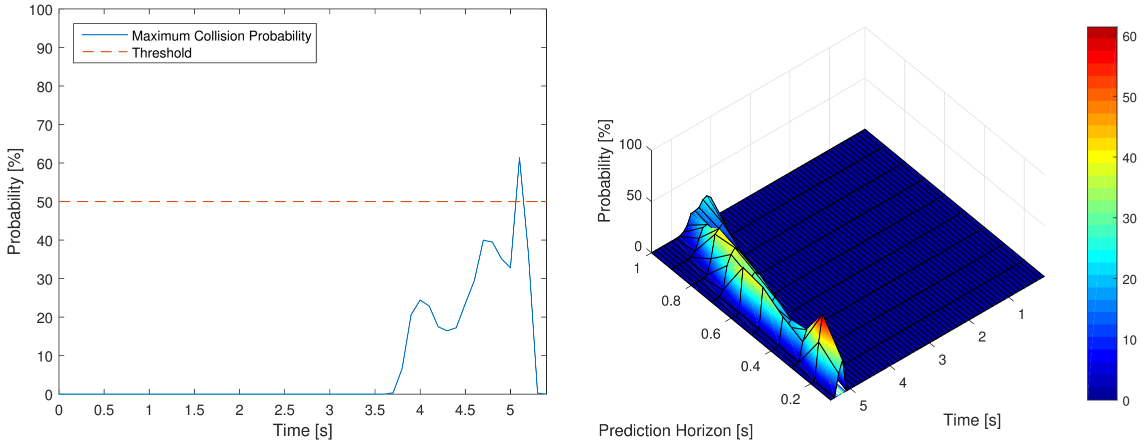
(a) Maximum probability of collision over the prediction horizon, for each time step.

(b) Probability of collision in each prediction step for every time step.

Figure B.3: Probability of collision according to the collision detection subsystem, where Figure B.3a also illustrates the threshold that triggers the *Danger* signal.

B.1.2 Abandoned turn with 10 predictions

If the number of prediction steps would be decreased to 10 instead of 15 for the abandoned turn scenario, the first peak seen in Figure 6.6 and 6.7 is gone, as seen in Figure B.4 and B.5 this since the predictions are getting to far ahead in the future which leads to false collision detection in the system. But since no intervention occurred, it is still possible to use 15 prediction steps.



(a) Maximum probability of collision of the prediction horizon, for each time step. Result from simulation in PreScan.

(b) Probability of collision in each prediction step for every time step. Result from simulation in PreScan.

Figure B.4: Probability of collision according to the collision detection subsystem, where Figure B.4a also illustrates the threshold that triggers the *Danger* signal.

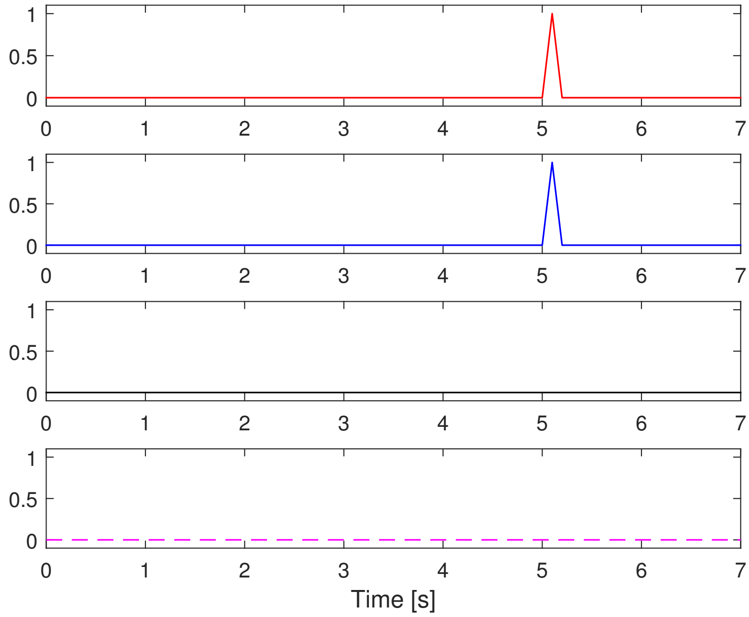


Figure B.5: Illustration of the intervention timing from when a threat is detected to when an braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).

B.2 Evasive manoeuvres

B.2.1 Stationary obstacle:

Test scenario: This scenario has two slightly different versions. In the in-vehicle test the EGO was travelling on a straight path where an obstacle (the POV) obstructed the EGO's path, an illustration can be seen in Figure B.6.

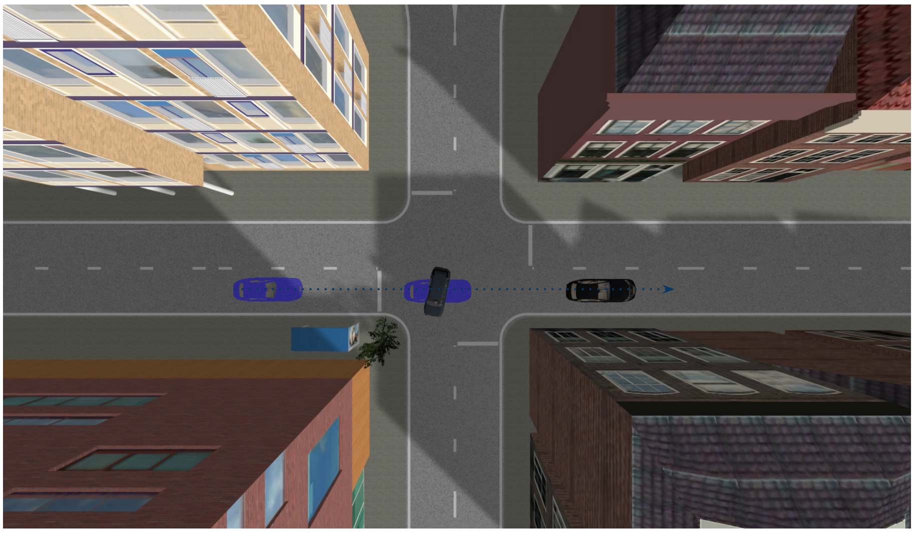
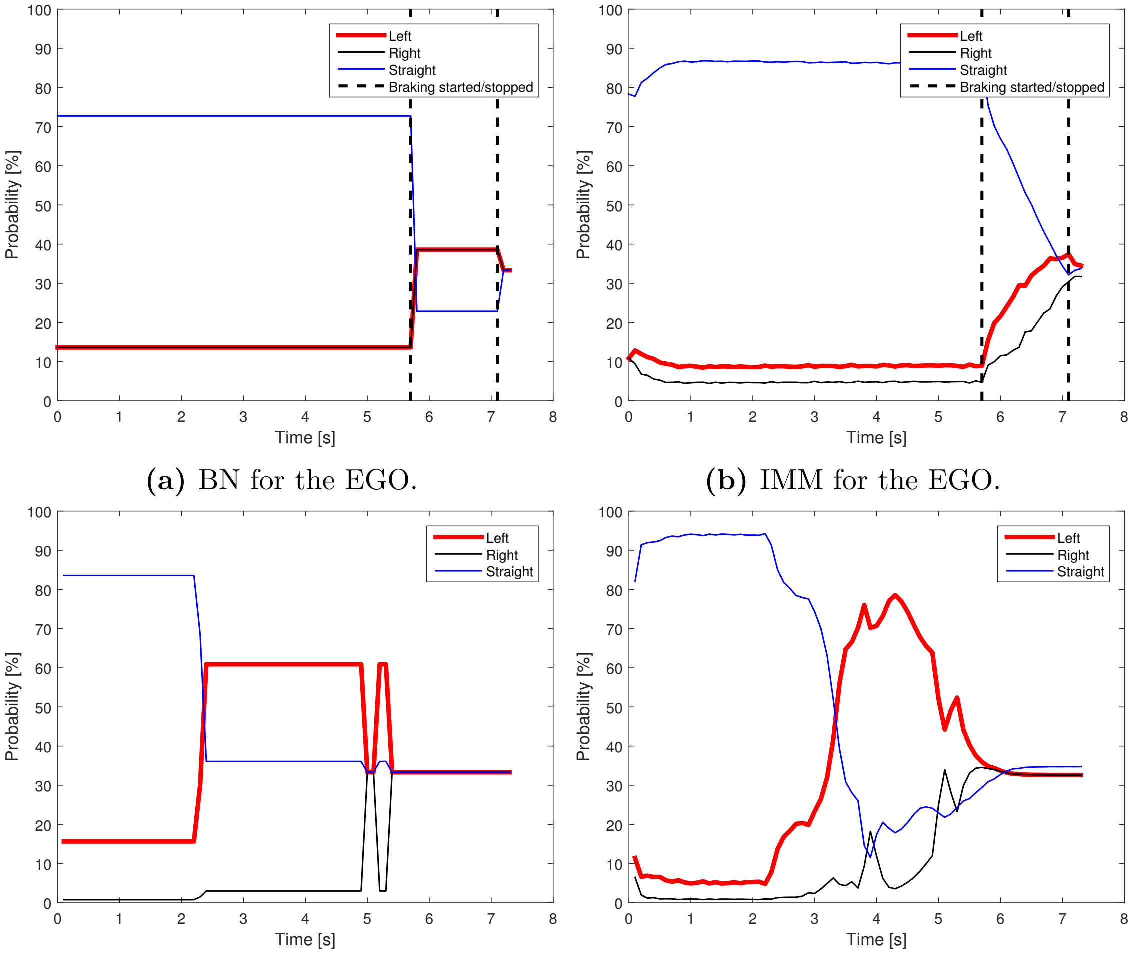


Figure B.6: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas black represents the end positions for both vehicles in the scenario.



(a) BN for the EGO.

(b) IMM for the EGO.

(c) BN for the POV.

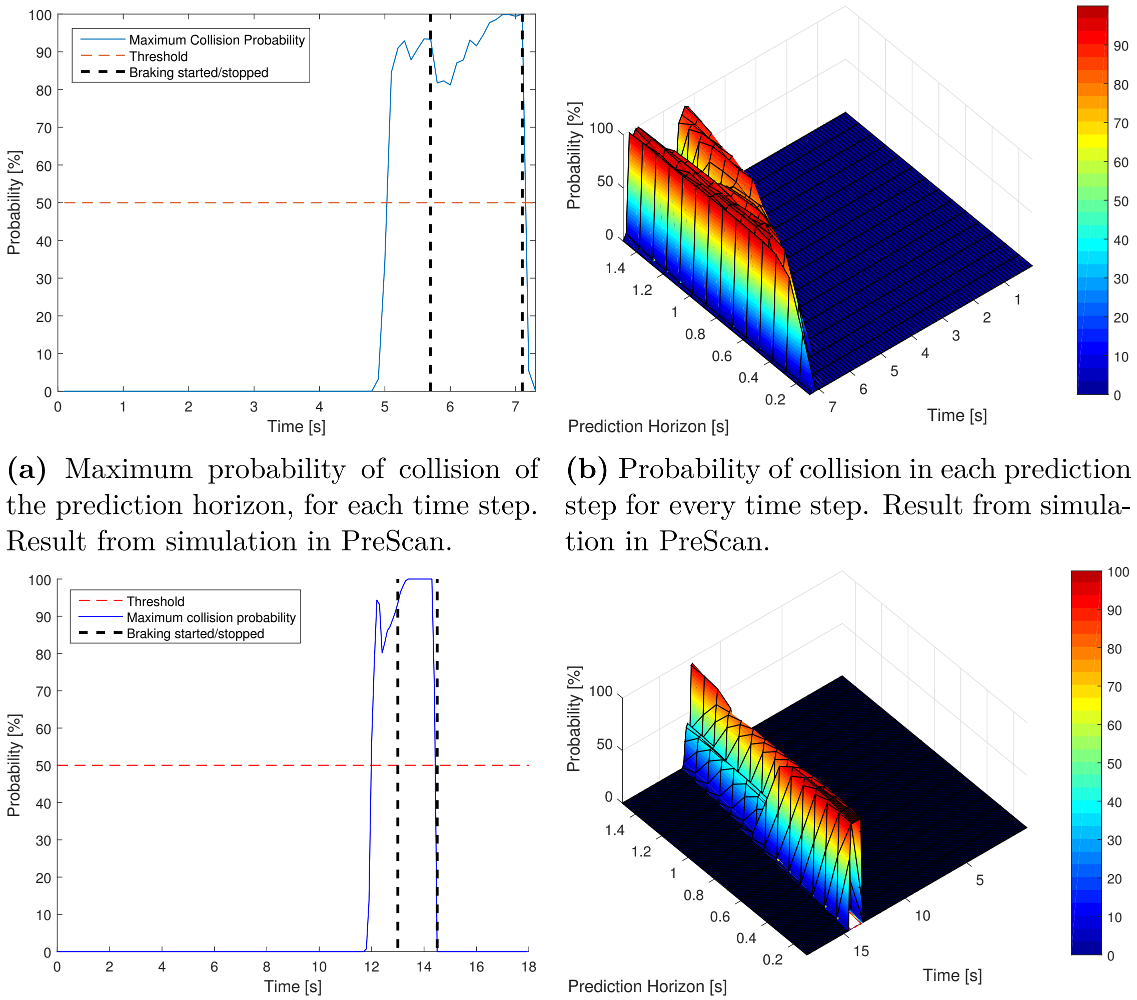
(d) IMM for the POV.

Figure B.7: The BN probabilities as well as the IMM weights for each of the three models in the simulation are here shown for both the EGO and the POV. The three different models are represented by red for left turn, black for right turn and blue for going straight. Additionally is the influence of the braking included for the EGO (black, dotted).

In the simulation in PreScan the POV instead travels straight, turns left and eventually stops in the middle of the EGO's lane, blocking the path for the EGO.

In Figure B.7a and B.7b are the probabilities of the BN and the IMM illustrated for the EGO's three different models (left, right and straight) for the simulations. As desired, the straight model has the highest probability in both the BN and the IMM. In the BN is the probability of turning increasing at around 6 seconds which is explained by looking at Figure B.9, since it is at this time instance the EGO makes a braking intervention. The IMM however corrects this by setting the models to be equally likely.

The probabilities for the POV's three different models according to the BN and IMM, are represented in Figure B.7c and B.7d. The traffic scenario was in this case assumed to be a T-intersection, hence the road network and legal turns were altered as opposed to the default setup. The BN for the POV was influenced by this data as can be seen in Figure B.7c, where the right model has been given a lower probability in the BN. The model with the highest probability in the BN can be seen to change from straight to left when the POV starts to turn. When the POV is standing still on the other hand, the probabilities for the three different models are set equal. The probabilities in the IMM are changing in a similar fashion.



(a) Maximum probability of collision of the prediction horizon, for each time step. Result from simulation in PreScan.

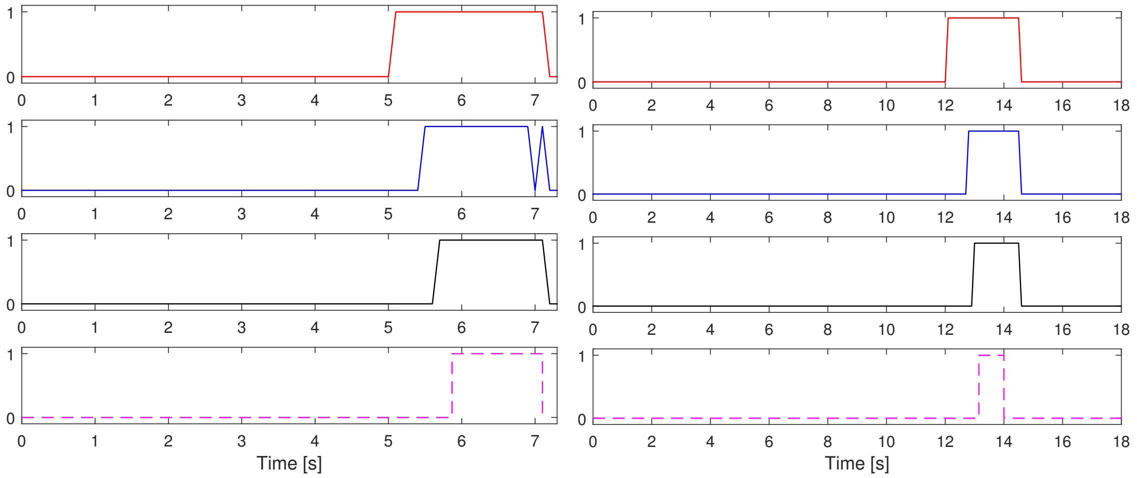
(b) Probability of collision in each prediction step for every time step. Result from simulation in PreScan.

(c) Maximum probability of collision of the prediction horizon, for each time step. Result from in-vehicle tests.

(d) Probability of collision in each prediction step for every time step. Result from in-vehicle tests.

Figure B.8: Probability of collision according to the collision detection subsystem, where Figure B.8a and B.8c also illustrates the threshold that triggers the *Danger* signal.

In Figure B.8 is the probability of collision depicted for both the simulation and in-vehicle test. Since the POV stops at around 5 seconds in the simulation, the probability rises above the threshold and the *Danger* signal is triggered at 5.1 seconds as shown in the top graph of Figure B.9a. According to the second graph in the same figure, the brake pressure is built up in preparation for emergency braking at 5.5 seconds and finally initiates the braking procedure at 5.7 seconds as seen in the third graph. The forth graph illustrates that the optimal braking should be triggered at 5.9 seconds.



(a) Intervention timing, result from simulation in PreScan.

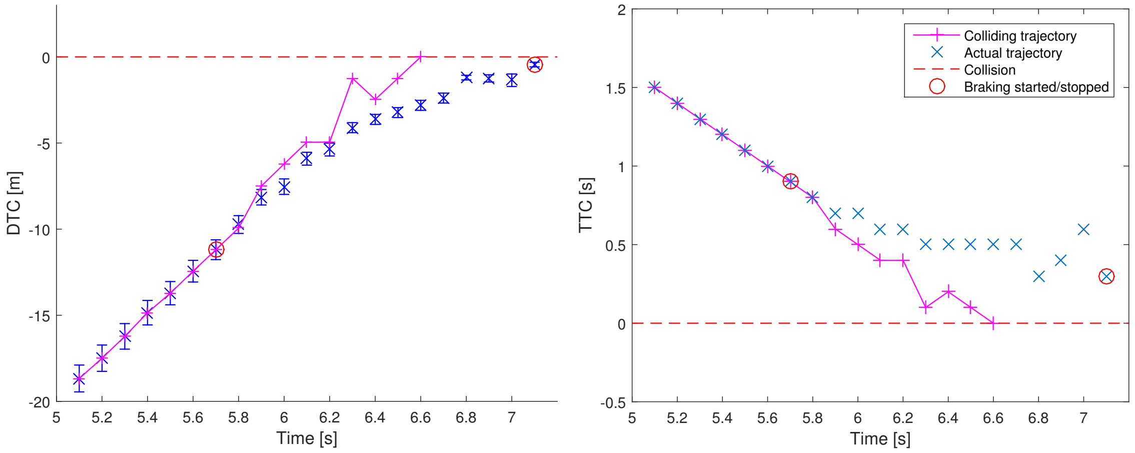
(b) Intervention timing, result from in-vehicle test.

Figure B.9: Illustration of the intervention timing from when a threat is detected to when a braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).

All of the graph's values are set equal to zero at 7.1 seconds since the brakes have been released and the threat has passed.

Figure B.8c, B.8d and B.9b instead illustrates the behaviour with data given from in-vehicle test. It can here be noted that the in-vehicle test was not actually triggering a braking intervention. The main purpose of the figures is therefore to merely demonstrate the systems ability to detect a danger and demanding a braking intervention, hence the danger as well as braking signal's length is not reflecting the reality. Moreover should the *optimal braking* seen in the fourth graph in Figure B.9b, also be considered to not fully reflect the reality.

Figure B.10 shows the DTC and TTC as well as when the braking is initiated and released for the simulations, which can be compared to the third graph (black) in Figure B.9a. Compared to the DTC for the colliding trajectory (where no braking is performed), is the evasive trajectory's DTC decreasing after the brakes have been applied and finally ends up really close to collision, but in a later time step. By instead looking at the TTC it is shown that the difference is bigger between the TTC for the colliding and the actual trajectory. Since the EGO's velocity is decreasing can the collision still be avoided, even though the DTC is close to and the TTC only is slightly above zero.



(a) The evolvement of DTC over time.

(b) The evolvement of TTC over time.

Figure B.10: Here are the DTC and TTC over time seen for the simulation. Blue represents the estimated DTC/TTC in each time step when the EGO executes an intervention. The red circles represent when the brakes are triggered and later released. Magenta represents the estimated DTC/TTC for a vehicle without a CA system. Figure B.10a also illustrates the distance uncertainty with vertical bars. The idea is that the vehicle should never end up in a DTC or TTC equal to 0, since this is when a collision occurs.

B.2.2 Left turn by the POV across the EGO's path

There is an additional turning scenario made by the POV, where the POV makes an aggressive left turn across the EGO's path, an illustration can be seen in Figure B.11.

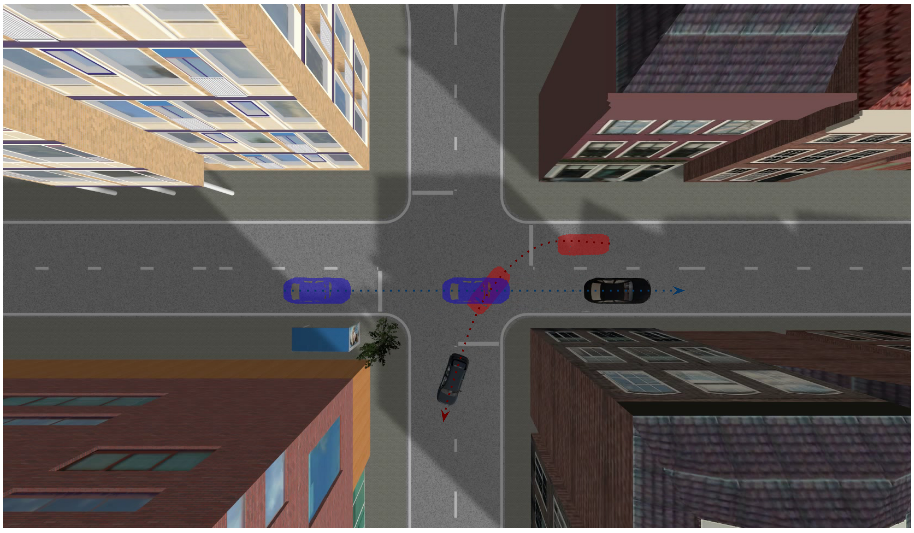
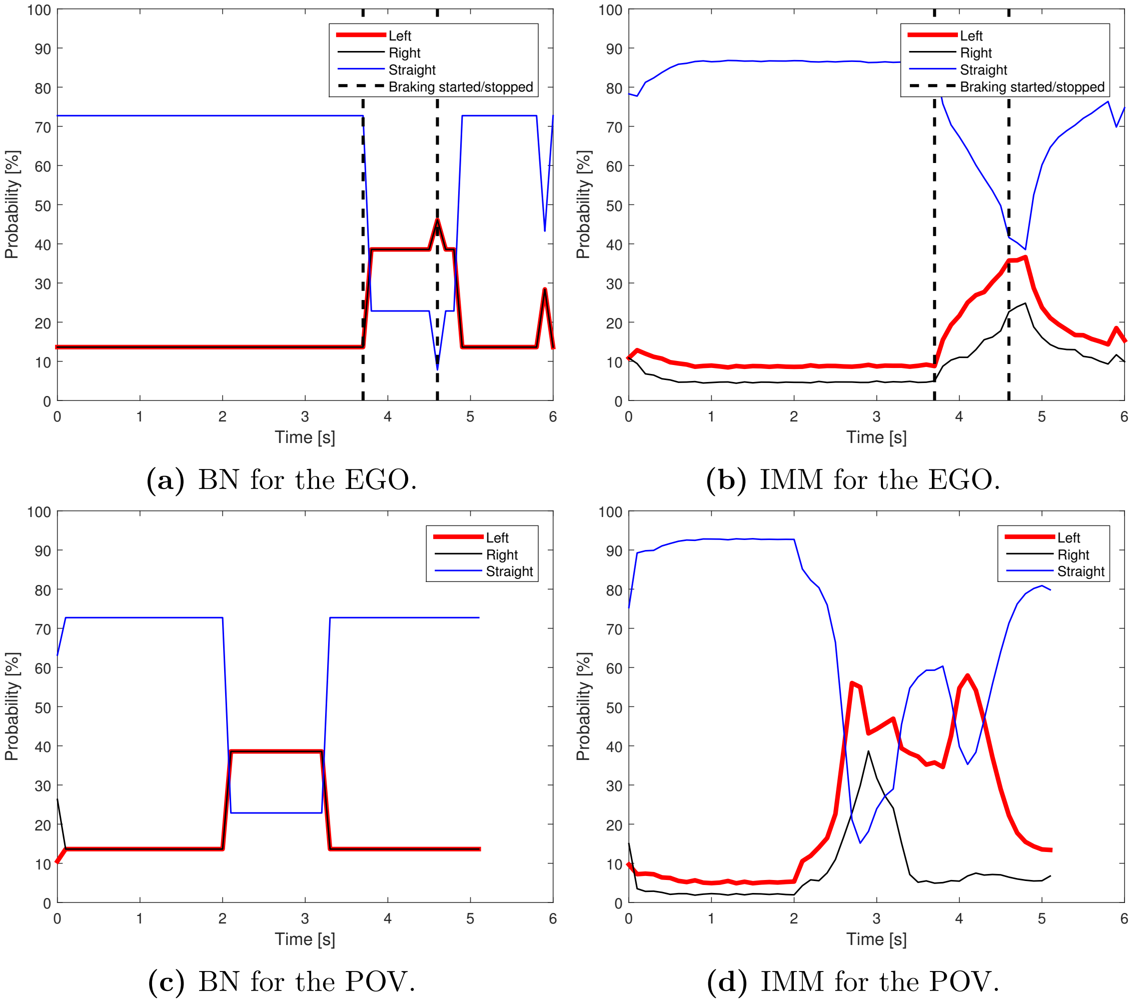


Figure B.11: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.



(a) BN for the EGO.

(b) IMM for the EGO.

(c) BN for the POV.

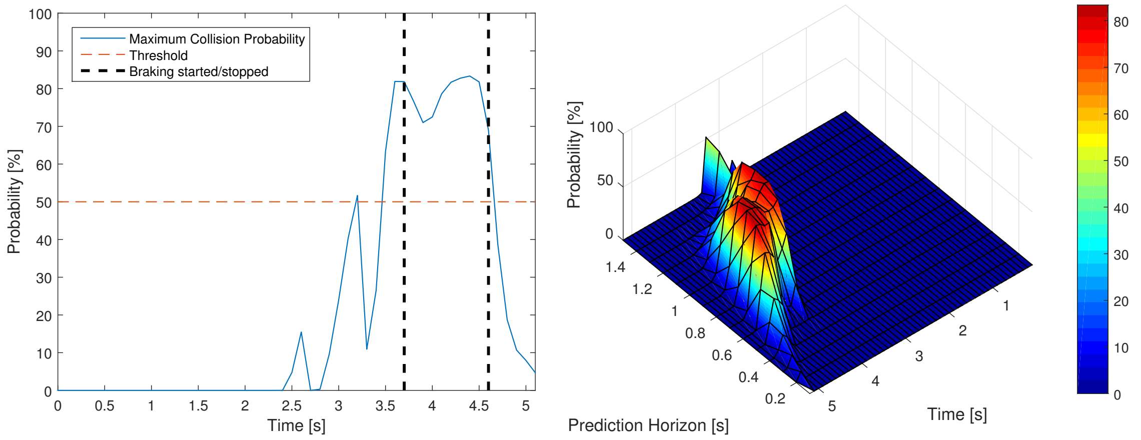
(d) IMM for the POV.

Figure B.12: The BN probabilities as well as the IMM weights for each of the three models in the simulation are here shown for both the EGO and the POV. The three different models are represented by red for left turn, black for right turn and blue for going straight. Additionally is the influence of the braking included for the EGO (black, dotted).

In Figure B.12a and B.12b is the BN and IMM depicted for the EGO. Similar to before the BN is decreasing its probability for the straight model when it brakes, which can be seen to be the reason by looking at Figure B.14. As before is the IMM correcting this.

Since the BN has no evidences which can distinguish between right or left turn for the POV, the BN is only hinting on a change of highest probability between straight and turn model as seen in Figure B.12c. The IMM probabilities for the POV, seen in Figure B.12d, can be seen to handle the change better and can distinguish between right and left.

The probability of collision, Figure B.13, has two peaks above the threshold which triggers the *Danger* signal as seen in Figure B.14. In this case an increase of the threshold would be desirable to prevent from unnecessary interventions. The system is not triggered to brake though until the second peak.



(a) Maximum probability of collision of the prediction horizon, for each time step. Result from simulation in PreScan.

(b) Probability of collision in each prediction step for every time step. Result from simulation in PreScan.

Figure B.13: Probability of collision according to the collision detection subsystem, where Figure B.13a also illustrates the threshold that triggers the *Danger* signal.

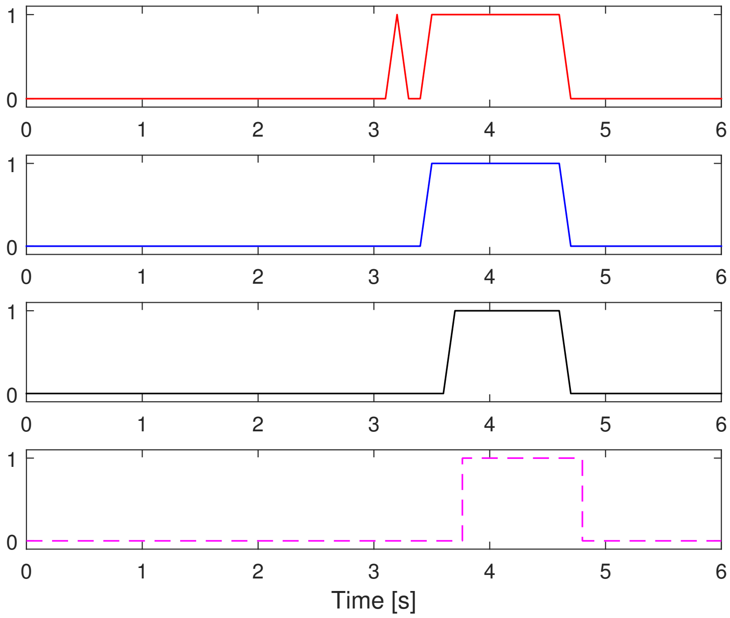
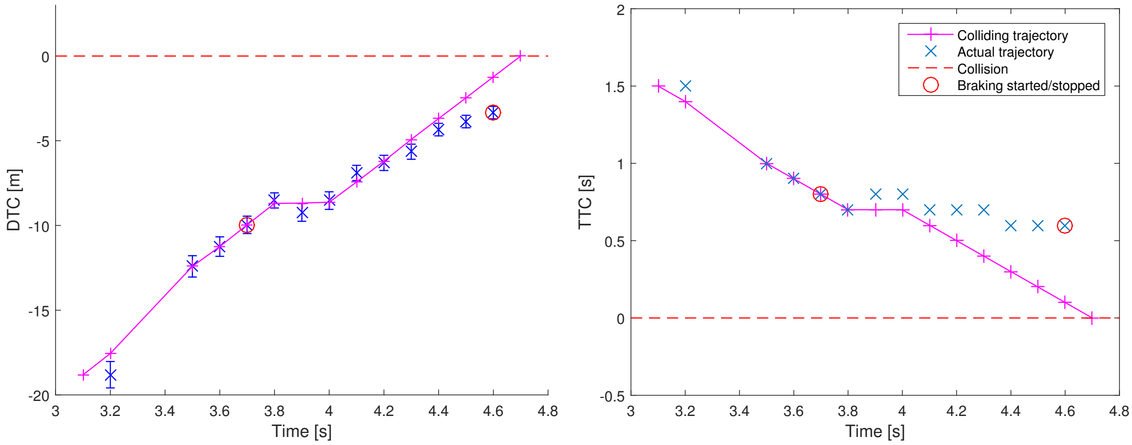


Figure B.14: Illustration of the intervention timing from when a threat is detected to when an braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).

As seen in Figure B.15 is the braking performed until the DTC is at a safe distance and with a TTC of around 0.5 seconds.



(a) The evolvement of DTC over time.

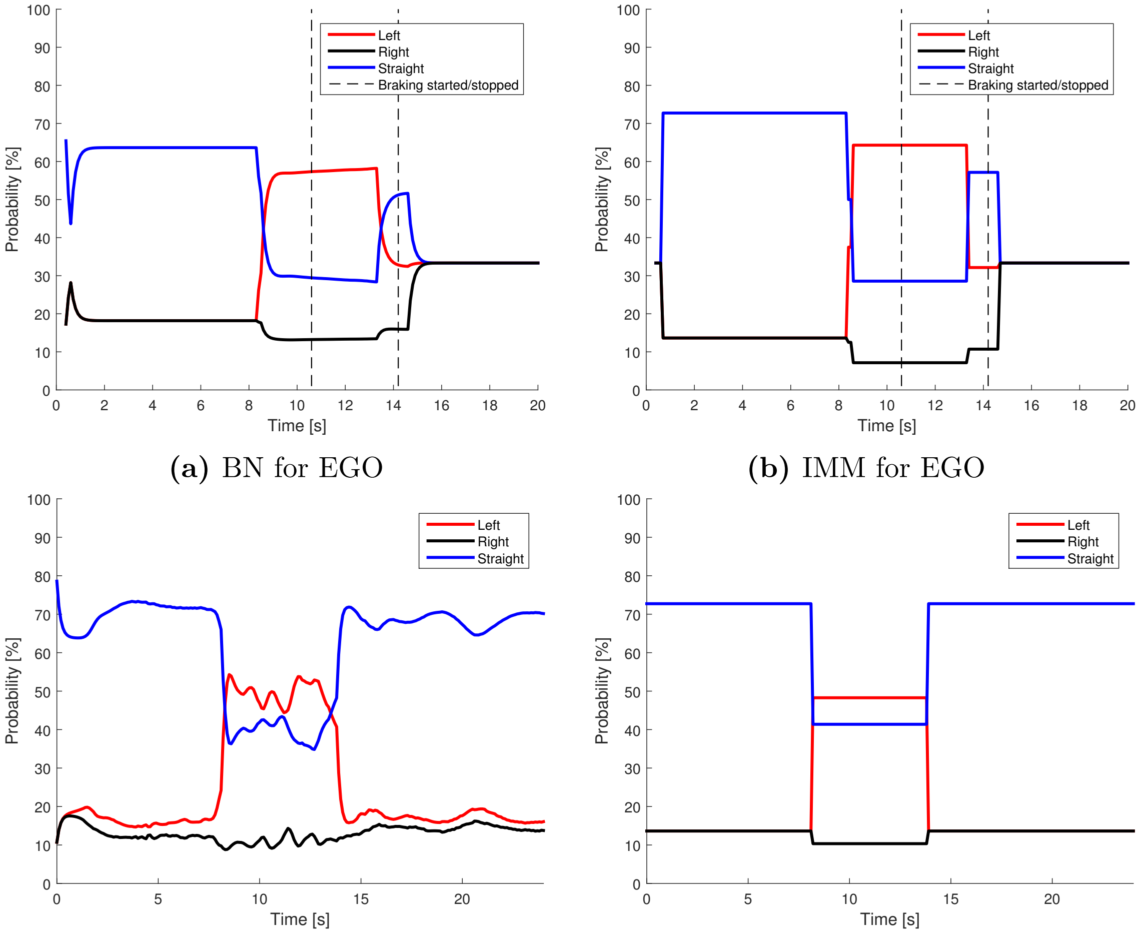
(b) The evolvement of TTC over time.

Figure B.15: Here are the DTC and TTC over time seen for the simulation. Blue represents the estimated DTC/TTC in each time step when the EGO executes an intervention. The red circles represent when the brakes are triggered and later released. Magenta represents the estimated DTC/TTC for a vehicle without a CA system. Figure B.15a also illustrates the distance uncertainty with vertical bars. The idea is that the vehicle should never end up in a DTC or TTC equal to 0, since this is when a collision occurs.

B.2.3 Real tests of left turn across path by either EGO or POV

Real test data of both the EGO and POV performing a left turn across the other vehicles path is illustrated in Figure B.16, B.17 and B.18. As illustrated in Figure B.16, there are several similarities in the result from both the BN and IMM. Here, the result for the EGO, Figure B.16a and B.16b, is more distinct in comparison to the POV result, Figure B.16c and B.16d, but both are nonetheless giving a similar result.

This is also reflected in both the collision probability, Figure B.17 as well as the intervention timing, Figure B.18, where both the collision detection and threat assessment is more narrow when POV performs the turn. Note that the optimal braking profile illustrated in Figure B.18a and B.18b is only accurate for the initiated braking phase.



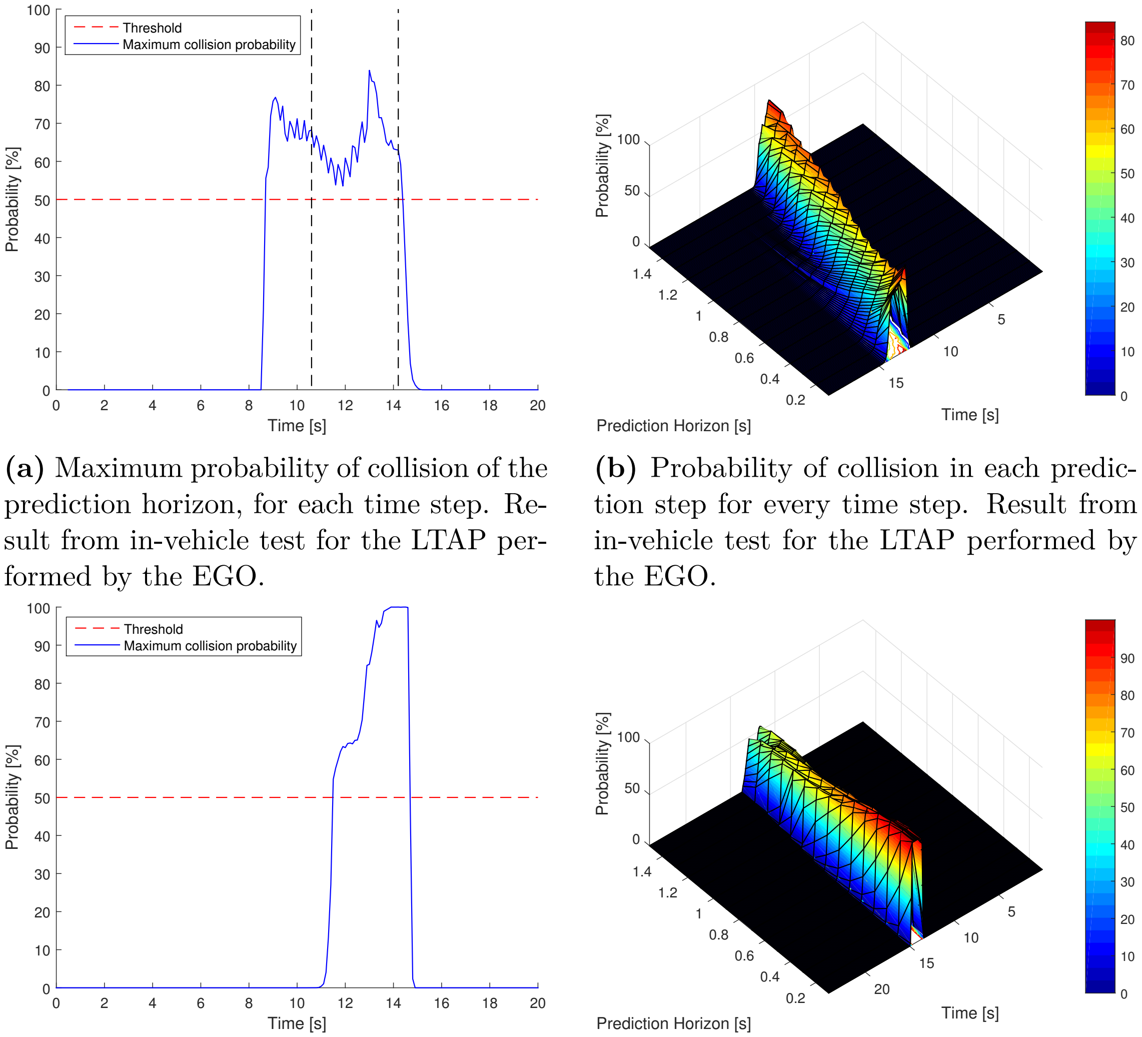
(a) BN for EGO

(b) IMM for EGO

(c) BN for POV

(d) IMM for POV

Figure B.16: In the left figure the BN probabilities is depicted, in the right the IMM weights, both for the EGO's three different models. Here red represents left turn, black represents right turn and blue represents going straight. The top plot demonstrates the result from BN and IMM for the EGO making a left turn across path, where as the bottom plot demonstrates the same scenario for the POV.



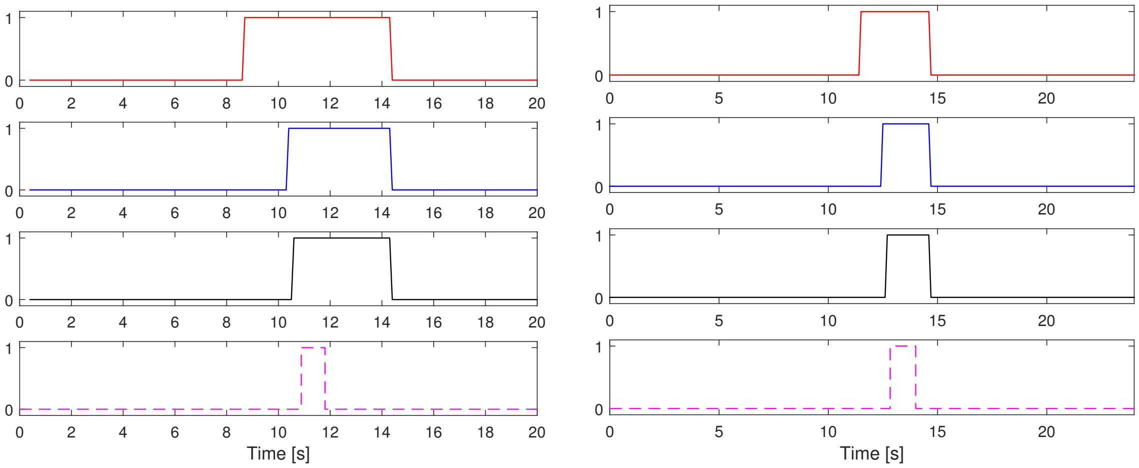
(a) Maximum probability of collision of the prediction horizon, for each time step. Result from in-vehicle test for the LTAP performed by the EGO.

(b) Probability of collision in each prediction step for every time step. Result from in-vehicle test for the LTAP performed by the EGO.

(c) Maximum probability of collision of the prediction horizon, for each time step. Result from in-vehicle test for the LTAP performed by the POV.

(d) Probability of collision in each prediction step for every time step. Result from in-vehicle test for the LTAP performed by the POV.

Figure B.17: Probability of collision according to the collision detection subsystem, where Figure B.17a and B.17c also illustrate the threshold that triggers the *Danger* signal.



(a) Performance when the EGO is making the turn.

(b) Performance when the POV is making the turn.

Figure B.18: Illustration of the intervention timing from when a threat is detected to when an braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).

B.2.4 Attempted turn into the same lane:

Test scenario: The POV makes a left turn across the EGO's lane, while the EGO initiates a right turn into the POV's intended path. The EGO will here have a right blinker signal turned on close to the intersection, whereas the POV has the road restrictions and legal turns set as left and straight, i.e. a T-intersection. An illustration can be seen in Figure B.19.

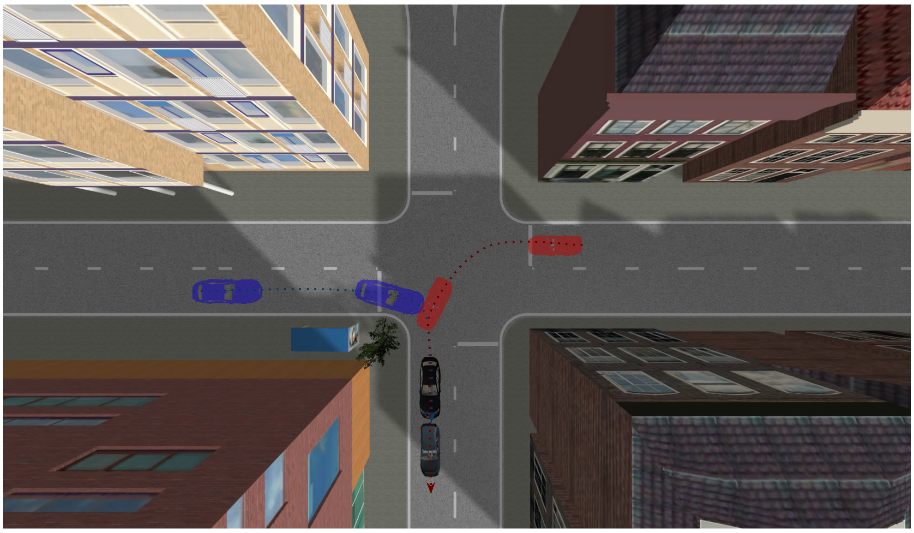
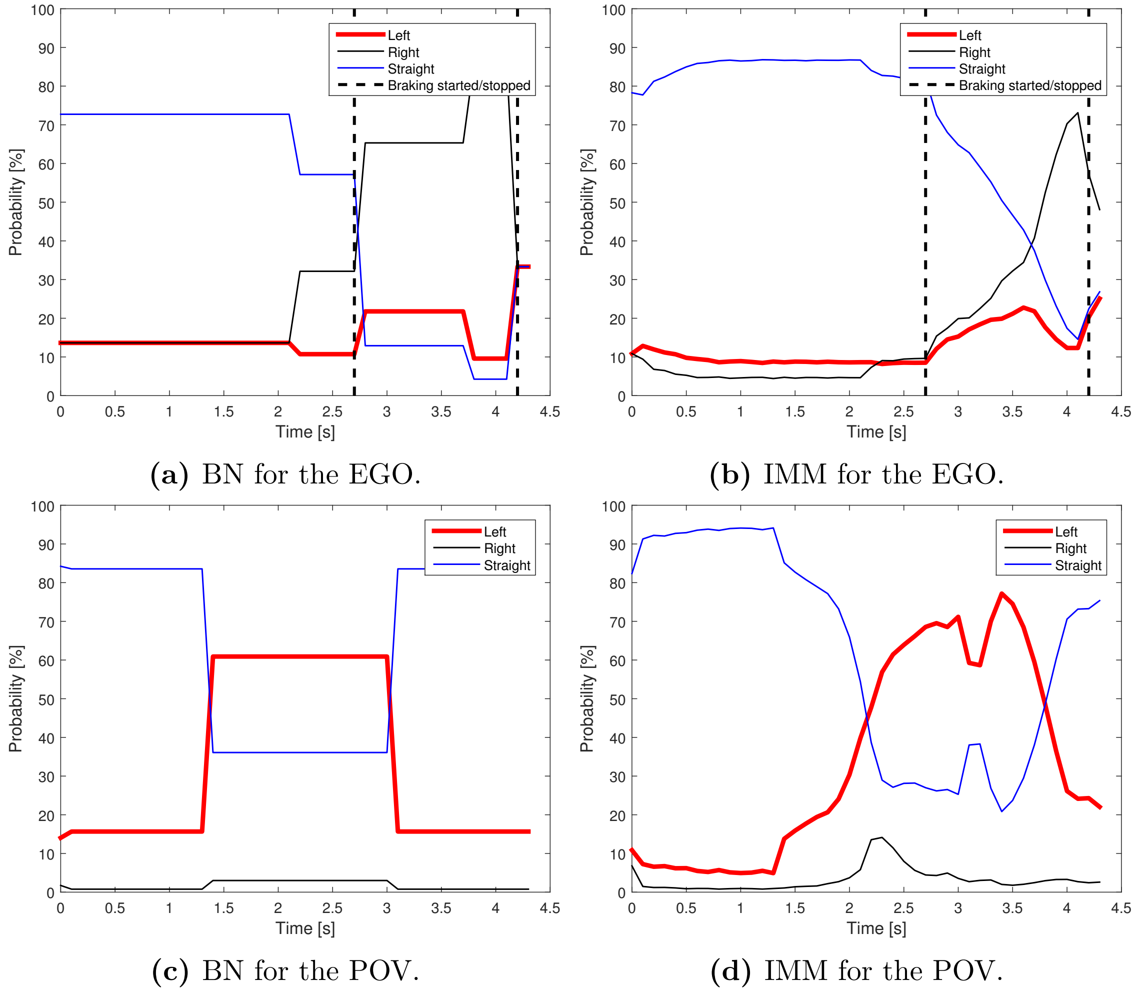


Figure B.19: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.

In Figure B.20a and B.20b is the BN and IMM illustrated for the EGO, where the model having the highest probability can be seen to change from straight to right model for both the BN and IMM, similar to what happens in the scenario. The BN and IMM for the POV can be seen in Figure B.20c and B.20d, where also here the BN and IMM are demonstrating the expected behaviour. In Figure B.23 is the braking phase illustrated and the related DTC and TTC. The TTC in the time step just before the brakes are released is really close to zero, but the TTC in the same time step is safe, whereas a collision never occurs but is very close.

As seen in Figure B.21, the probability of collision is growing more certain with time and the danger and braking signal are initiated at the same time instance, seen in Figure B.22.

There are two additional results showing the behaviour when a turn into the same lane is performed, *EGO turn, POV straight* and *POV turn, EGO straight*. In *EGO turn, POV straight* is the EGO making a right turn into the same lane as the POV is travelling on. Here the EGO's right turn signal is activated close to the intersection, an illustration can be seen in Figure B.24.



(a) BN for the EGO.

(b) IMM for the EGO.

(c) BN for the POV.

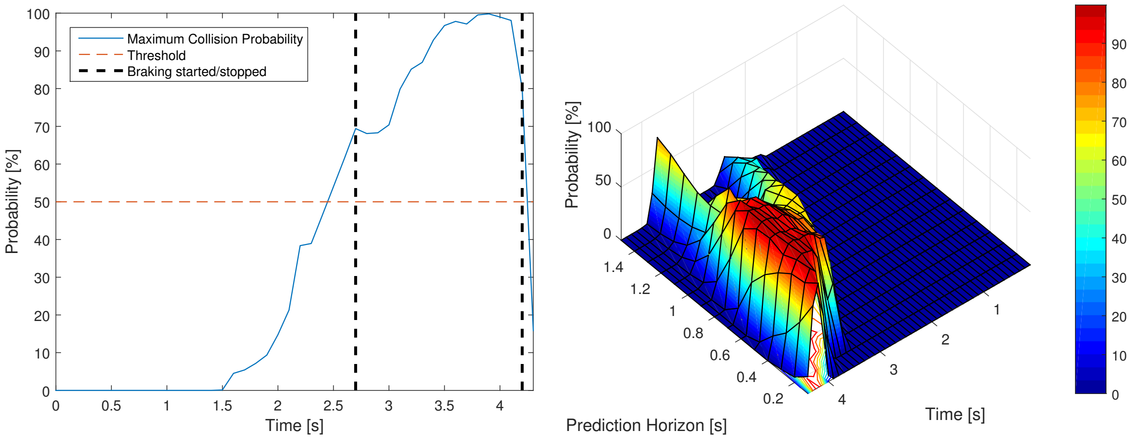
(d) IMM for the POV.

Figure B.20: The BN probabilities as well as the IMM weights for each of the three models are here shown for both the EGO and the POV. The three different models are represented by red for left turn, black for right turn and blue for going straight. Additionally is the influence of the braking included for the EGO (black, dotted).

The BN and IMM for the EGO is depicted in Figure B.25a and B.25b, which behaves as expected where the model with the highest probability changes from straight to right. In Figure B.25c and B.25d can the BN and IMM for the POV be seen. The model of highest probability for the BN is same throughout the whole scenario whereas the IMM has some difficulties in the beginning due to the transient phase of the filter.

The probability of collision can be seen in Figure B.26 and has a long time interval when the *Danger* signal is set high, as seen in Figure B.27. The DTC and TTC as a function of time is illustrated in Figure B.28 where both the TTC and DTC can be seen to get really close to zero. This means that the collision is not fully avoided in this case, but mitigated to a big extent.

In the scenario *POV right, EGO straight* is the POV instead making a right turn into the EGO's straight path, an illustration can be seen in Figure B.29.



(a) Maximum probability of collision of the prediction horizon, for each time step.

(b) Probability of collision in each prediction step for every time step.

Figure B.21: Probability of collision according to the collision detection subsystem, where Figure B.21a also illustrates the threshold that triggers the *Danger* signal.

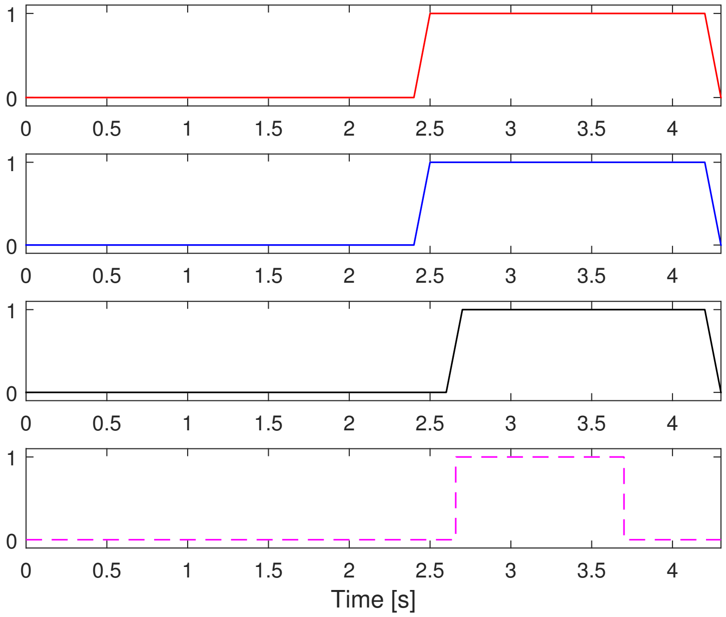
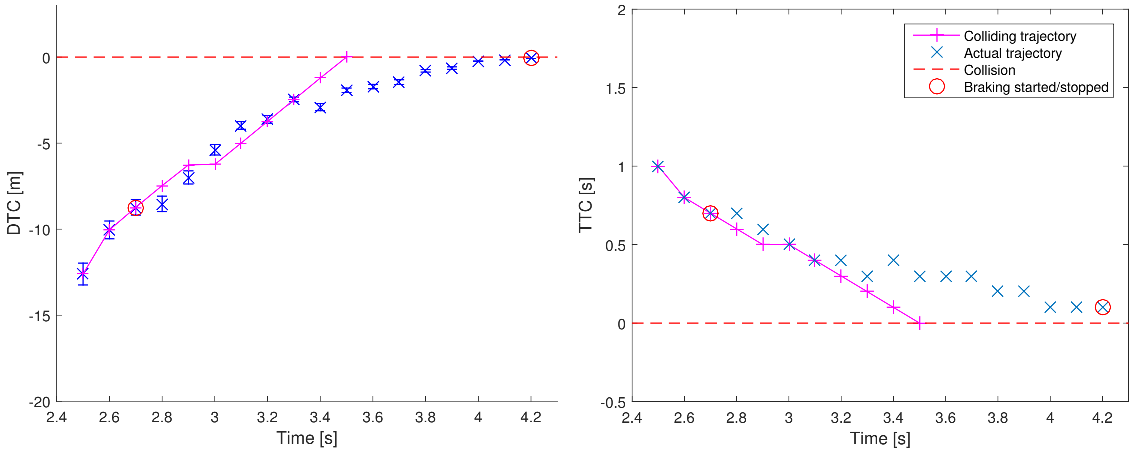


Figure B.22: Illustration of the intervention timing from when a threat is detected to when a braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).

In Figure B.30a and B.30b can the BN and IMM for the EGO be seen, as before is the probability of straight model going down due to the braking in the BN but corrected by the IMM filter. For the POV the BN cannot distinguish between right or left turn since it does not have enough of evidence. The IMM also has some difficulties in the beginning of the change but is certain about the right model after some time steps. The BN and IMM for the POV can be seen in Figure B.30c and B.30d.



(a) The evolvement of DTC over time.

(b) The evolvement of TTC over time.

Figure B.23: Here are the DTC and TTC over time seen for the simulation. Blue represents the estimated DTC/TTC in each time step when the EGO executes an intervention. The red circles represent when the brakes are triggered and later released. Magenta represents the estimated DTC/TTC for a vehicle without a CA system. Figure B.23a also illustrates the distance uncertainty with vertical bars. The idea is that the vehicle should never end up in a DTC or TTC equal to 0, since this is when a collision occurs.

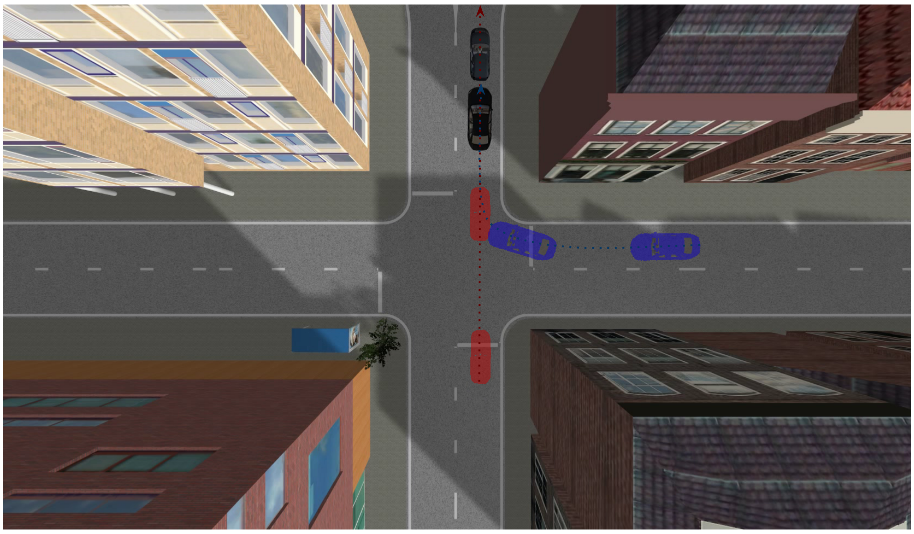
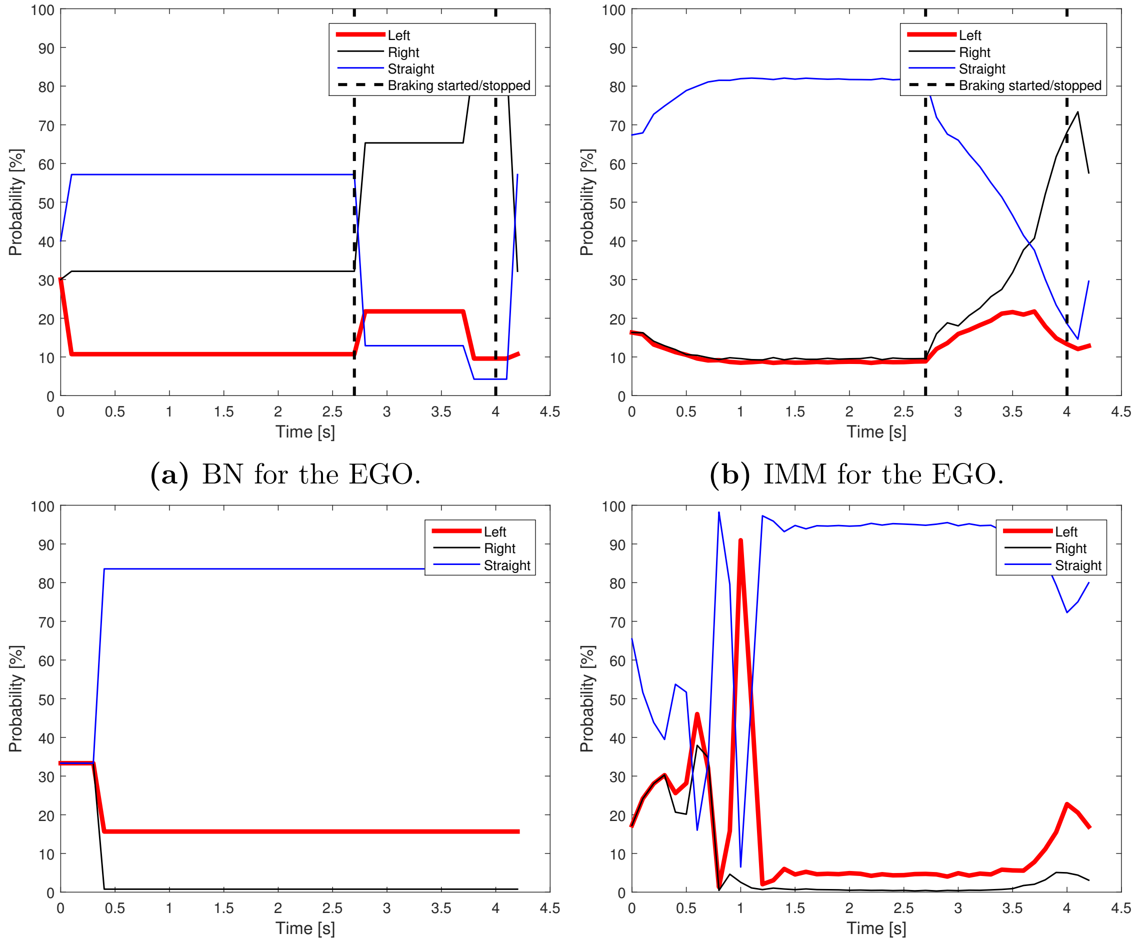


Figure B.24: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.

Since there are difficulties for the IMM to decide which model to choose between around 3 and 3.5 second, also the collision detection has some problems which results in a drop of the probability in this time interval, seen in Figure B.31. In Figure B.32 can the resulting effect be seen, where the braking is performed in two intervals. The TTC and DTC as a function of time, seen in Figure B.33, shows that a collision is close, but a collision never occurs in reality.



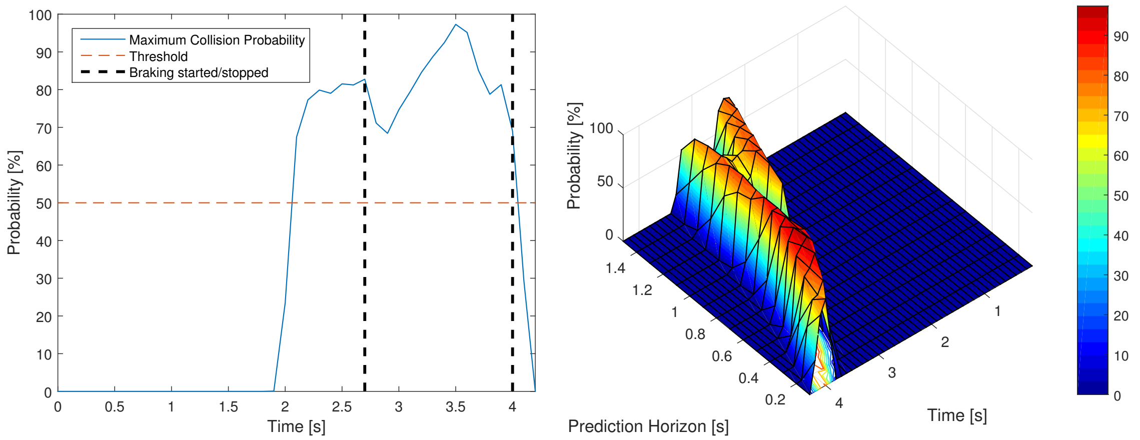
(a) BN for the EGO.

(b) IMM for the EGO.

(c) BN for the POV.

(d) IMM for the POV.

Figure B.25: The BN probabilities as well as the IMM weights for each of the three models in the simulation are here shown for both the EGO and the POV. The three different models are represented by red for left turn, black for right turn and blue for going straight. Additionally is the influence of the braking included for the EGO (black, dotted).



(a) Maximum probability of collision of the prediction horizon, for each time step. Result from simulation in PreScan.

(b) Probability of collision in each prediction step for every time step. Result from simulation in PreScan.

Figure B.26: Probability of collision according to the collision detection subsystem, where Figure B.26a also illustrates the threshold that triggers the *Danger* signal.

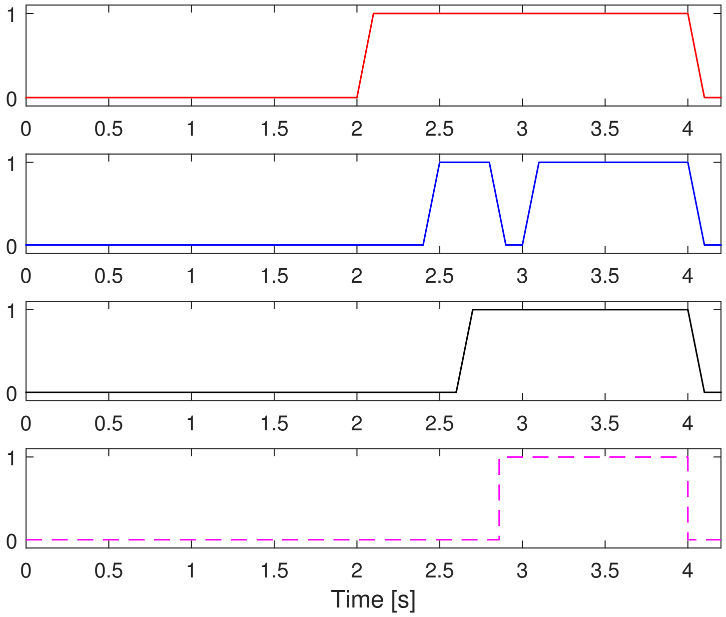
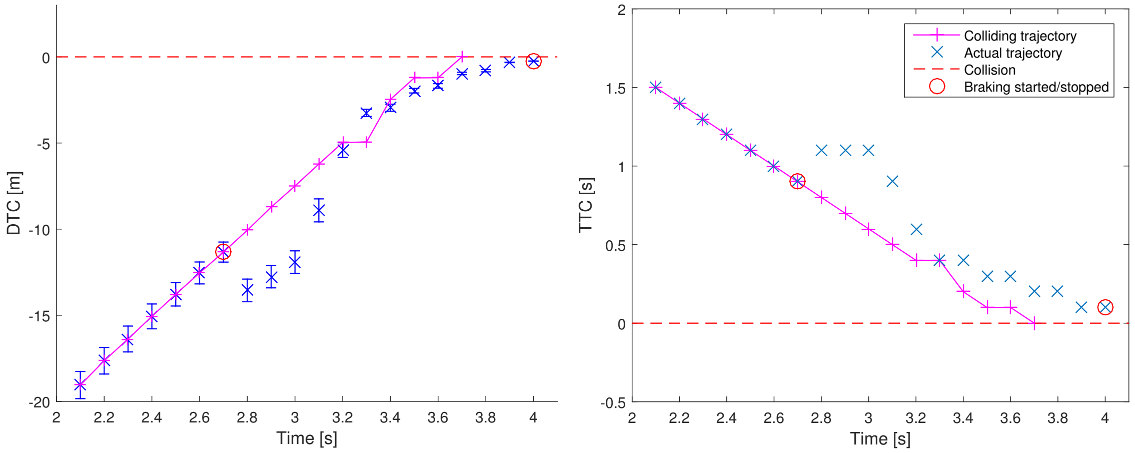


Figure B.27: Illustration of the intervention timing from when a threat is detected to when an braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).



(a) The evolvement of DTC over time.

(b) The evolvement of TTC over time.

Figure B.28: Here are the DTC and TTC over time seen for the simulation. Blue represents the estimated DTC/TTC in each time step when the EGO executes an intervention. The red circles represent when the brakes are triggered and later released. Magenta represents the estimated DTC/TTC for a vehicle without a CA system. Figure B.28a also illustrates the distance uncertainty with vertical bars. The idea is that the vehicle should never end up in a DTC or TTC equal to 0, since this is when a collision occurs.

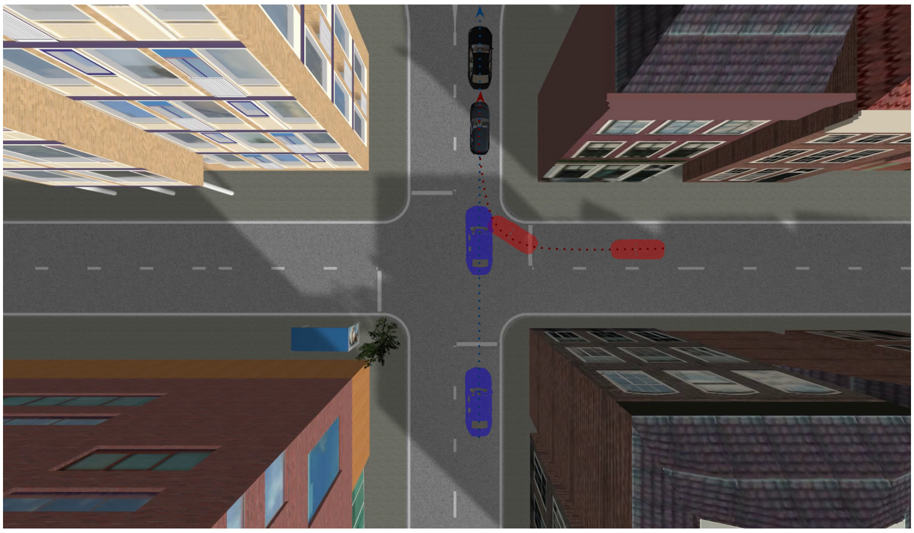
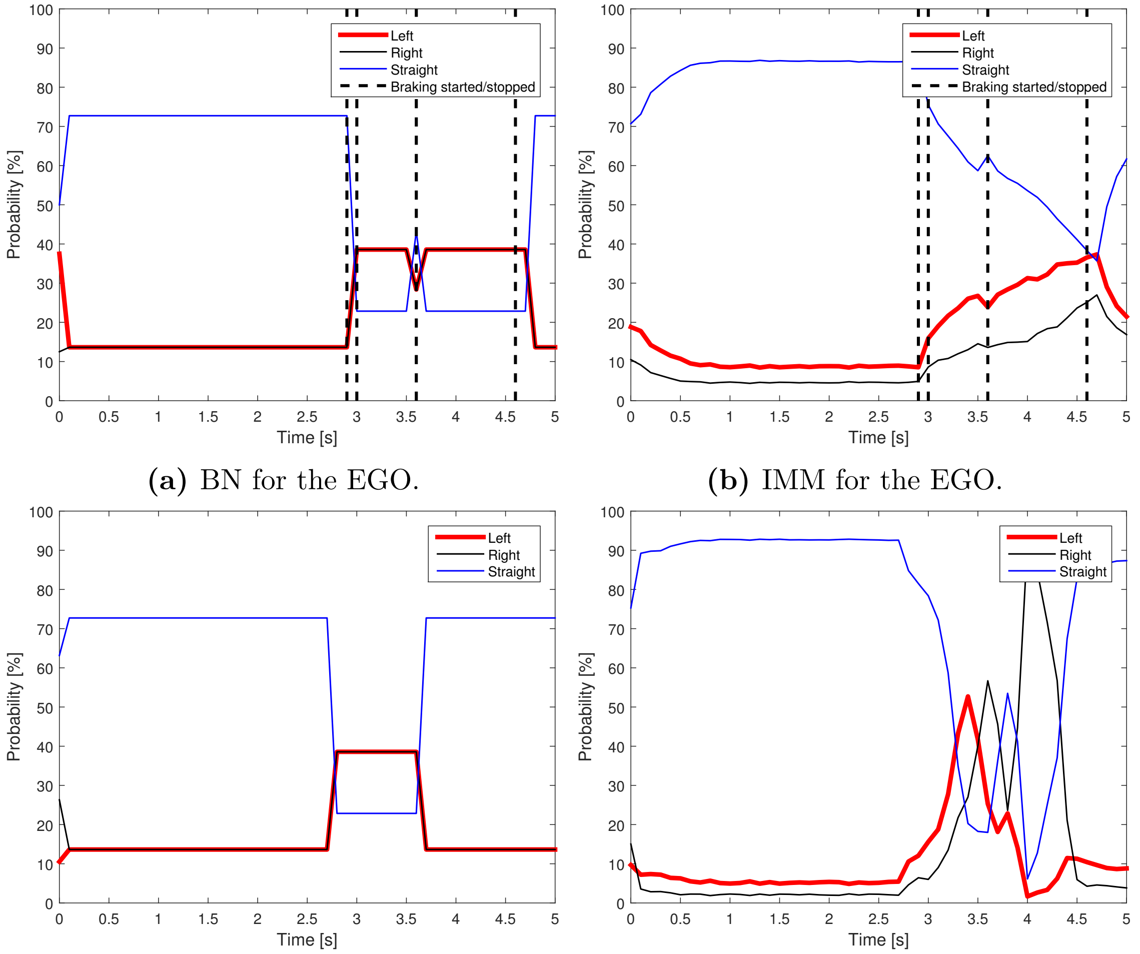


Figure B.29: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.



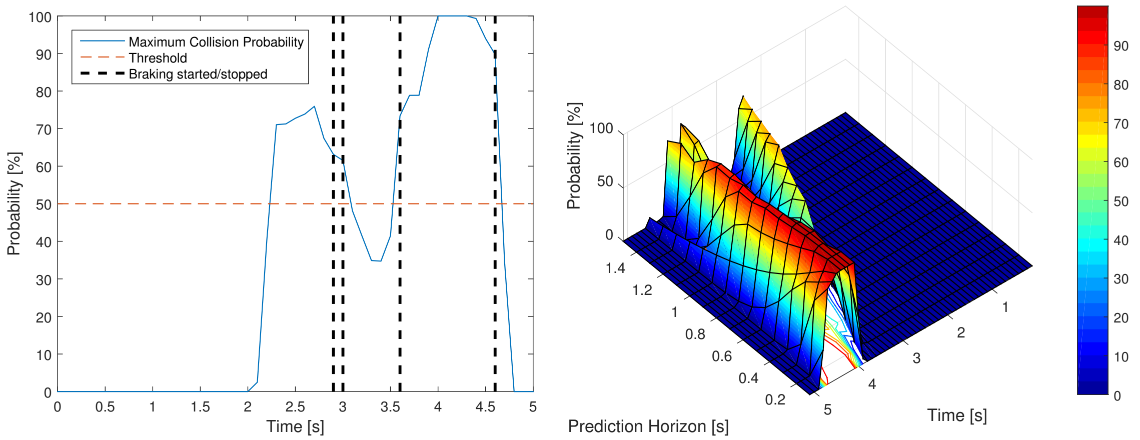
(a) BN for the EGO.

(b) IMM for the EGO.

(c) BN for the POV.

(d) IMM for the POV.

Figure B.30: The BN probabilities as well as the IMM weights for each of the three models in the simulation are here shown for both the EGO and the POV. The three different models are represented by red for left turn, black for right turn and blue for going straight. Additionally is the influence of the braking included for the EGO (black, dotted).



(a) Maximum probability of collision of the prediction horizon, for each time step. Result from simulation in PreScan.

(b) Probability of collision in each prediction step for every time step. Result from simulation in PreScan.

Figure B.31: Probability of collision according to the collision detection subsystem, where Figure B.31a also illustrates the threshold that triggers the *Danger* signal.

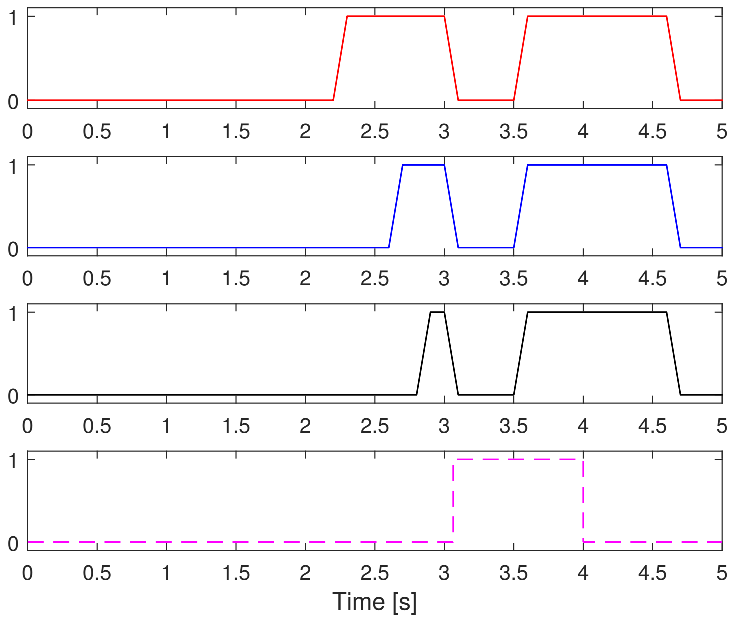
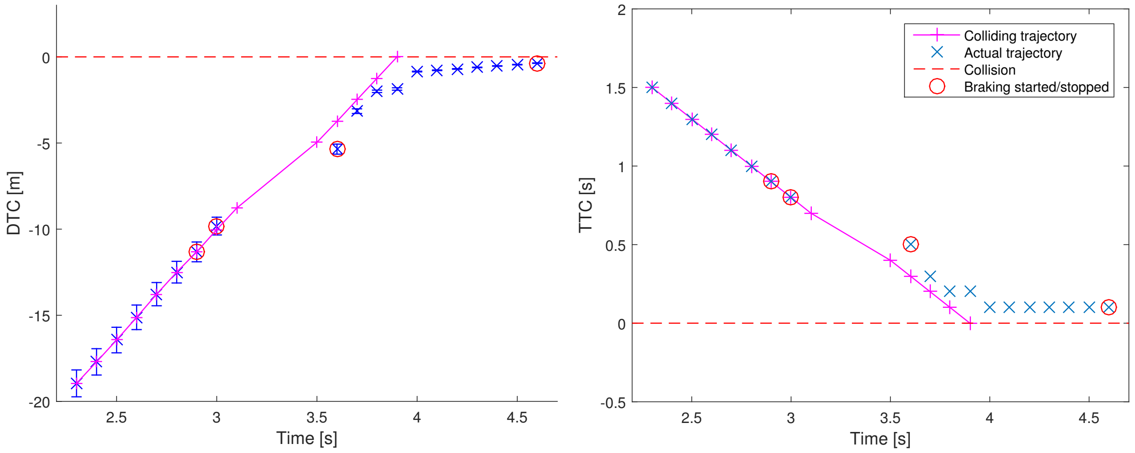


Figure B.32: Illustration of the intervention timing from when a threat is detected to when an braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).



(a) The evolvement of DTC over time.

(b) The evolvement of TTC over time.

Figure B.33: Here are the DTC and TTC over time seen for the simulation. Blue represents the estimated DTC/TTC in each time step when the EGO executes an intervention. The red circles represent when the brakes are triggered and later released. Magenta represents the estimated DTC/TTC for a vehicle without a CA system. Figure B.33a also illustrates the distance uncertainty with vertical bars. The idea is that the vehicle should never end up in a DTC or TTC equal to 0, since this is when a collision occurs.

B.2.5 Road merging scenario:

Test scenario: As a part of cooperative driving evaluation, this test aims at evaluating the behaviour of the EGO when it is travelling on a road merging into another road. One example is when the POV is travelling on a straight path and the EGO travels on the entrance lane to this path, for example an entrance lane to the highway. An illustration of the intended scenario can be seen in Figure B.34.

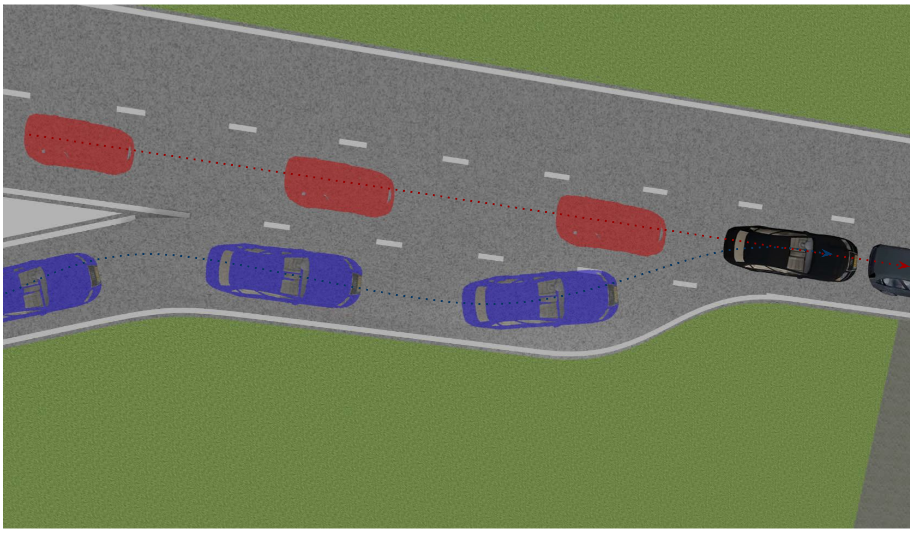
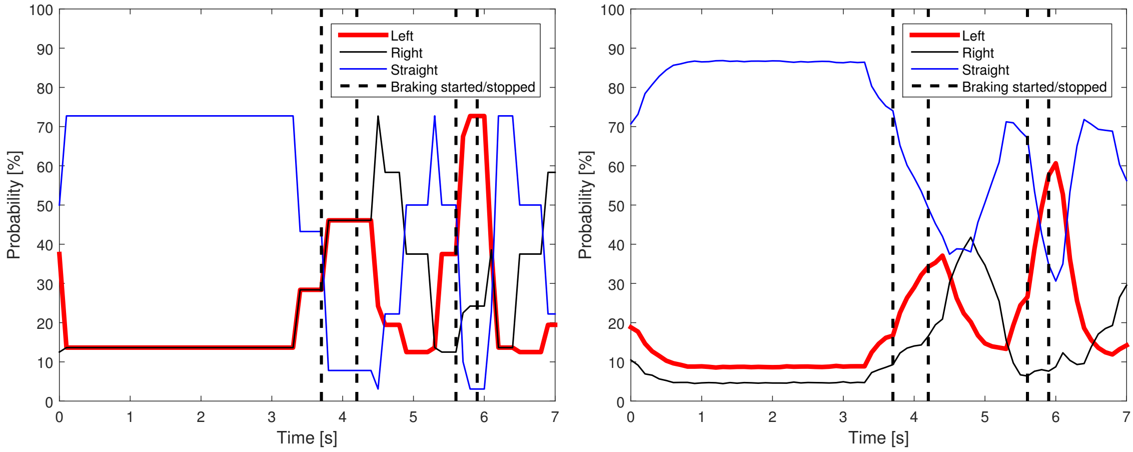


Figure B.34: Illustration of the used scenario for evaluation. Blue represents the EGO's trajectory whereas red represents the POV's, black represents the end positions for both vehicles in the scenario.



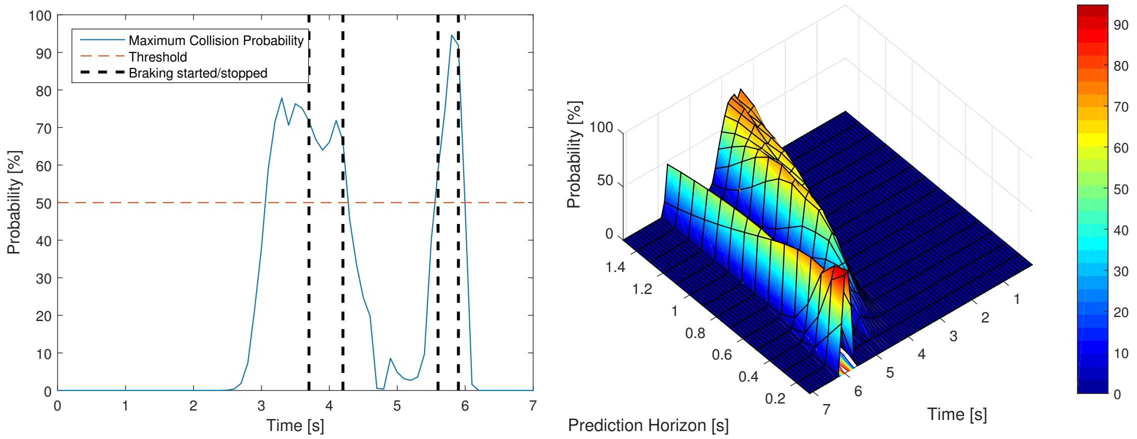
(a) BN for the EGO.

(b) IMM for the EGO.

Figure B.35: The BN probabilities as well as the IMM weights for each of the three models are here shown for the EGO. The three different models are represented by red for left turn, black for right turn and blue for going straight. Additionally is the influence of the braking included for the EGO (black, dotted).

The EGO is in this scenario going straight, turning right and thereafter left to merge into the POV's lane to continue on a straight path. This is illustrated by the BN and IMM in Figure B.35. The IMM is bringing down the probability of going right, which is reasonable since the turn is just a soft transfer into a new lane.

The BN and IMM for the POV is behaving as expected, similar to what was seen in the bottom plots of Figure 6.9.



(a) Maximum probability of collision of the prediction horizon, for each time step.

(b) Probability of collision in each prediction step for every time step.

Figure B.36: Probability of collision according to the collision detection subsystem, where Figure B.36a also illustrates the threshold that triggers the *Danger* signal.

Since the EGO is approaching the POV's lane from the right side but then turns right, the probability of collision goes above the threshold at an early time instance. When the EGO thereafter performs the actual turn into the lane, a braking sequence is yet again initiated as can be seen in Figure B.36 and B.37. This phenomena occurs due to the connection of the roads, where in Figure B.38 the TTC and DTC can be seen to change values drastically at these time instances.

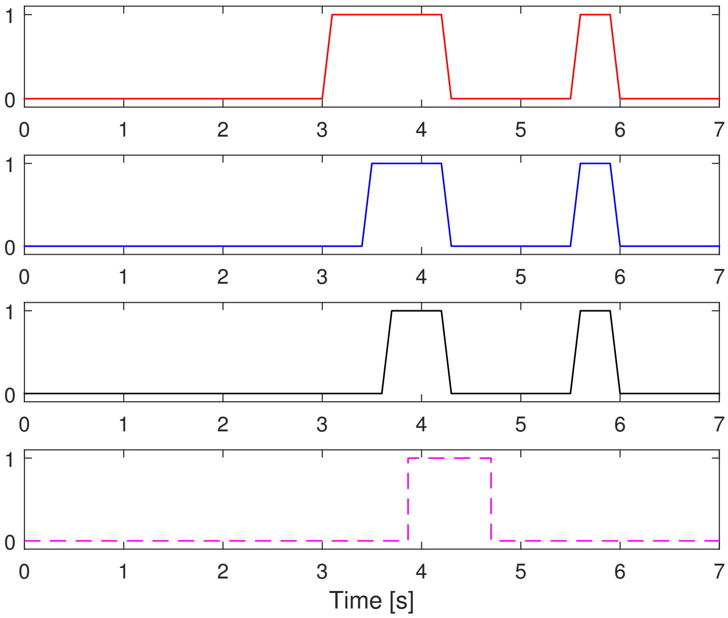
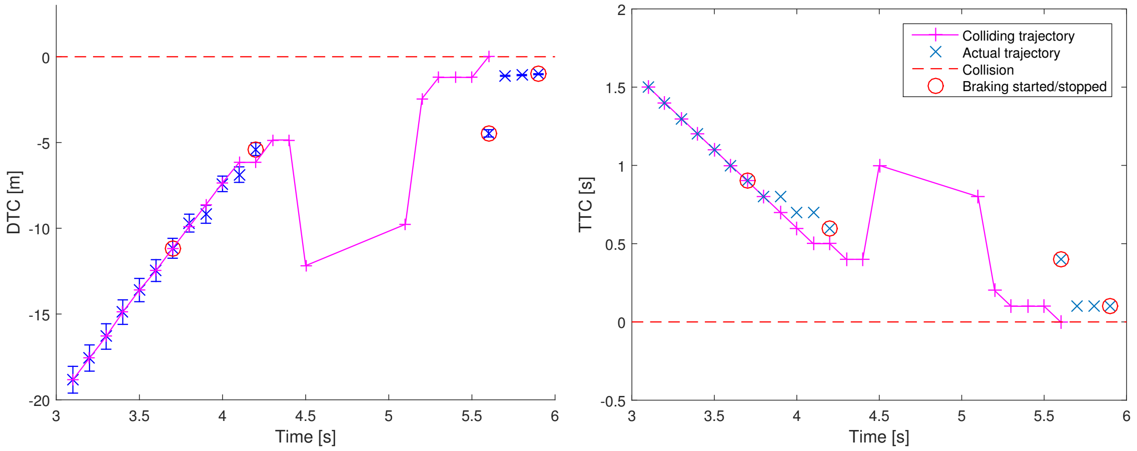


Figure B.37: Illustration of the intervention timing from when a threat is detected to when a braking intervention is performed. From top to bottom, threat detected in collision detection subsystem (red), intervention performed in collision avoidance subsystem (blue), brakes initiated after reaction time (black) and optimal braking profile (magenta, dotted).



(a) The evolvement of DTC over time.

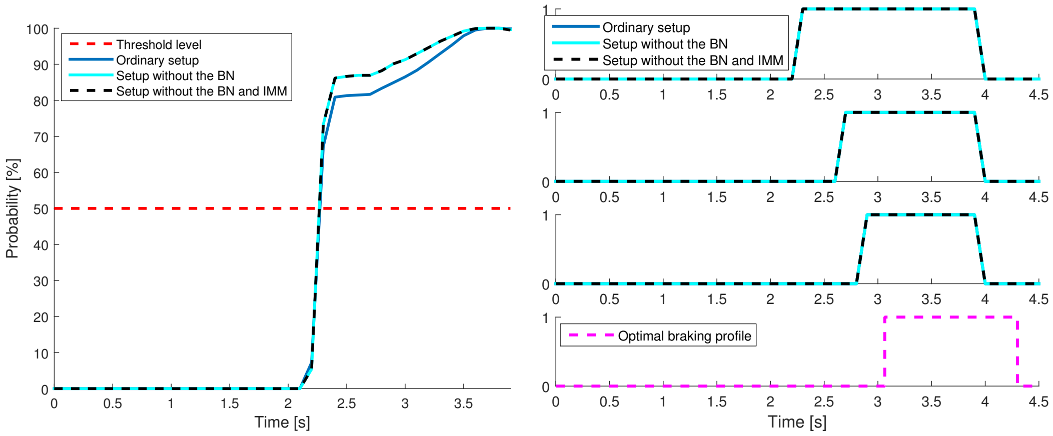
(b) The evolvement of TTC over time.

Figure B.38: Here are the DTC and TTC over time seen for the simulation. Blue represents the estimated DTC/TTC in each time step when the EGO executes an intervention. The red circles represent when the brakes are triggered and later released. Magenta represents the estimated DTC/TTC for a vehicle without a CA system. Figure B.38a also illustrates the distance uncertainty with vertical bars. The idea is that the vehicle should never end up in a DTC or TTC equal to 0, since this is when a collision occurs.

C
System robustness

C.1 Performance comparison for the crossing intersection scenario

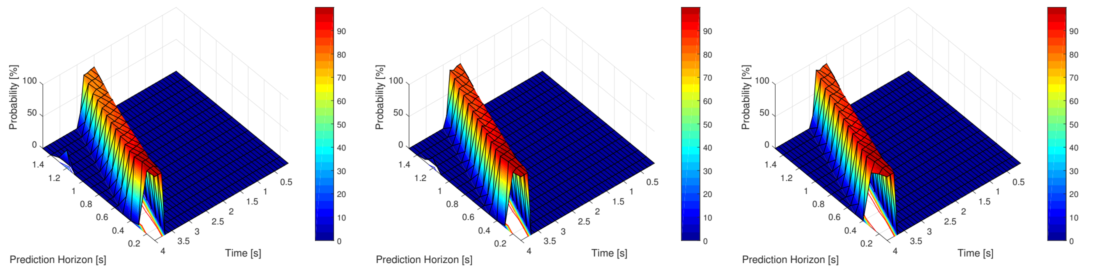
Figure C.1 and C.2 illustrates the comparison of using the ordinary system setup with having the BN turned off as well as both the BN and IMM turned off.



(a) Collision probability **(b)** Threat/brake assessment

Figure C.1: Illustration of how each part influences the overall subsystems behaviour for the crossing intersection scenario, where (a) shows the probability of collision and (b) the threat/brake assessment for the three different setups.

The evaluation of the crossing intersection scenario resulted in the system being equally good at detecting and evade a collision irrespective of if either the BN or IMM was active. This is explained by that the intended manoeuvre in the crossing intersection scenario is a straight path hence even without both BN and IMM, the system is being able to evaluate this scenario to be an imminent risk of collision.



(a) Ordinary setup

(b) Setup without the BN

(c) Setup without the BN and IMM

Figure C.2: A probability graph showing the detection rate of a probable collision for the left turn across path by EGO scenario, where (a) illustrates the ordinary setup, (b) without the BN and (c) shows the affect of not using the IMM as well.

D

Test cases for evaluation of the collision detection algorithm

D.1 Apart

In this scenario both the vehicles are driving against each other on a straight path. The different steps of the scenario can be seen in Figure D.1.

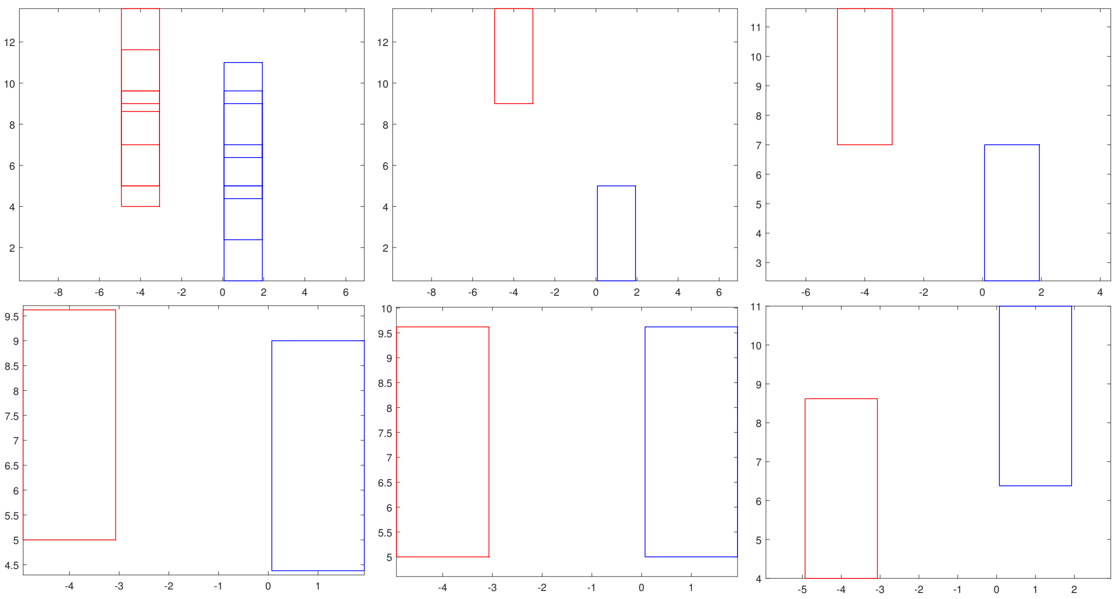


Figure D.1: The figures are depicting the scenarios used for the evaluation of the collision detection subsystem. From top left to bottom right: The whole scenario, P1, P2, P3, P4 and P5. Here blue represent the EGO and red the POV.

D.2 Next to eachother

In this scenario, similar to the previous scenario, is the vehicles travelling on a straight path. The difference is that they are hear approaching eachother at a closer distance sideways. The details of the scenario can be seen in Figure D.2.

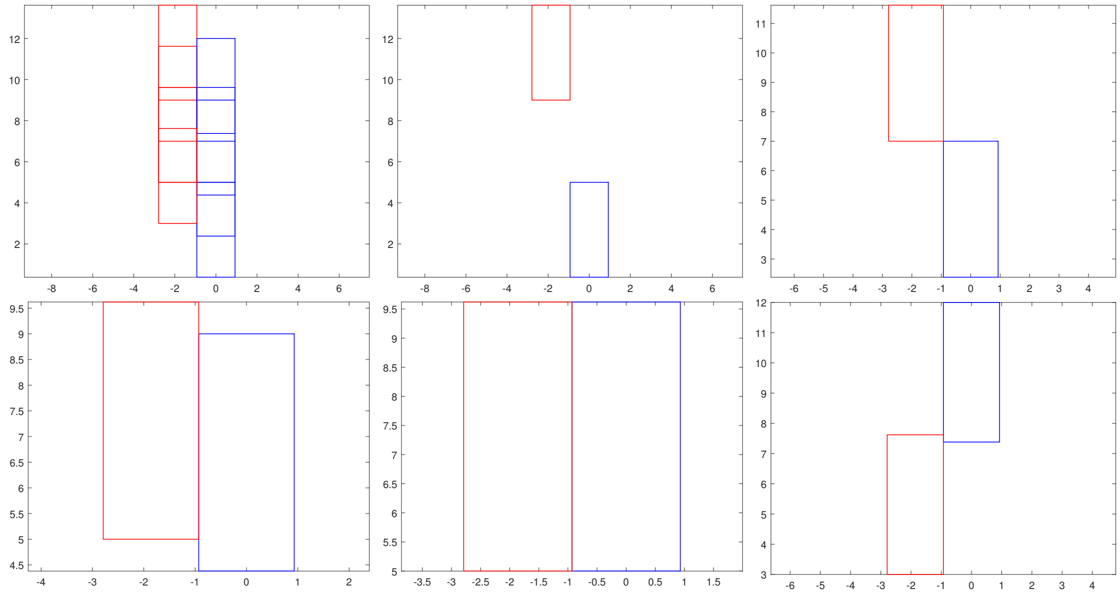


Figure D.2: The figures are depicting the scenarios used for the evaluation of the collision detection subsystem. From top left to bottom right: The whole scenario, P1, P2, P3, P4 and P5. Here blue represent the EGO and red the POV.

D.3 Overlapping

In the third scenario is the EGO driving into the POV more and more, which gives a growing overlapping area, this can be seen in Figure D.3. In the last picture of the Figure is the vehicles overlapping 100%.

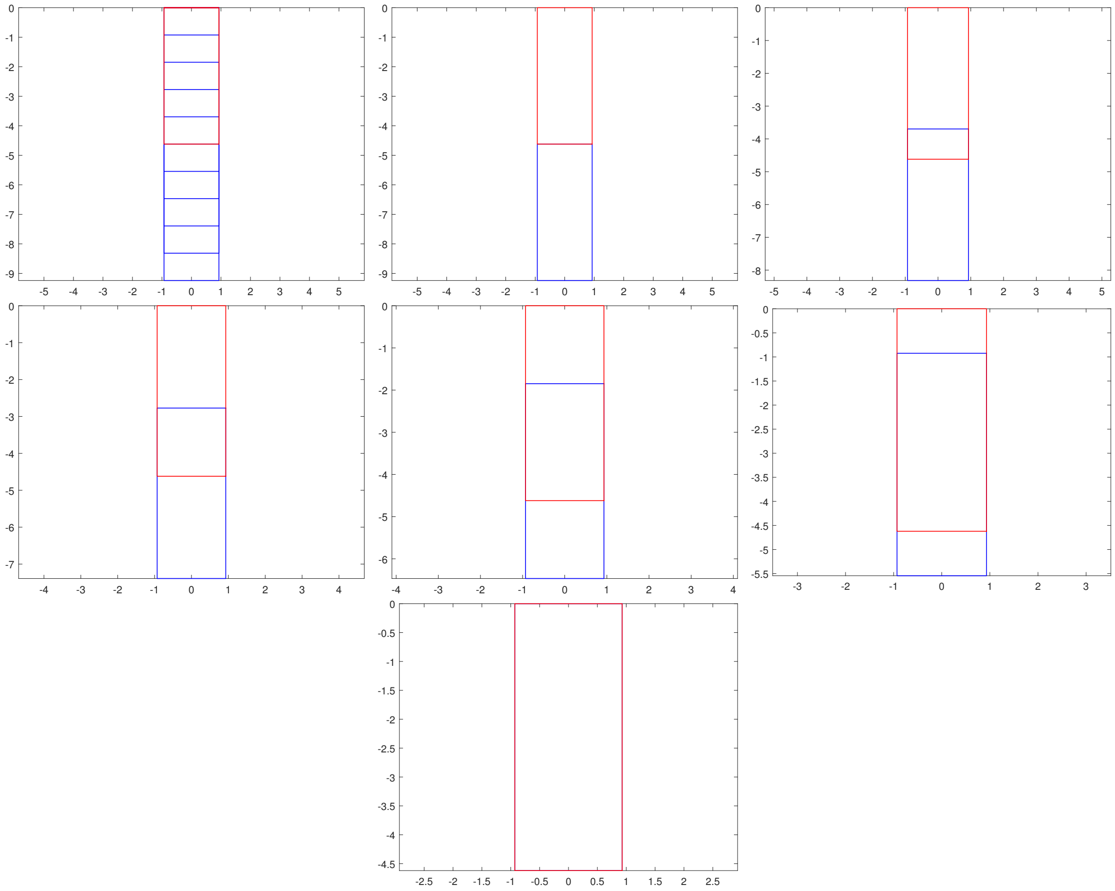


Figure D.3: The figures are depicting the scenarios used for the evaluation of the collision detection subsystem. From top left to bottom right: The whole scenario, P1, P2, P3, P4, P5 and P6. Here blue represent the EGO and red the POV.

D.4 EGO into POV

In this scenario the EGO is driving into and through the long side of the POV, seen in Figure D.4.

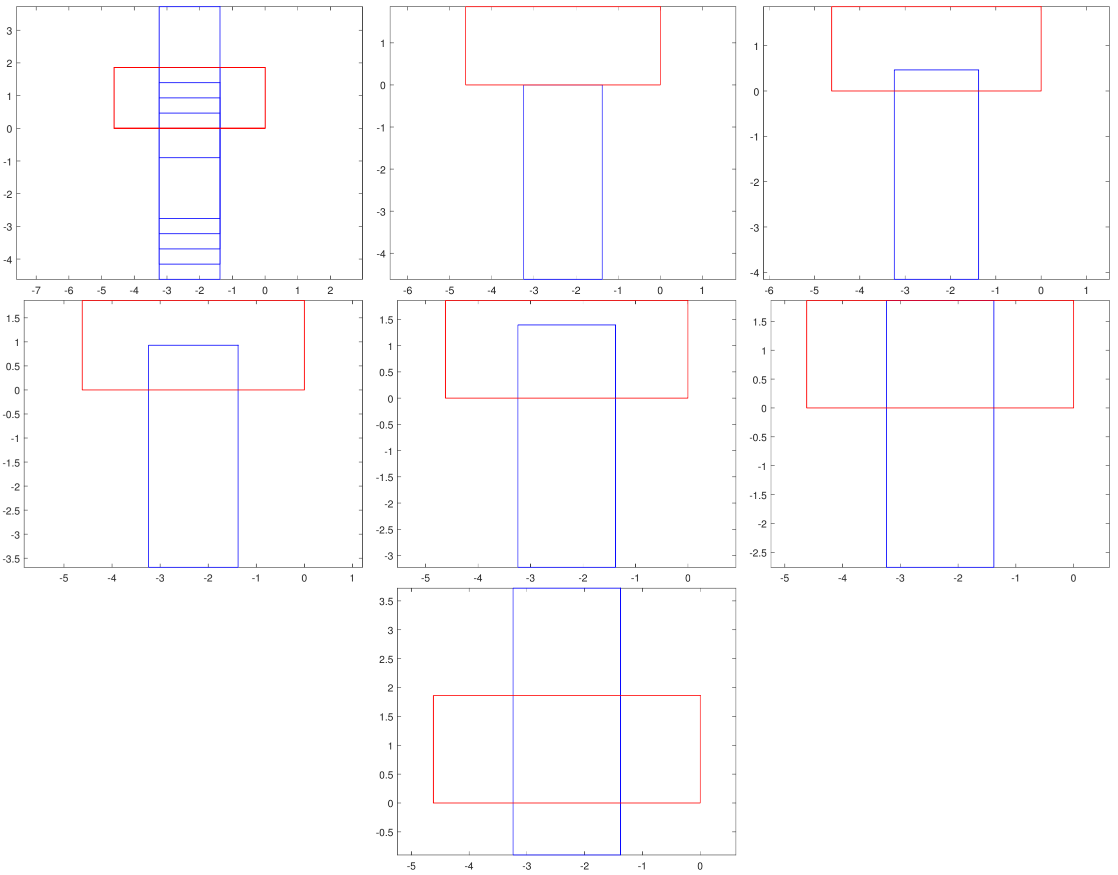


Figure D.4: The figures are depicting the scenarios used for the evaluation of the collision detection subsystem. From top left to bottom right: The whole scenario, P1, P2, P3, P4, P5 and P6. Here blue represent the EGO and red the POV.

D.5 POV into EGO

Similar to the previous scenario, but here is the POV driving through the EGO seen in Figure D.5.

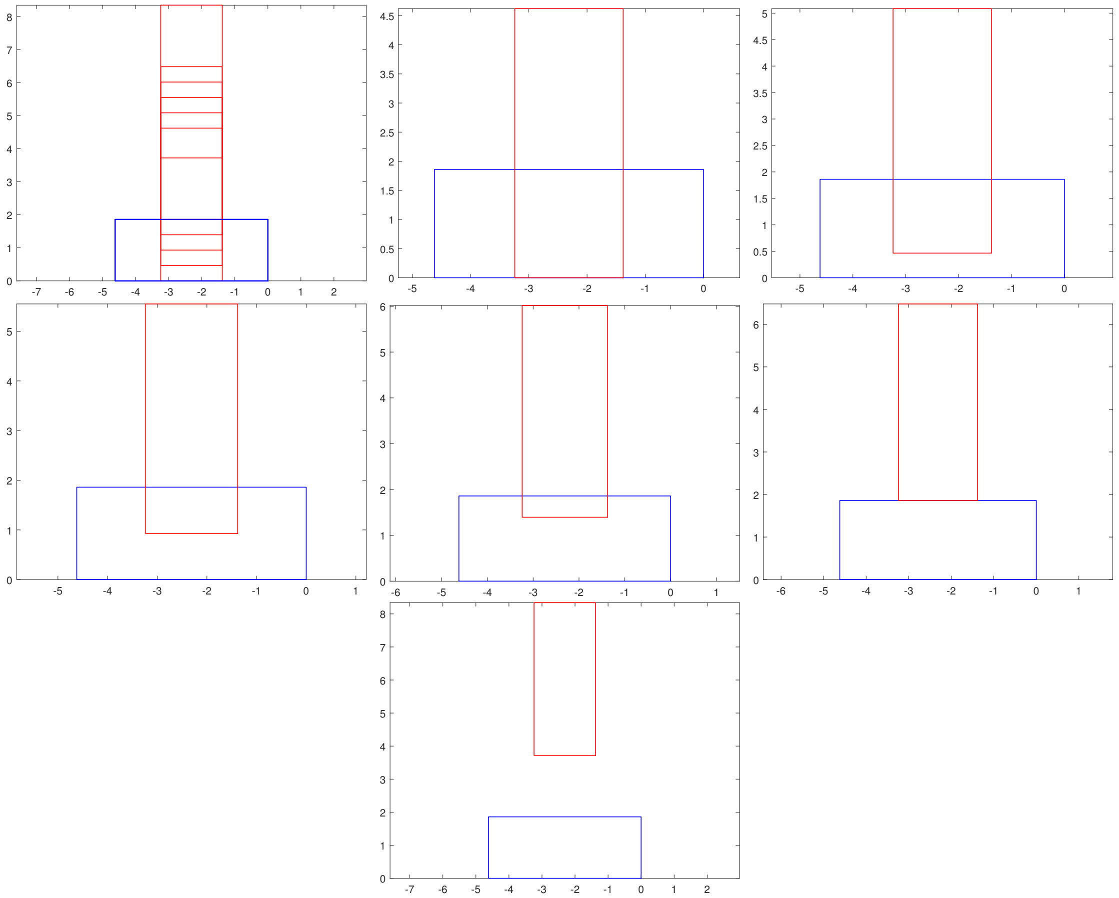


Figure D.5: The figures are depicting the scenarios used for the evaluation of the collision detection subsystem. From top left to bottom right: The whole scenario, P1, P2, P3, P4, P5 and P6. Here blue represent the EGO and red the POV.

D.6 EGO turn, POV straight

Here is the EGO turning at the same time as the POV is going straight, which results in some interesting configurations of the vehicles seen in Figure D.6.

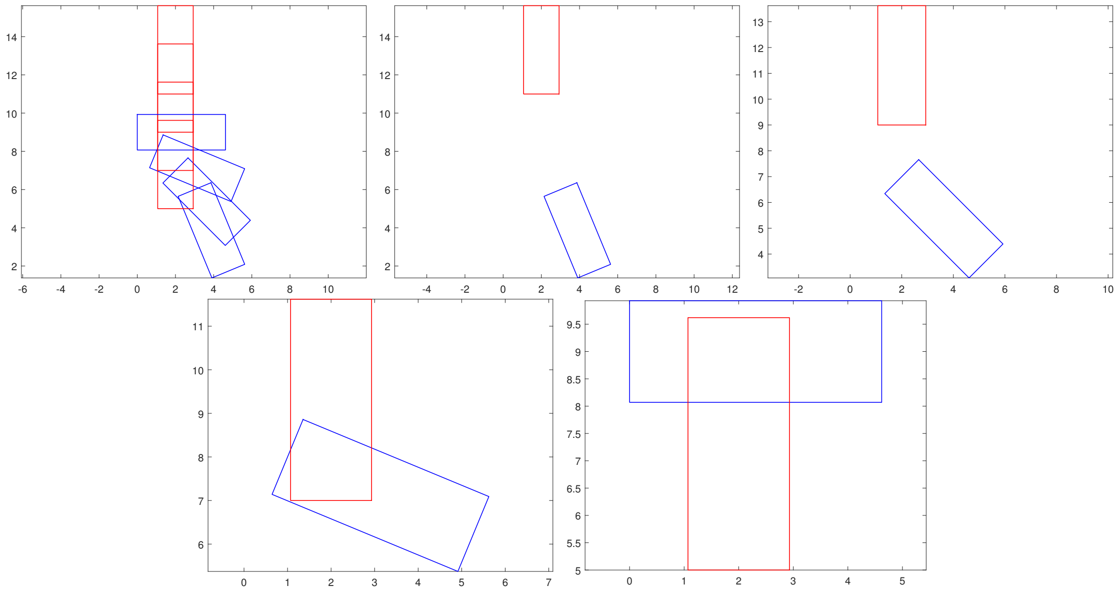


Figure D.6: The figures are depicting the scenarios used for the evaluation of the collision detection subsystem. From top left to bottom right: The whole scenario, P1, P2, P3 and P4. Here blue represent the EGO and red the POV.

D.7 EGO and POV turn

In the last scenario is both of the vehicles turning, leading to interesting results. The configuration of the scenario can be seen in Figure D.7

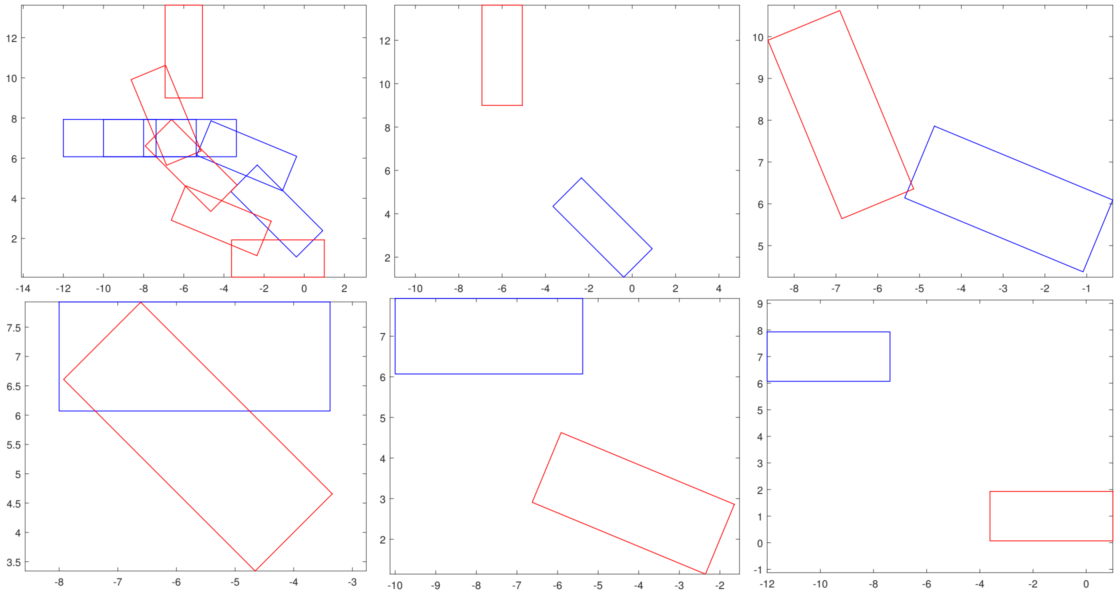


Figure D.7: The figures are depicting the scenarios used for the evaluation of the collision detection subsystem. From top left to bottom right: The whole scenario, P1, P2, P3, P4 and P5. Here blue represent the EGO and red the POV.