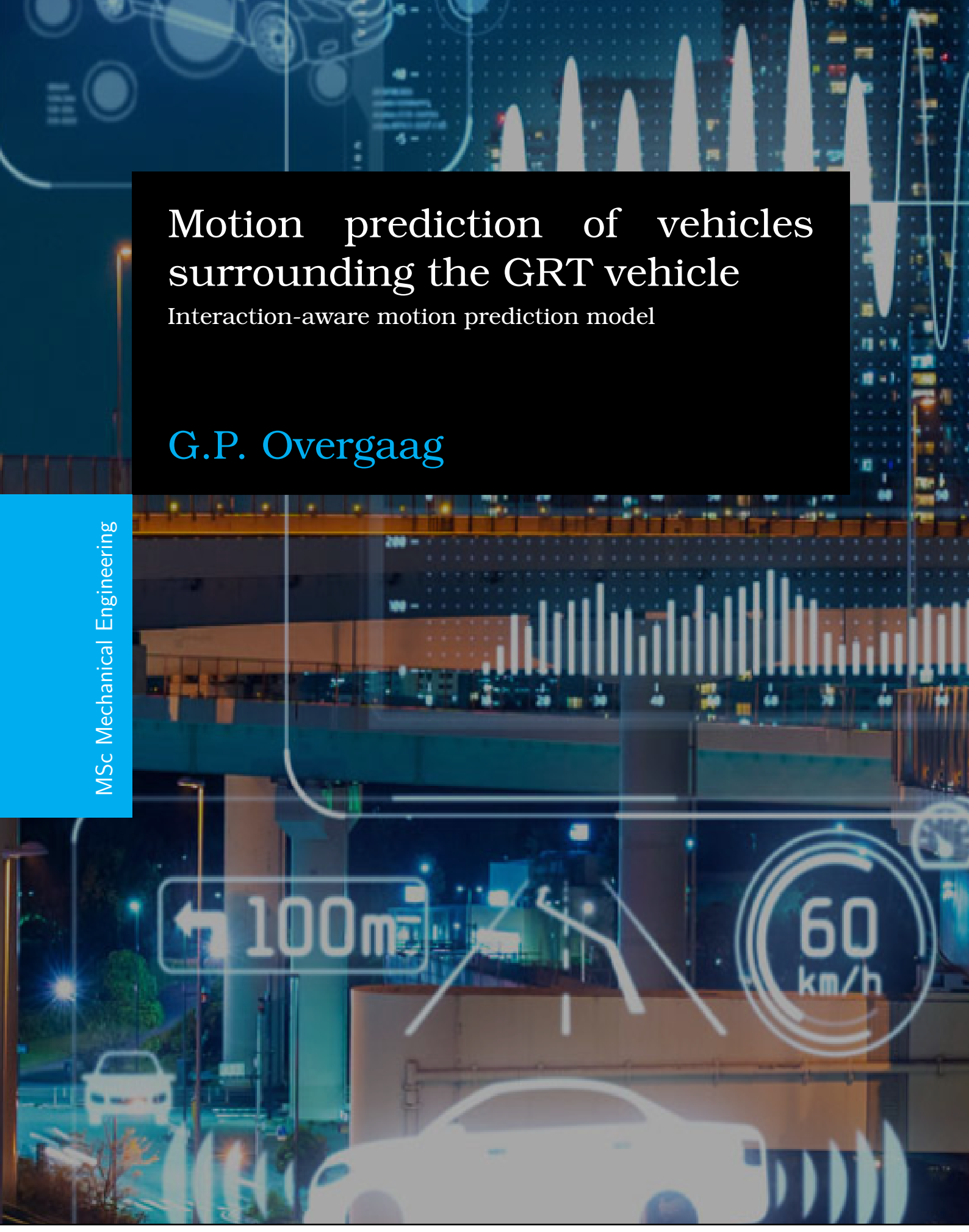

Motion prediction of vehicles surrounding the GRT vehicle

Interaction-aware motion prediction model

G.P. Overgaag

MSc Mechanical Engineering

TU Delft Delft University of Technology

Delft Center for Vehicle Engineering

Motion prediction of vehicles surrounding the GRT vehicle

Interaction-aware motion prediction model

Master of Science Thesis

G.P. Overgaag

May 31, 2021

Student number:	4351894	
Thesis duration:	August 1, 2020 - May 31, 2021	
Thesis committee:	Prof. dr. ir. B.H.K. De Schutter,	TU Delft (3mE)
	Prof.ir. B. Shyrokau,	TU Delft (3mE)
	ir. R. Ruigrok,	2getthere

Faculty of Mechanical Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by 2getthere (a company of ZF). Their cooperation is hereby gratefully acknowledged.

Abstract

2getthere specialises in autonomous people transport through their GRT vehicle, used for transporting people at the airport from the parking to the terminal to ensure the GRT can operate comfortably and safely in a mixed traffic environment. The vehicle needs to plan a smooth and collision-free path. An essential aspect of a safe path is to predict the motion of the surrounding vehicles of the GRT to guarantee a smooth adaption to the constantly changing traffic scenario. The smooth adaption to the changing traffic scenario is essential for the GRT. The GRT cannot brake with the same magnitude as surrounding traffic due to standing people inside the vehicle. The importance of the smooth adaption to the traffic scenario resulted in the following problem statement; is it possible to create a motion prediction model for the surrounding vehicles of the ego vehicle to allow for pro-active velocity planning of the ego vehicle itself. From the literature, motion prediction models can be divided into physics-based motion models, manoeuvre-based motion models, and interaction-aware motion models. Each level has a certain amount of building blocks, with each level being an extra addition of situation awareness to the previous level.
An interaction-aware motion model is the most advanced prediction model taking the infrastructure and interaction between the vehicles in the traffic scene into account. Therefore this principle of motion prediction is chosen for the motion prediction of the surrounding vehicles of the GRT. First, all possible state and route information of the surrounding vehicles are gathered. Secondly, a tree is created based on the vehicles' routes in the traffic situation, with the unique combination of manoeuvres known as a single branch of the scenario tree. Hereafter, the possible conflict areas between the individual manoeuvres are determined based on the desired trajectories for a specific route. The first vehicle passing the conflict area will influence the second vehicle's trajectory for passing the same conflict area, which will influence the third vehicle's trajectory and so on. Each vehicle can avoid a collision by passing in front or behind the previous vehicle(s) at the conflict area. Evaluating all passing possibilities will result in the velocity trajectory with the lowest cost, constrained by comfort. The sum of all costs for each branch of manoeuvres is used as a metric to determine the possible velocity profiles for each vehicle in the current traffic situation, resulting in a better understanding of the traffic scenario and ensuring a comfortable adaption to a changing traffic scenario. This thesis will evaluate several methods found in the literature to built a motion prediction model suitable for the application at 2getthere. The chosen motion model will be evaluated by two scenarios in a T-junction intersection.

Acknowledgement

During my thesis, I have been supported by a great set of people to get me to where I am now. Primarily, I am grateful to 2getthere for allowing me to work on this thesis topic. During my time at 2getthere, I have been pleased with the assistance I received from the people at 2getthere. With this, I would like to give a special thanks to Rob Ruigrok, who has been my daily supervisor during this period and also I would like to thank Jeroen van der Ploeg for supervising my thesis on a higher level.
Besides the people at 2getthere, I would also like to thank Bart de Schutter for being my professor supervisor and keeping an eye on the project's direction during the whole process.
Moreover, I would like to thank the TU Delft for the past years. With this, I thank the professors and coordinators at the faculty of Cognitive Robotics for creating an exciting master's program that was very educational and amusing.
It has been an exciting experience to conduct the thesis mainly from home due to the COVID-19 pandemic. Despite this, I learned a lot, and I am very grateful for the opportunity I received from the TU Delft and 2getthere.

Table of Contents

Chapter 1

Introduction

2getthere specialises in autonomous people transport through their GRT vehicle, used for transporting people at the airport from the parking to the terminal. To ensure the GRT can operate comfortably and safely in a mixed traffic environment, the vehicle needs to plan a smooth and collision-free path. An essential aspect of a safe route is to predict the motion of the surrounding cars of the GRT to guarantee a smooth adaption to the constantly changing traffic scenario. The smooth adaptation to the changing traffic scenario is essential for the GRT because the GRT cannot brake with the same magnitude as surrounding traffic due to standing people inside the vehicle. Currently the GRT uses an Adaptive Cruise Control (ACC) with emergency braking. The emergency braking will be activated if the distance to the leading vehicle becomes to small, resulting in a heavy braking. In all circumstances it is desired not to activate the emergency braking, because it will brake with maximum braking force resulting in a very unpleasant drive for the passengers.

Figure 1-1: The Group Rapid Transit of 2getthere for Brussels Airport

1-1 Motion prediction: motion prediction levels

The concept of predicting the motion of the surrounding vehicles has been broadly researched in the literature, resulting in three general levels of motion prediction [28]. The three levels will each be explained using a simple traffic scenario seen in the Figures 1-2, 1-3 & 1-4. This traffic scenario contains a blue vehicle approaching a T-junction, with a black vehicle approaching from the right. According to the traffic rules, the blue vehicle will have to give way

to the black vehicle to avoid a collision. The prediction models will predict the motion of the blue vehicle in the traffic scenario. Each image in the figures below is a separate timestamp of the scenario.
The first level is a physics-based motion model, taking only the vehicle kinematics and dynamics of each vehicle into account. This way, the actor will be addressed as a moving object without any dependencies or interactions to the surrounding environment (vehicles and infrastructure). Figure 1-2 depicts a graphical overview of such a motion model in the simple T-junction scenario described above. The physics-based motion model will extract the necessary data (e.g. position, velocity) of the blue vehicle at the left image of the figure. For instance, if the car is driving at 30 km/h with zero acceleration, the motion model predicts that the blue car will continue its path with 30 km/h over the predefined time horizon. If at the next state update, the vehicle is driving 25 km/h with an acceleration of -1 m/s^2, the physics-based model will predict that the blue car will continue decelerating with the same rate of -1 m/s^2 till the next state update. The most common type of physics-based motion model is a constant acceleration model. The slow adaption to the traffic scenario without taking the infrastructure and surrounding vehicles into account will result in a collision, as seen in the picture on the right of Figure 1-2.

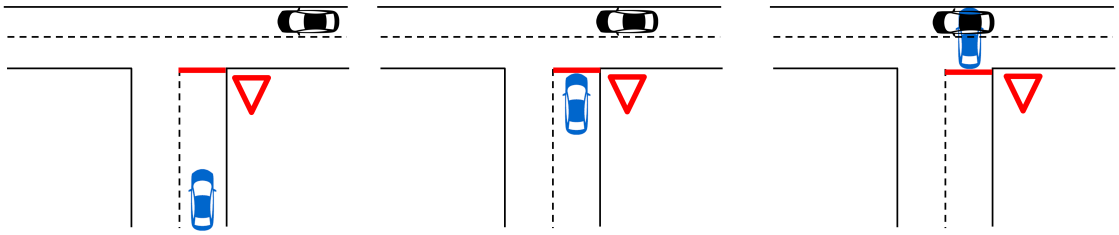

Figure 1-2: A graphical representation of a physics-based motion model in a T-junction scenario with the ego vehicle represented as the blue car [28]

A manoeuvre-based motion model is the second level of motion prediction models. Figure 1-3, depicts a graphical overview of such a prediction model in the same traffic scenario as in Figure 1-2. The essence of a manoeuvre-based motion is defined as follows; *'Maneuver-based motion models represent vehicles as independent maneuvering entities, i.e. they assume that the motion of a vehicle on the road network corresponds to a series of maneuvers executed independently from the other vehicles.'* [28]. In other words, it means that the state information and the infrastructure's layout are taken into account for the motion model. In the left picture of Figure 1-3 the vehicle state of the blue vehicle (e.g.position, velocity and acceleration) and the surrounding infrastructure is captured. Therefore the manoeuvre-based motion model knows that the blue car is approaching a T-junction, where it will have to execute a turn (right or left). The vehicle's predicted motion will be adapted accordingly, knowing it is approaching a T-junction. In the case of Figure 1-3 the blue actor drives with 30 km/h with zero acceleration while approaching the intersection. Although the blue vehicle has not started decelerating yet, the manoeuvre-based motion model still predicts a motion model for both a turn left and right. As the traffic scenario progresses (from left to right), the blue vehicle will execute a turn left. However, it will still collide with the black car because there is no information about traffic rules or surrounding vehicles taken into account.

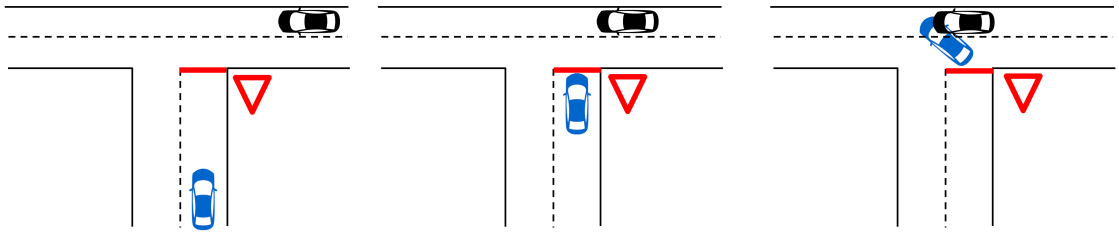

Figure 1-3: A graphical representation of a manoeuvre-based motion model in a T-junction scenario with the ego vehicle represented as the blue car[28]

The final level, known as an interaction-aware motion model, is an addition to the previous manoeuvre-based motion model by taking surrounding traffic into account. Figure 1-4 shows such a motion model in the T-junction scenario described at the beginning of this section. The motion model will know that the blue car is approaching a give-way intersection by taking the traffic rules and interaction into account. The motion model needs information about the surrounding actors approaching the T-junction to create the appropriate motion prediction for the blue vehicle. Therefore, the motion model will also gather the black car's data (position and velocity) approaching the intersection. By extracting the data of surrounding vehicles, it can calculate the arrival time at the conflict point and how long a car will occupy the intersection. As an example, the scenario depicted in Figure 1-4 shows that the blue vehicle will stop at the T-junction to let the black car pass the traffic scenario avoiding a collision. Therefore, the interaction-aware motion model is the most advanced and is closest to humans' actual driving behaviour.

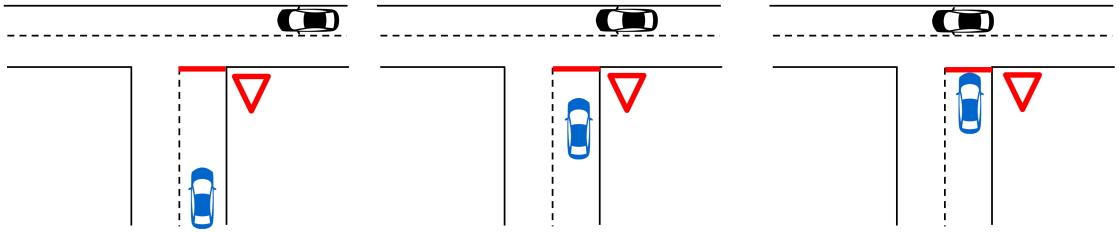

Figure 1-4: A graphical representation of an interaction-aware motion model in a T-junction scenario with the ego vehicle represented as the blue car [28]

1-2 Motion prediction: block overview

By reading and comparing papers from the literature about motion prediction models, each model uses the same building blocks to create a motion prediction for the relevant vehicles shown in Figure 1-5. Therefore, this thesis is divided into these individual steps to ensure a clear overview of the chosen approach. The amount of building blocks is different for each of the three levels mentioned in the previous Section 1-1. The first block, measurements, is used in all motion predictions. The measurement block determines the relevant vehicle's current state (e.g. position and velocity). The model only gathers the ego vehicle's state information for the physics-based and manoeuvre-based motion models. In contrast, the

model collects all surrounding vehicles' states for the interaction-aware motion models. Secondly, the block route information is only used in manoeuvre-based and interaction-aware motion models shown in Figure 1-5. The route information gathers the infrastructure layout for the ego vehicle's immediate surroundings to understand the traffic scenario based on the infrastructure. The motion model will determine the possible manoeuvres for each relevant vehicle. A route option is a designation for a specific type of action, taking a left/right turn or merging between two cars, it does not say anything about the execution of such a manoeuvre. The third building block is the interaction determination, which is only present in the interaction-aware motion model due to advanced understanding of the traffic scenario and surrounding vehicles. The interaction determination will decide how a particular traffic scenario will progress using traffic rules and surrounding traffic. In other words, it determines which vehicles have the right of way and which have to give way. The fourth building block comes into play by combining the three previously mentioned steps to calculate each relevant vehicle's predicted trajectory in the traffic scene. A trajectory estimates how a specific manoeuvre is executed by finding the corresponding velocity/acceleration profile based on the state information and manoeuvre type. The trajectory planning is a vast topic in the literature. The literature study will address several methods for trajectory calculation. Finally, the manoeuvre estimation will estimate the probability between the possible manoeuvres for each relevant vehicle. This block is not used for the physics-based motion models because it will only create one trajectory at each time instant. There will be an estimation between the different possible trajectories for the manoeuvres-based and interaction-aware motion model. The literature study addresses several methods for calculating the probability estimation between the possible manoeuvres instantly.

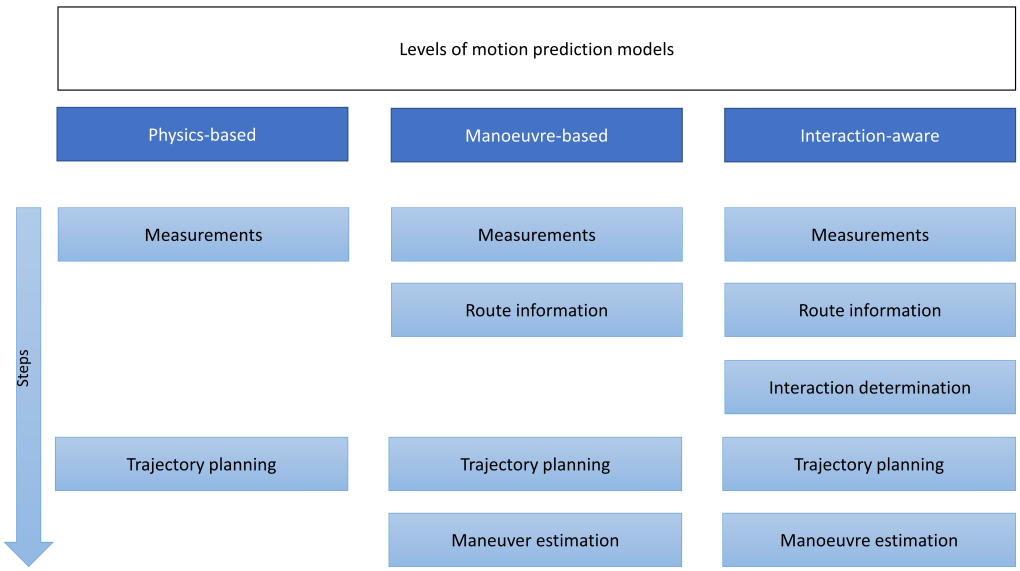

Figure 1-5: Overview of the building blocks for motion prediction

Figures 1-2, 1-3 & 1-4 are addressed in this section to create a better understanding of the building blocks. The measurement block will determine the state of the blue vehicle, which can consist of position, velocity and acceleration. The route intention block will determine that the vehicle is approaching an intersection and has two options, turn left or turn right.

In the interaction determination block, the model gathers the black car's information for approaching this junction from the right. Therefore stopping at the stop line is added to the set of manoeuvres for the blue car. The manoeuvres set is now extended to three manoeuvres; turn left, turn right and stop at the stop-line. The trajectory prediction block will determine a set of trajectory hypotheses for each manoeuvre in the set of manoeuvres. The manoeuvre estimation block uses the predicted trajectories to create a probability between the several manoeuvres (e.g. turn left, turn right and stop).

1-3 Scope & limitations

This thesis aims to develop a motion prediction model for the surrounding vehicles of the ego vehicle to allow for pro-active velocity planning of the ego vehicle itself. To reach this goal the motion model is tested on a predefined map of the Zaventem airport.

The choice for a maximum of four actors is due to the limitations on the sensors of the GRT. The GRT is equipped with a specific amount of sensors, and they are limited in the number of vehicles they can see due to their line of sight. Moreover, these sensors cannot detect and measure the acceleration and yaw rate of the surrounding vehicles. Therefore, the state variables used for the surrounding cars are limited to the velocity's position, magnitude, and direction.

Furthermore, as the main focus lies in the intersection domain, the vehicles are assumed to follow the centerline of a lane. Moreover, overtaking behaviour is not considered, and vehicles will comply with the maximum velocity for each road section. The top speed of a straight road section will be $14m/s$, while for turns, their radius constrains the maximum.

Moreover, the scenarios are created by the driving simulator toolbox in MATLAB. The toolbox determines the velocity profile of a vehicle based on the drawn waypoints. The waypoints hold information about velocity and position of vehicle. The toolbox will linearly create a velocity profile between the waypoints, resulting in step function for the acceleration profiles.

The created motion model is mainly focused on creating a framework for motion prediction. It is not focused on creating the best tuned motion prediction model, but more on creating a modular prediction model that could be used in all type of scenarios and future work.

1-4 Thesis outline

The upcoming chapters of this thesis are structured as follows. Chapter 2 reviews the existing state-of-the-art for each building block mentioned in Section 1-2 found in the literature. The most suitable and common methods will be introduced and briefly explained for each building block. Hereafter, Chapter 3 will highlight the chosen methods for each building block and how these preferred methods are used in the final model. Chapter 4 will first introduce the set-up for both scenarios. Hereafter, the chapter shows the results of the model and will discuss these results. Moreover, the results will show the predicted velocity profiles of the lead vehicle and the ego vehicle with their corresponding probabilities based on the motion prediction of the surrounding vehicles. Lastly, in Chapter 5, conclusions on the developed model are drawn, and the thesis is closed by recommendations for further research on this topic.

Chapter 2

State of the motion prediction literature

This chapter gives an insight on the current state of the art found in the literature, using the building blocks described in Chapter 1. The literature's commonly used methods are addressed and briefly explained in the coming chapter. Firstly, the measurement block gathers the individual state information for each vehicle. Hereafter, Section 2-2 will introduce several methods for collecting the route possibilities (also known as manoeuvres) for each actor. Third, the various methods for interaction determination between the vehicles in the traffic scenario are introduced. Fourthly, the possible ways for determining the trajectory for each manoeuvre of the actors in the scenario are briefly explained. Section 2-5 will introduce the methods for manoeuvre estimation. Finally, a summary of this chapter will be given.

2-1 Block 1: Measurements

This section is dedicated to the first building block, known as measurements. The measurement block is present in all levels of motion prediction shown in Figure 1-5. Inside the measurement block, the needed states are collected for the motion prediction of the relevant vehicle(s) (e.g. position and velocity). Each motion model in the literature uses different state variables. This section will mention the most common variables used in the literature. In [8, 17, 21, 23, 40, 44, 48] sufficient information is established if only the available position and velocity are collected. Besides these papers, there are also two papers in the interaction domain that use an additional variable; the orientation of a vehicle [15, 27]. Due to the vehicle's nonholonomic properties, the car's orientation is equal to the available velocity direction. Some papers use the vehicle's acceleration as input, seen in [4, 30]. Besides the models adding acceleration, there are methods in the literature which use additional information. For instance, in [14, 37, 38, 43], these models create their motion predictions based on position, velocity, acceleration, and yaw rate. Measuring the yaw rate of vehicles is very complex with sensors, and calculating the yaw rate increases the computational burden of the model.

2-2 Block 2: Route possibilities

Route information determines the possible routes an individual vehicle can execute in the upcoming traffic scenario up to a specified prediction horizon. In many papers, the route possibilities of a vehicle are known as the route information of that particular vehicle. For instance, when a vehicle is approaching a T-junction, as shown in Figure 1-3 & 1-4, the blue car will have two route possibilities; turn left or turn right. In highway driving, a vehicle will always have the possibility to keep its lane or make a lane change to the left or right if these lanes exist. For an intersection scenario, the connection between specific lanes needs to be specified to determine the route possibilities. The block route information is only used in manoeuvre-based and interaction-aware models, as shown in Figure 1-5, by adding the route information to the prediction model, the situation awareness increases, resulting in improved long-term predictions. Differences can be found in how these route possibilities are determined in the literature. This section will describe several methods for determining the route possibilities found in the literature. At first, the breadth-first method determines the route possibilities by slowly increasing the prediction horizon. Secondly, the use of a network will be described by addressing the Bayesian network of [36]. Finally, the papers using predefined route information are addressed.

2-2-1 Breadth-first method

The breadth-first is used in [39, 40], where the route possibilities are determined for intersection scenarios. The breadth-first method determines the route possibilities up to a certain specified distance horizon l_H, by slowly increasing the horizon while checking for the successor lanes at each defined predecessor. As the name suggests, the method first searches in breadth before going in-depth. The meaning breadth-first means that the method will first determine all possibilities for the next step before going a step further into depth. Figure 2-1 depicts the principle of a breadth-first method.

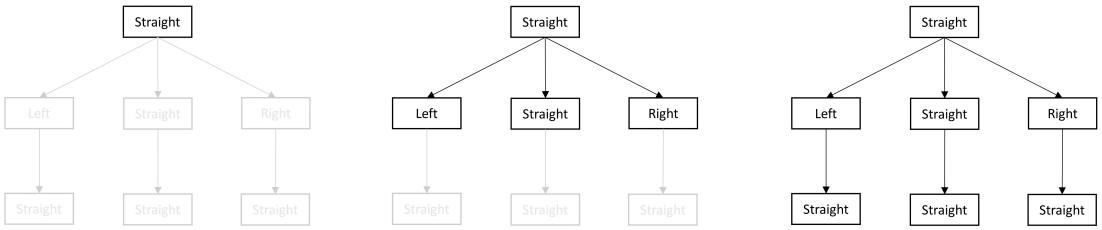

Figure 2-1: A graphical visualisation of the breadth-first method

A visual representation of this method in a traffic scenario is depicted in Figure 2-2, where the route possibilities at each step are shown in green. The method first determines the current lane the vehicle is driving on. Hereafter, the method determines the successor lanes of the current lane, resulting in three independent manoeuvres; straight, left and right. Finally, each manoeuvre has its successor lane shown in the most right picture. The figure shows that the breadth-first method eventually will find three route possibilities in this 4-way traffic scenario for this vehicle.

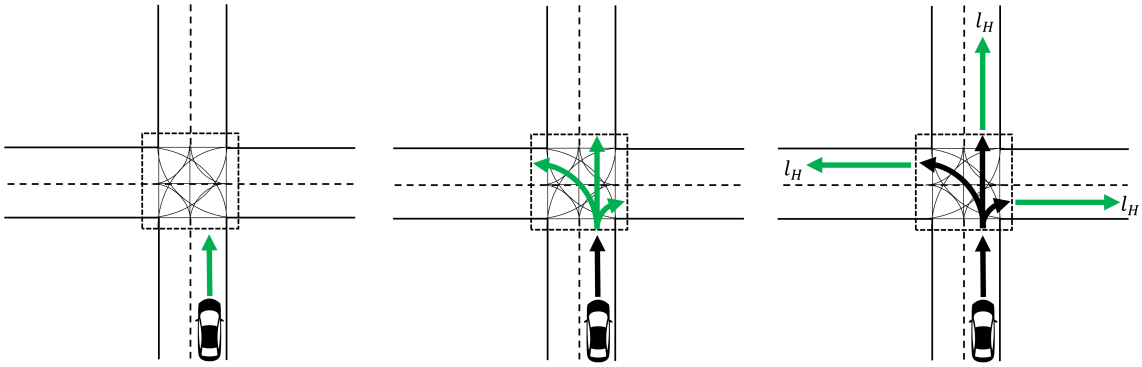

Figure 2-2: The breadth-first method used at an intersection, with the possible routes shown in green

2-2-2 Networks

In the literature, several networks are used to determine the route possibilities. These networks can be divided into three concepts; (Dynamic) Bayesian networks (DBN) and Hidden Markov Models (HMM), and neural networks (NN). The difference between these networks is mainly found in determining the probabilities between the manoeuvres, known as manoeuvre estimation. Therefore this chapter will only highlight the Bayesian network as an example of how the route possibilities for a network are found. In Chapter 2-5 also the application of a neural network will be mentioned to highlight the difference in manoeuvre estimation.

Bayesian networks

"The Dynamic Bayesian Networks are Bayesian networks for modelling time series data. In time-series data an event can cause another event in the future, but not vice-versa" [9]. This results in a simplified design of a Bayesian network. Also, Hidden Markov Models fall in the same class as dynamic Bayesian networks [9]. These type of network models are used in [10, 17, 21, 23, 26, 27, 37, 40], where a predefined network is compared to the situation context at each time iteration. A Bayesian network is represented by a directed acyclic graph that contains a set of nodes and links, where the links present the relationship between the nodes. The nodes represent random variables, and the edges represent the relationship between the nodes. A conditional probability table represents the distributions to determine the uncertainty of an event occurring based on each random variable's conditional probability distribution. A large data set is necessary to determine these conditional probabilities. If the probability table can be determined, the concept of a Bayesian network can be used for motion prediction. The concept of a Bayesian network is explained by the Bayesian network used in [36, 37]. The network is described as a dynamic Bayesian network because it determines the possible routes for every iteration. In Figure 2-3 can be seen that the network contains three separate layers (causal evidence layer, diagnostic evidence layer and the manoeuvre layer) and four different coloured circles; causal evidence (CE), diagnostic evidence (DE), helper nodes (HN) and manoeuvre nodes (MN)) [37].
Each layer and node have their purpose. First, the causal evidence layer determines the requirements for a specific manoeuvre by checking if there is, for example, a turn existence

to the left or right [37]. Secondly, the diagnostic evidence layer models the consequence of a specific manoeuvre by their measurable physical states [37]. Finally, the manoeuvre layer will determine possible manoeuvres based on the given evidence [37]. The helper nodes are introduced to simplify the parametrization by using lateral and longitudinal motion [36]. The network will use the CE, DE and HN to determine if a particular manoeuvre in the MN layer is possible, based on the gathered evidence. It will output a set of possible manoeuvres from the Maneuver Node layer, determining the optimal trajectory for each possible manoeuvre in Section 2-4.

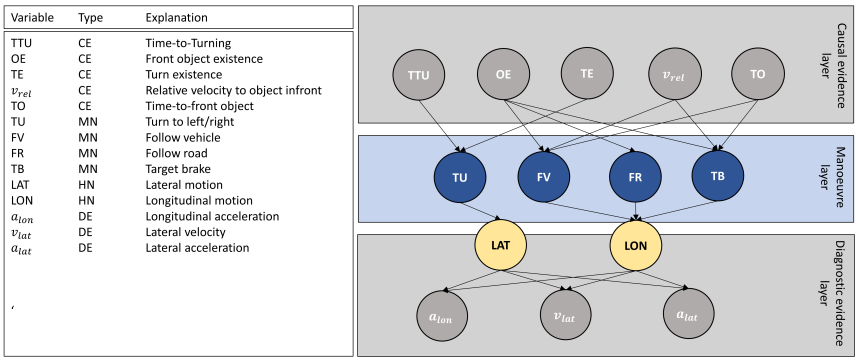

Figure 2-3: An overview of the visual representation of the Bayesian network inspired by [37]

.

2-2-3 Predefined set of route possibilities

Besides models determining the route possibilities, many papers consider the route information as given because the vehicle is only tested in a small set of traffic scenarios. For instance, in [8, 12, 12, 17, 18, 21, 23, 24, 30] the possible manoeuvres of surrounding vehicles are given. A predefined set of manoeuvres is most common for highway driving scenarios. Another method to gather the route information is considered in [43, 44], where the vehicle route possibilities are compared with GPS data. In [48] the motion prediction is applied on a non-structured road network. Therefore, a set of possible manoeuvres is not defined because the vehicle is free to move in any direction.

2-3 Block 3: Interaction determination

The block interaction determination is only used in interaction-aware motion models, as shown in Figure 1-5. The interaction model no longer describes a vehicle as an individual moving object but as an object with dependencies with other traffic participants [28]. Therefore the vehicles' motion is assumed to be influenced by the action of different actors in the traffic scene. Besides incorporating the dependencies between individual vehicles, resulting in a better understanding of the traffic scene and more reliable predictions [28]. Despite this, there are only a few interaction-aware motion models in the literature [28]. This chapter will mention how different interaction-aware motion models tackle the interaction between the individual vehicles, starting with an Occupancy Grid Map, followed by two models using the conflict area differently.

2-3-1 Occupancy grid map

This section will give a brief explanation of the principle of the Occupancy Grid Map (OGM) found in the literature ([6, 15, 21, 32, 42]). An OGM presents the space around the ego-vehicle as individual cells which are free or occupied. The input sensor data determines the state of the cells. The OGM mainly focuses on static objects and therefore performs insufficient in dynamic environments. Planning based on the OGM results in uncomfortable planning requiring frequent correction [31]. The insufficient path planning is due to poor object detection in a dynamic environment. To ensure the proper object detection Dynamic Occupancy Grid Maps (DOGMs) are introduced, representing each cell with their occupancy information combined with their velocity estimates. So each cell contains the probabilistic occupancy, and velocity estimates [11]. Proper object detection needs to be present to produce a sufficient grid map to create DOGMs [46]. A DOGM is created in two steps; First, a classical occupancy grid map is created based on the sensor data. Second, the spatial occupancy and velocity distribution are estimated by a filter for each cell independently. The output of the DOGMs is applied to neural networks, mentioned in Section 2-5, as shown in [3, 11, 46**?**]. Two advantages of the DOGM avoids the feeding of raw data into the network: First, DOGM provides a spatial occupancy and velocity distribution based on a velocity covariance matrix for each cell. Second, the output of a DOGM is not dependent on the type of sensors and their setup.

2-3-2 Conflict area

Besides the occupancy filter, the most popular method for determining any interaction between the vehicles is by constructing the possible conflict area between the individual cars used in [17, 40]. By comparing each vehicle's set of possible manoeuvres, the conflict area between the vehicles can be specified. Identifying specific routes intersecting with each other is based on their centerlines or polygon shapes. The choice between the centerlines and the polygon shapes depends on the available data. By choosing the polygon intersection, the model will better understand when inevitable conflict may occur because it will determine when a lane conflicts with another lane. Whereas the centerline only determines the conflict between the single centerlines. Figure 2-4 shows a visual representation of both methods in finding these conflict areas for a 4-way intersection, where yellow shows the conflict areas. As shown in the figure, the only non-conflict area occurs when vehicle A will turn right. Otherwise, there will always be some conflict between vehicle A & B.

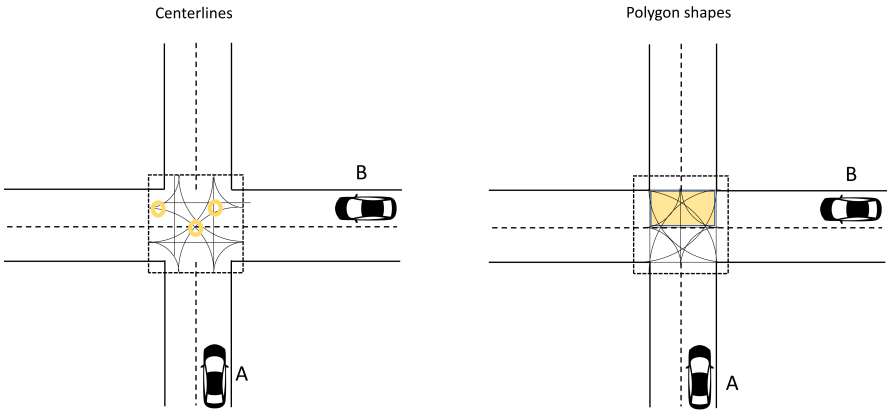

Figure 2-4: a) The left image shows the centerline method. b) The right image shows the polygon method. The conflict areas are shown in yellow

Once the conflict area is determined, the interaction between the vehicles can be determined using one of the following methods. The formation tree describes a traffic scenario utilising the formation of the vehicles in the traffic situation and the expansion by building a formation tree [39]. While in [17] the interaction is determined by assuming a constant velocity for the surrounding vehicles in the traffic scenario.

Formation tree

Once the conflict areas are known, the vehicles' formation at the conflict area is constructed. The formation of the traffic participants is built based on the distance between the cars or the vehicle's distance to the intersection [40]. Figure 2-5 shows an example of such a formation in a highway-like scenario.

Figure 2-5: Determination of the formation in an intersection scenario

Once the current formation is established, the next step is determining the formation tree. The concept of the formation tree is explained by addressing the highway-like scenario from Figure 2-5. The formation is constructed by expanding the initial formation based on the three options for each vehicle; go left, go straight and go right. *"These actions do not represent continuous motion but changes in the relative order"* [39]. The expansion of the formation is checked for their legitimacy following the trimming conditions mentioned in [39]. The expansion repeats itself till the termination formation conditions are satisfied. The two terminate formation conditions are; all vehicles with different driving directions have passed each other, and all vehicles are on a lane with their correct driving direction [39]. As can be seen in Figure 2-6, the expansion using the trimming conditions resulted in three possible termination

manoeuvres: F_1, vehicle B will stay behind A. F_2, vehicle B will overtake A when vehicle C has passed. F_3, vehicle B will overtake A before vehicle C passes.

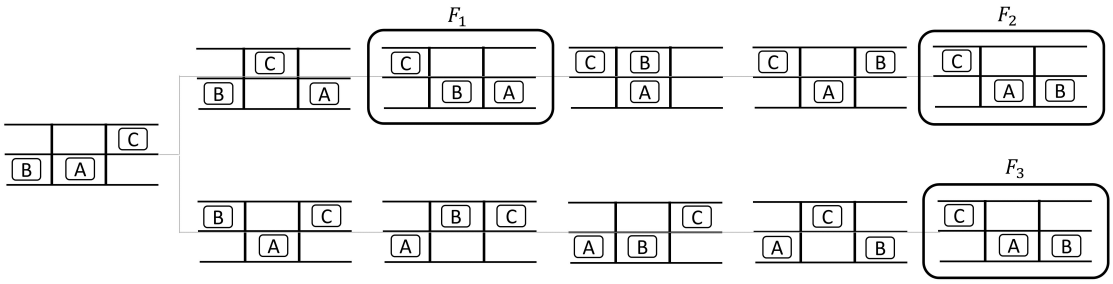

Figure 2-6: Full formation tree for a highway scenario [39]

Constant velocity

The last method for considering the interaction between vehicles is considered in [17, 18], where the possible conflict with other cars influences the ego vehicle's motion. The model only determines the conflict of the ego vehicle route with the possible routes of the surrounding vehicles. Figure 2-7 shows the possible conflict points between the ego vehicle and vehicle 1. The only data gathered from the other vehicles is the distance to the conflict areas and the current velocity. So for the scenario of Figure 2-7 the surrounding vehicle has two distances to the two conflict areas (A & B). The arrival times of car one are calculated assuming a constant velocity. Therefore the arrival time for car 1 at the conflict points is $arrivaltime = \frac{distancetoconflict}{currentvelocity}$. Based on the known arrival time of the ego vehicle itself at these conflict point, the ego vehicle can determine if the vehicles will conflict or not. If the vehicles do conflict, the trajectory of the ego vehicle will be adjusted. Section 2-4 describes how the potential conflict areas influence the final trajectory of the ego vehicle.

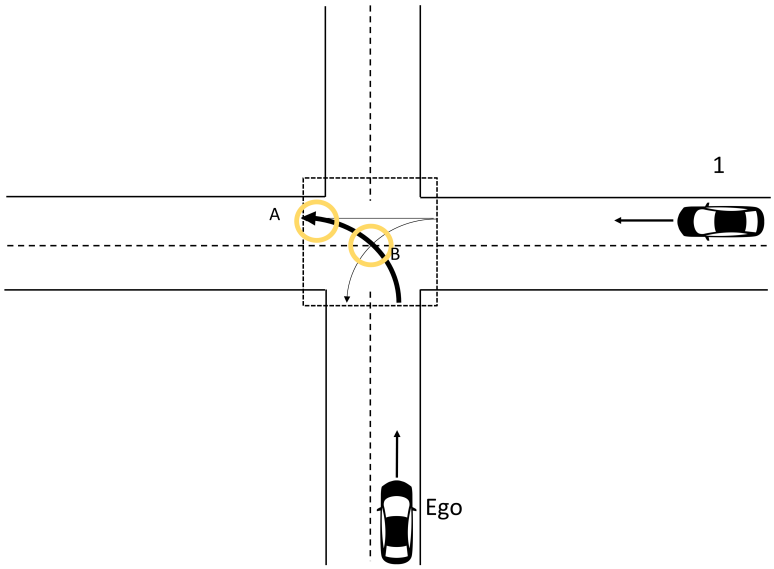

Figure 2-7: Visual representation of a scenario where the constant velocity model is applied

2-4 Block 4: Trajectory planning

Trajectory prediction is used to create a trajectory for each possible manoeuvre. The possible manoeuvres of each car were determined in the previous Sections 2-2 and 2-3. The output of these sections is a set of possible manoeuvres without any information on how a specific manoeuvre is executed. Trajectory planning is a vast and complicated subject. To restrict the search area for the trajectory planner, the planner needs to be quick, smooth and does not have to be very accurate because the planner only makes a predicted path. By keeping this constraint in mind, the following trajectory planners are found in the literature; At first, the Constant Yaw Rate and Acceleration model [14], where the future position of a vehicle is calculated by assuming that the yaw rate and acceleration remain constant during an iteration, while not considering the infrastructure layout. Secondly, creating a path based on the longitudinal and lateral jerk optimisation in the Frènet-Frame [51]. Followed by the methods using the principle of separating the longitudinal and lateral components to determine the total cost function by optimising the cost function based on the control inputs and deviation to the desired velocities and position [39]. Fourth, the path-velocity decomposition (PVD) is explained [16]. The PVD assumes a vehicle will follow the centerline and determine the velocity profile. Finally, a constrained acceleration model is introduced, determining the acceleration boundaries at each iteration, assuming the vehicle will follow the centerline [40]. The output is combined to determine the probabilities between the manoeuvres of each actor.

2-4-1 Constant yaw rate and acceleration

Constant yaw rate and acceleration (CYRA, also known as CTRA) is used in physics-based motion models [14]. Therefore the model does not consider the map information for the motion prediction. So the vehicle is represented as a dynamic object, using dynamic and kinematic models. *"The prediction is completed by linking the control inputs (e.g. yaw rate, acceleration), car properties (e.g. length of the vehicle) and external conditions (e.g. friction with the road surface) to predict the future state of the vehicle (e.g. velocity, heading and position)"*[28]. The future state of the vehicle is determined as follows, with the state vector described as $\vec{x}(t) = \begin{pmatrix} x & y & \theta & v & a & w \end{pmatrix}^T$, with x & y being the world coordinates, θ the heading, v the velocity, a the acceleration, ω the yaw rate and T the time step. These state variables determine the future position in $x-$ & $y-$coordinates as follows; $\Delta x(T) = \frac{1}{\omega^2}[(v(t)\omega + a\omega T)\sin(\theta(t) + \omega T) + a\cos(\theta(t) + \omega T) - v(t)\omega\sin\theta(t) - a\cos\theta(t)]$ and $\Delta y(T) = \frac{1}{\omega^2}[(-v(t)\omega - a\omega T)\cos(\theta(t) + \omega T) + a\sin(\theta(t) + \omega T) + v(t)\omega\cos\theta(t) - a\sin\theta(t)]$ [13, 14]. As shown in the equations, does the CYRA prediction model assume that the yaw rate and acceleration of the vehicle remain constant during one time step T. Due to the variables staying constant, the CYRA model is only suitable for physics-based motion models.

2-4-2 Trajectory planning in the Frenét-Frame

A trajectory planner which can quickly calculate a consistent, smooth trajectory (no overshoot) within a short time is crucial for a motion prediction model. A smooth trajectory needs to be jerk optimal because the jerk directly affects the comfort of a trajectory. In [45] was determined that a quintic polynomial will create a jerk-optimal connection between the start

and end state of a certain trajectory, the polynomial is $x(t) = a_0+a_1t+a_2t^2+a_3t^3+a_4t^4+a_5t^5$ and $y(t) = b_0 + b_1t + b_2t^2 + b_3t^3 + b_4t^2 + b_5t^5$. Based on the quintic polynomials, the papers [50, 51] introduced a new method for determining a trajectory within a short iteration time. The method will determine the optimal trajectory in the Frènet-Frame (shown in Figure 2-8). A Frenét-Frame is a special form of the Euclidean group. Instead of using the Cartesian coordinates for formulating the trajectory, the model will change to a dynamic reference frame and generates a one-dimensional trajectory for both the longitudinal and lateral position [33]. As a result, the Frènet-Frame assumes that the centerline of a lane is the reference line. The Frènet-frame is chosen because a human driver will also plan its trajectory using the centerline as a reference. The optimal trajectory in the Frènet-Frame is based on the separation of the coordinate system to the lateral (s) and longitudinal (d) coordinates, shown in Figure 2-8. The figure shows that the s component is focused on the longitudinal direction while d will reduce the lateral offset to the centerline of a lane.

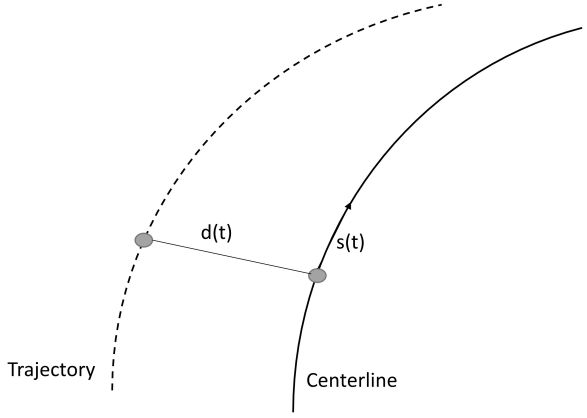

Figure 2-8: Trajectory in the Frenét-Frame, with d being the lateral deviation and s the longitudinal position on the centerline perpendicular to the trajectory line

The lateral component of each trajectory is calculated separately from the longitudinal component to ensure a quick calculation time. The lateral part (d) is calculated with a quintic polynomial, while the longitudinal component (s) is calculated using a quartic polynomial. The quintic polynomial will determine a trajectory to reduce the lateral offset to zero ($d = 0$) or when the vehicle is parallel to the centerline. The longitudinal component (s) of the trajectory is calculated using a quartic polynomial. It can be assumed that the lateral segment is constant (the vehicle is parallel or on the centerline of the road). The model will find the jerk optimal trajectory for the longitudinal and lateral components, constrained by time. Several methods can be found in the literature to determine the local optimum between these trajectories. This section will introduce four methods for determining the optimal trajectory based on the optimisation in the Frènet-Frame described above. These four methods were chosen because they can create a smooth trajectory in a short computation time, which is crucial for trajectory planning. At first, optimization by splitting the lateral and longitudinal trajectories. Secondly, by using the acceleration to determine the lowest cost trajectory. Hereafter, the path-velocity decomposition method assuming $d = 0$, resulting in a one-dimensional trajectory optimization problem. Finally, the acceleration model computes the acceleration boundaries, assuming the vehicle will follow the centerline.

Lateral and longitudinal trajectory optimization

This section will explain the lateral and longitudinal trajectory optimisation method using a weighted sum on the separate cost functions [51]. The cost function for the lateral movement is defined as $C_d = k_j J_t(d(t)) + k_t T + k_d d_1^2$, based on J the jerk, T duration time and d the deviation from the desired lateral position (also seen in Figure 2-8). The longitudinal movement is divided into three types of scenarios; following scenario, merging & stopping scenario and velocity keeping scenario. The general cost function for the following, merging & stopping scenarios is defined as $C_t = k_j J_t + k_t T + k_s \left[s_1 - s_d\right]^2$. Each of the two scenarios has its target position s_d for the general cost function. For the following scenario, the target position is a moving point, so the desired position is determined by a minimum safety distance between the leading and following vehicle. While at a merging & stopping scenario, the target position is a specific predefined position. The velocity keeping scenario occurs when the vehicle is driving independently with no direct vehicles ahead. Therefore the car will have a constant desired velocity instead of a target position.
Eventually, the cost function of the longitudinal and lateral component are combined to create the total cost of each trajectory. The total cost is calculated by a weighted sum formula $C_{tot} = k_{lat} C_{lat} + k_{lon} C_{lon}$, where the weight factors need to be chosen for an optimal result. By doing so, the method described in [51] & [50] will find the jerk optimal trajectory in the Frènet-Frame. Also, will the trajectory be consistent and will result in no overshoot. Due to the Frènet-Frame, the model cannot handle high curvatures in the road. Therefore the separation of longitudinal and lateral optimisation in the Frènet-Frame will only be suitable for highway scenarios, where the curvature is low. So the model can handle overtaking, following and stopping scenarios but cannot handle urban scenarios.

Double integrator

This section will explain a trajectory planner using an acceleration based cost function. The states for optimisation are assumed to be the control inputs of the vehicle $u^n = [a_s^n, a_d^n]^T$, where the a_s is longitudinal and a_d lateral acceleration [39]. These variables can be directly influenced by the driver and are therefore chosen as the optimization inputs. The accelerations are limited by the sideslip angle $\beta = \arctan(v_d/v_s)$ due to the nonholonomic properties of a vehicle. By separating the longitudinal and lateral components the cost function for each of these components are based on the deviation of the current state of the vehicle from the reference state; $\Delta x_k^n = x_k^n - x_k^{n,ref}$ (with n a certain maneuver and k an iteration step), with the reference state given as $x^{n,ref} = [s^{n,ref}, v_s^{n,ref}, d^{n,ref}, v_d^{n,ref}]^T$. By calculating the difference between the current state and the reference state, the cost function is as follows for the lateral and longitudinal component; $J_s^n = \sum_{k=1}^{K} \left(\Delta \boldsymbol{x}_{s,k}^n\right)^\top \boldsymbol{Q}_s^n \Delta \boldsymbol{x}_{s,k}^n + \sum_{k=0}^{K-1} u_{s,k}^n R_s u_{s,k}^n$ and $J_d^n = \sum_{k=1}^{K} \left(\Delta \boldsymbol{x}_{d,k}^n\right)^\top \boldsymbol{Q}_d^n \Delta \boldsymbol{x}_{d,k}^n + \sum_{k=0}^{K-1} u_{d,k}^n R_d u_{d,k}^n$ with Q the state weighting matrices $\boldsymbol{Q}_s^n = \mathrm{diag}(0, \omega)$ and $\boldsymbol{Q}_d^n = \mathrm{diag}(\omega, \omega/2)$ and R the control inputs weights $R_s^n = \omega$ and $R_d^n = \omega/2$. Where $\omega = 1 + (\gamma - 1)\rho$ with right of way ($\rho = 1$) or not ($\rho = 0$) and γ how strong the drivers weight the traffic rules. The cost function for the lateral (J_d^n) and longitudinal (J_s^n) component will be optimized by quadratic programming. Eventually the combined cost function is calculated by combining the lateral and longitudinal components $J_{M_i}^* = J_{M_i,s}^* + J_{M_i,d}^*$

Path-velocity decomposition

The path-velocity decomposition (PVD) is used in [16–18] and was introduced in [20], *"in which a specific manoeuvre of a vehicle is calculated in the one-dimensional direction, the longitudinal direction only"* [7]. The (PVD) uses the same principle mentioned in the previous sections, where it calculates its prediction based on the Frènet-Frame. Each vehicle has a set of possible goal positions placed on each possible exit lane centre. Therefore the model assumes that a vehicle will follow the centerline of a lane for each possible manoeuvre, so the lateral deviation is assumed to be zero ($d = 0$). The model will calculate the cost for each velocity profile based on the possible routes. The movement of the vehicle is described by equation 2-1, with the bounded velocity $\dot{s} \in [0, v_{\max}]$ with v_{max} being a function of the curvature and the bounded acceleration $u \in [a_{\min}, a_{\max}]$ known as the input.

$$\begin{bmatrix} \dot{s} \\ \ddot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ \dot{s} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{2-1}$$

Along each path, a finite set of possible events can occupy the future path divided into two classes; dynamic (E_D) or static (E_S) events. A dynamic event occurs when another car crosses its path. The static event can be described as a traffic light or parked car. The vehicle should never collide with these events. In addition, traffic laws are also imposed along the road and limit the input and velocity. The goal of the PVD is to find the local optimum for each manoeuvre. The local optimum is found based on a velocity-dependent cost function. The cost function is a combination of three individual costs. At first, the cost for the desired velocity (c_V). Hereafter, the cost for events (c_E); dynamic (E_D) and static (E_S) events. The cost function for a static event is infinity to prevent the vehicle from colliding with the object. The dynamic event occupies an area for a certain amount of time, know as the temporal-spatial cost map. Both the static and dynamic cost functions are combined to calculate the total cost of events (c_E). Finally, the cost of action (c_A) is added to the total cost function. The cost of action is calculated by quadratically punishing the acceleration of a certain path. By punishing the acceleration quadratically, the driver's comfort is increased because the acceleration directly affects the comfort of a certain trajectory. Combining all the cost functions mentioned above results in $c\left(x_i, a, x_{i+1}, E\right) = c_{\mathrm{V}}\left(x_{i+1}\right) + c_A(a) + c_E\left(x_i, x_{i+1}, E\right)$ will find the local optimum for each path. The PVD has only been used in intersection scenarios, where no overtaking is considered.

Bounded acceleration model

In [40] the action model is given as $\boldsymbol{a} = [a, \dot{\theta}]$, where the acceleration a and heading rate $\dot{\theta}$ are important control parameters. The acceleration is constrained by many factors; vehicle dynamics, speed limit, preceding vehicle, road curvature and conflicting agent [40]. The range of these constraints is given in Table 2-2.

Influence	Features	Action Range
vehicle dynamics	$-$	$[a^{\min}_{\text{vd}}, a^{\max}_{\text{vd}}]$
speed limit	$d_{v_{\text{lim}}}, v_{\text{lim}}, v^i$	$[-\infty, a^{\max}_{\text{IDM}}]$
preceding agent V^p	d^p, v^p, v^i	$[-\infty, a^{\max}_{\text{IDM}}]$
road curvature	d_ρ, ρ, v^i	$[-\infty, a^{\max}_{\text{curve}}]$
conflicting agent V^c	$\chi^{i,c}, d^c_{\text{entry}}, d^c_{\text{exit}}, v^c,$ $d^i_{\text{yield}}, d^i_{\text{entry}}, d^i_{\text{exit}}, v^i$	$[a^{\min}_{\text{conf}}, a^{\max}_{\text{conf}}]$

(2-2)

The trajectory planner described in [40] will determine the predicted acceleration of a vehicle assuming the vehicle is travelling on the centerline of a lane and determining all the constraints from Table 2-2. The maximum acceleration a^{max}_{IDM} is calculated using the intelligent driver model (IDM) introduced in [47]. The IDM formula has several advantages and is used to calculate the maximum acceleration. First, it is accident-free because it is dependent on the relative velocity. Secondly, the model parameters of the IDM can be empirically measured and have an expected magnitude [29]. Finally, the stability can be easily calibrated to the data and allows for fast simulations. The formula for the maximum acceleration is primarily designed for highway scenarios, where the road's curvature is not considered. The allowable velocity needs to be dependent on the road curvature $v_\rho = \sqrt{\rho a^{\max}_{\text{lat}}}$ to ensure that the model is applicable in urban scenarios. By incorporating the velocity constraint the maximum acceleration formula is changed to $a^{\max}_{v_\rho, d_\rho} = \tilde{a} = \frac{-2v + \Delta T b_d + \sqrt{4v\Delta T b_d + \Delta T^2 b_d^2 - 8 b_d d_\rho + 4v_\rho^2}}{2\Delta T}$, with minimum spacing d_d, desired headway time T_d, comfortable acceleration a_d, braking deceleration b_d. Using the centerline information, the model assumes that each vehicle will remain on the centerline of a lane during each traffic scene, resulting in a one-dimensional trajectory planner. Therefore the method is only used in intersection scenarios, where overtaking is not considered. The final acceleration will lay within the acceleration constraints mentioned in Table 2-2, assuming that a vehicle will always leave the traffic scene as fast as possible.

2-5 Block 5: Manoeuvre estimation

Manoeuvre estimation is used to create a probability density for the possible manoeuvres, based on the calculated trajectories in the previous Section 2-4. The manoeuvre probabilities are constructed following specific methods found in the literature. This chapter introduces the most relevant techniques. At first, the interaction multiple model (IMM) will be described, used in [39]. Here the probability of a specific manoeuvre is calculated using multiple filters. Second, the cost function probabilities are explained. The lowest optimal cost is based on a single measurement. The lowest optimal cost gradient is based on the current and previous measurement. Finally, this section introduces two networks; a Bayesian network and a neural network.

2-5-1 Interacting multiple model filter

In the paper, [39] the model uses an interacting multiple model (IMM) estimator to determine the probabilities between the manoeuvres. The IMM determines the future trajectory of vehicles by combining separate filters. An IMM filter contains four steps; Interaction, filtering, update and combination. In the first step, interaction, the initial values received from the sensors will be calculated for the filtering step. The filtering step executes separate filtering for each model to create a prediction. Primarily a Kalman filter is used in this step. Hereafter, the models' probabilities are updated and eventually combined and sent to the interaction step to finish an IMM cycle [41]. The IMM model is widely used in the literature because it incorporates both kinematic and dynamic models of the vehicles [19]. Figure 2-9 will create a better understanding of an IMM filter by addressing the steps shown in the figure. Step one and two separates and filters the measurement data (velocity, turning and acceleration). Hereafter, the model predicts the future state of each separated measurement data. Step four, calculates the combined future state by blending the different predictions based on their model likelihood. Step five, the final combined prediction, is compared with each filter's outputs to determine the covariance of each filter. Finally, the covariance of each filter is used to determine the new likelihood between the filters for the next prediction step. The filter repeats the described steps above at each iteration step.

The IMM estimator takes a long assessment time because it has to wait for the computation of each filter at each time step. The main advantage of the IMM filter is that the estimator allows random changes between models, which matches the behaviour of many real-world driving scenarios.

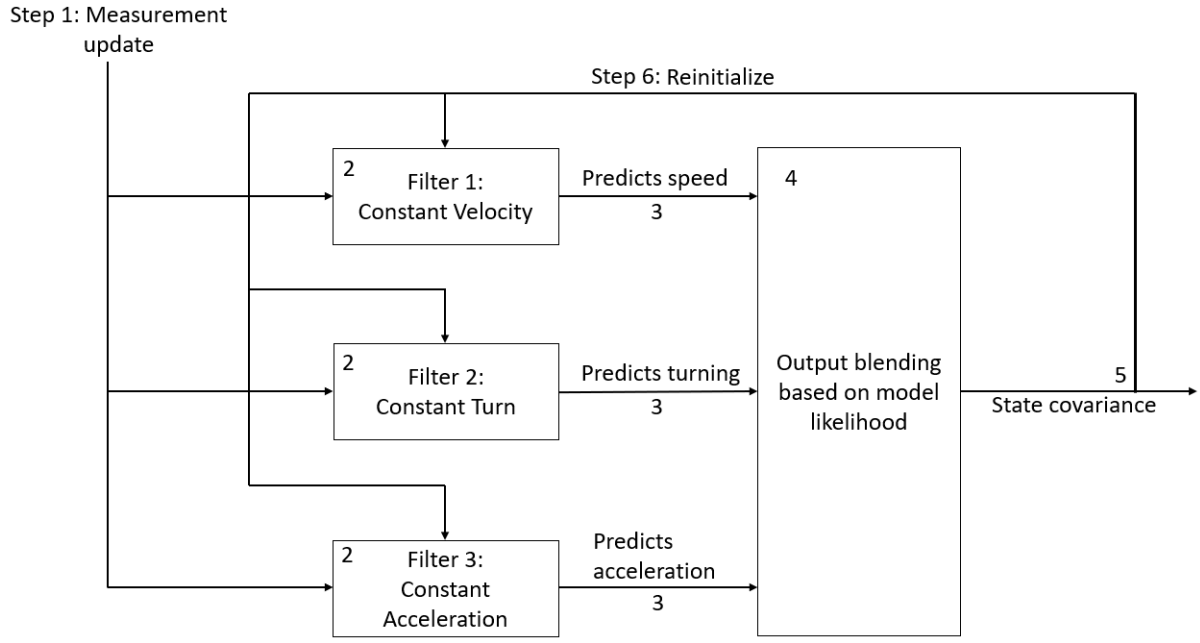

Figure 2-9: Visual representation of each step of the IMM filter

2-5-2 Cost-function probability

Cost-function probability calculation is used in [49], where the cost function of different manoeuvres are compared with each other. The cost functions were determined in the previous building block, mentioned in Section 2-4. The actual probability is calculated by determining the percentage a certain trajectory has compared to the total cost function. The higher the cost function of a certain trajectory, the lower the probability of that specific manoeuvre will be and vice versa. The determination of the probability based on the cost-functions can be used in two ways; lowest optimal cost $J^*_{M_i}(k)$ [39] or lowest optimal cost-gradient $\Delta J^*_{M_i}(k) = J^*_{M_i}(k) - J^*_{M_i}(k-1)$ [39]. The lowest optimal cost gradient produces the probabilities based on two measurements, the current measurement and the previous measurement. Instead of the optimal cost method, only uses the current measurement.

2-5-3 Networks

In the literature, different networks are used to determine the manoeuvre probabilities. These networks can be divided into three concepts; (Dynamic) Bayesian networks (DBN) and Hidden Markov Models (HMM), and neural networks (NN). This section highlights two of these networks: the Bayesian network and the neural networks.

(Dynamic) Bayesian networks

(Dynamic) Bayesian networks (DBN) are broadly used in the literature as mentioned in Section 2-2. There are two types of Bayesian networks; Bayesian network and Dynamic Bayesian network. Dynamic Bayesian networks are used for modelling time series. *"For time series it is assumed that an event can cause another event in the future, but not vice-versa. This assumption simplifies the design of a Bayesian network because all arcs move forward in time"* [9]. This section will explain the Bayesian network of Section 2-2. The probability assessment of the Bayesian network is divided into two environments; dynamic and static environments [37]. In dynamic environments, the general equations [5] are adapted to determine if the ego-vehicle will collide with any vehicle V_i in the prediction horizon. The joint density of the surrounding vehicles configuration for all future time steps needs to be calculated to determine where in the future certain vehicles may collide with each other based on their configuration at a particular time step. The combined density of the vehicles based on the configurations and the independent trajectories of all vehicles are combined. For static environments, the network uses another method for determining the probability of collision. A static environment only contains the ego vehicle as a moving object in the environment. Two methods can determine the free space for the ego vehicle; Occupancy Grid Map (OGM) described in Section 2-3-1 and building a Parametric Free Space (PFS) [35] map representation. The PFS has the main advantage that the current map is shortly outlined, easily extracted, and provides a compact representation of the reachable, drivable free space. The PFS is combined in a so-called B-spline curve. An example of a PFS map with the corresponding B-spline curve is shown in Figure 2-10, with the red lines being boundaries and the yellow lines are unknown boundaries. The PFS will remain constant during the prediction horizon if the static environment does not change.

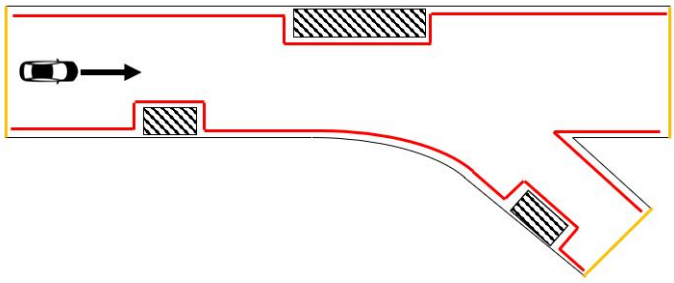

Figure 2-10: The red line shows the obstacle boundaries of the surrounding environment, the orange lines show the unknown boundaries [35]

A probability function is created to determine the probability of the ego vehicle colliding with the static map. The probability function will check if the ego vehicles' rectangle shape will intersect with the B-spline curve. Eventually, the total probability of the ego vehicle colliding with a static or dynamic object will be calculated by combining both probability functions. To determine each of the individual probabilities for the dynamic and static environments, the network uses a Monte Carlo simulation. The probability will be used to determine the $TTCCP = \min(T_P(C_k(T_P)) > CCP) \cdot T$ (Time-To-Critical-Collision-Probability), with CCP being the Critical Collision Probability. This Bayesian network mainly focuses on determining the probability of the ego vehicle colliding with a dynamic or static object.

Neural networks

The literature also provides several examples using a neural network instead of a Bayesian network. The papers using a neural network are [4, 8, 15]. A neural network is built to ensure a network that works similar to the human brain. A neural network uses large data sets to train the human-like network. A neural network is constructed by three layers; the input layer, the hidden layer and the output layer. Figure 2-11 shows a visual representation of such a network. The first step of a neural network is to transition and process the input data so the network can efficiently process it. When the data is sent from the input layer to the hidden layer, certain weights ($w_{1,2,3,4}$) are added to the input values. Once the weights are added to the input data, an activation function will determine if a particular bias (b) should be activated. The biases are added to the multiplication of the input data and the weights. Finally, the data will be sent to the output, where the probability between the manoeuvres will be given. By comparing the output of the network and the actual output the network checks if the neural network is working correctly. If the network does not operate as it is supposed to, the network will use backpropagation to adjust the biases and weights of the links to improve the network and minimize the error.

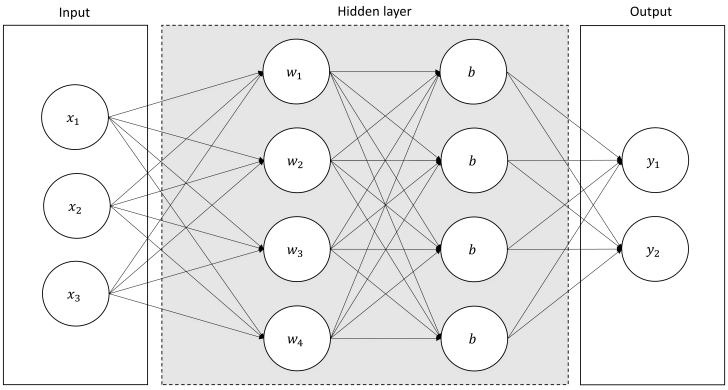

Figure 2-11: A visual representation of a neural network

2-6 Summary

This chapter briefly explained several methods found in the literature for each building block. As mentioned in Section 1-2, the interaction-aware motion model contains five building blocks, and therefore each of these blocks was mentioned in this chapter to make a considered choice in the next chapter.
At first, the measurements block. This block determines and collects the state information of the surrounding vehicles and the ego vehicle for the motion prediction model. The mainly used state information are position and velocity. Additionally, the acceleration and yaw rate are added in some papers.
Block two gathers the route possibilities of the surrounding vehicles based on their current lane position. The breadth-first method determines the future lanes by the successor lane information of each lane in the scenario. The networks will also find the route possibilities based on the current lane. Other methods in the literature use a predefined set of possibilities for each vehicle.
In the third block the model determines the possible conflict points and interaction in the block interaction determination based on the route possibilities. This block contains two main methods; occupancy grid map and conflict areas. The occupancy map divides a map into cells, where each cell holds a probability of being occupied or free. The conflict areas method is divided into two approaches; the formation tree and the constant velocity model. The formation tree builds a tree based on all possible order changes for the vehicles in traffic scenario. The constant velocity model assumes that a vehicle will continue on the centerline with a constant velocity till the conflict point. Once the data is gathered in block one, the route possibilities in block two, and the interaction determination in block three, the data from these blocks are combined in the block trajectory planning.
Trajectory planning is a vast subject in the literature, so only the methods that can rapidly create a smooth trajectory are mentioned. First, the constant yaw rate and acceleration model is introduced, which only works for physics-based motion models. It assumes constant acceleration and yaw rate during a time step and does not consider the infrastructure. Secondly, the trajectory planning in the Frenét-Frame contains four different methods using this principle. At first, the lateral and longitudinal separated trajectory optimization. The separation ensures a smooth trajectory within a short processing time. Hereafter, the double integrator method optimizes the cost of acceleration of the trajectory because a human driver directly affects the acceleration of a vehicle. The path-velocity decomposition assumes the vehicle will drive on the centerline of a lane, resulting in a one-dimensional problem. The trajectory is determined by finding the lowest cost for avoiding static and dynamic obstacles. At last, the bounded acceleration model determines the trajectory of the ego vehicle based on five aspects that bound the vehicle's acceleration.
Eventually, block five, manoeuvre estimation, determines the probability between the several trajectories of each vehicle. The literature introduces several methods for determining these probabilities, integrating multiple model filter, cost function probability, and networks. The IMM filter uses separate filters for each vehicle state and updates the filter based on their covariance. The cost function probability determines the cost of a trajectory compared to the cost of other trajectories to find the probability of a specific trajectory. The networks are split into Bayesian networks and neural networks. The networks are trained on big data sets to determine their weights and biases.

Chapter 3

Chosen model

Since a motion prediction model is built using the building blocks shown in Figure 3-1, this chapter will conclude which method should be chosen for each building block. To do so, first, the level of motion prediction have to be selected. The choice between the three levels of motion prediction models, the physics-based motion model, the manoeuvre-based motion model and the interaction-aware motion model, will determine the number of building blocks considered in this chapter. To choose the level of motion prediction, the application in which 2getthere will apply the motion prediction model is essential.

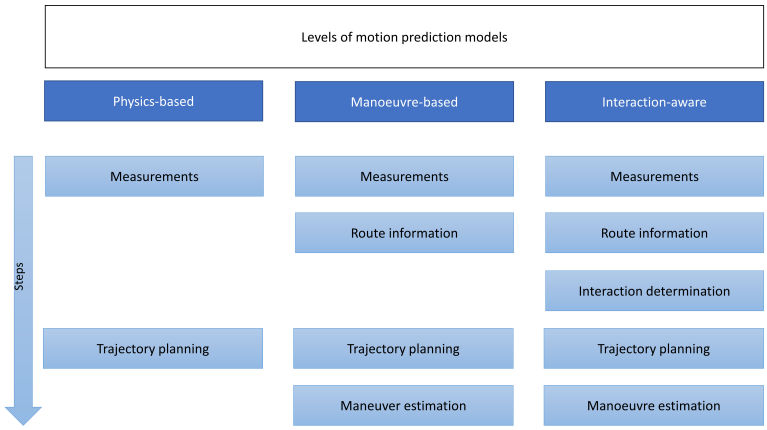

Figure 3-1: Overview of the building blocks in motion prediction

Currently, the GRT of 2getthere is driving on a segregated infrastructure. Still, they aim to operate their vehicle in mixed traffic scenarios. Therefore, the GRT must create a collision-free smooth path. The prediction model will predict the motion of the vehicles up to 2 seconds, because for this horizon the prediction will be accurate and the GRT can still react on it. To make such a prediction, the GRT has to consider the infrastructure (e.g. layout, traffic lights and traffic rules) and the movement of the surrounding vehicles. The infrastructure data will be provided using OpenDrive [1] and converted to a lanelet format [2, 34], where the lanes will be constructed as polygons with lane properties and successor & predecessor lane information. The surrounding vehicles will be detected using the sensors on the GRT itself. These sensors can determine the velocity (absolute velocity and direction of the velocity) and the position with respect to the GRT. The physics-based motion models will not be considered because a

physics-based motion model is only suitable for short time horizons (up to 1 second) and has a slow adaption rate. Therefore, a decision must be made between a manoeuvre-based motion model or an interaction-aware motion model. These motion models are very similar to each other, besides the fact that an interaction-aware model extends a manoeuvre-based model by adding the interactions between the individual vehicles resulting in a better understanding of the traffic situation, more reliable and improved motion prediction [28]. In [40] both motion models are compared based on two intersection scenarios. This comparison concludes that the interaction-aware motion model outperforms the manoeuvre-based motion model when specific conflicts occur. The interaction model has a less fluctuating result and has a minor error in these conflict scenarios. Therefore should the motion prediction consider all vehicles in the traffic scenario to create a reliable prediction. In everyday traffic, the movement of one vehicle will influence the behaviour of another vehicle. Therefore, an interaction-aware motion model is the best choice for this problem statement. The choice for the interaction motion model results in a choice for all blocks shown in Figure 3-2. The chosen models for each block are influenced by limitations of 2getthere (Section 1-3). Therefore, in some cases, not the best method is selected, but the best choice for the 2getthere application.

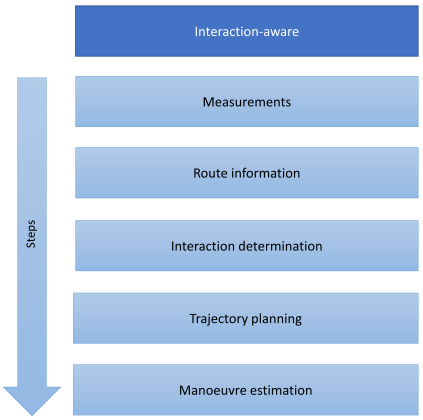

Figure 3-2: Building blocks used for an interaction-aware motion prediction model

This Chapter will address each of the building blocks one by one. For each building block, the choice for a specific method is justified by discussing the several methods and making the best choice for the 2getthere application. Hereafter, an explanation of the chosen method for each building block is given. At the end, this chapter will be summarized.

3-1 Block 1: Measurements

This section will address the choices made for building block 1, measurements. A choice is made on the variables used for creating the motion model and how the model collects these variables. It is essential to determine which state variables will be available at 2getthere. The choices are limited to the limitations of 2getthere as mentioned in Section 1-3.

3-1-1 Method choice

At 2getthere, the state of surrounding vehicles is determined using sensors on the GRT. The GRT is equipped with short- and long-range radars, cameras and LiDARs creating a field of view shown in Figure 3-3. The figure shows that the GRT has a 360-degree close-range coverage (40 - 60 meters) and long-range radars at the corners, front and back of the GRT. The LiDAR and cameras are placed on the roof of the GRT and will create an additional short-range coverage. Due to the use of radar for the long-range sensors, the GRT will only be able to gather the following state information of a surrounding vehicle:

- Position, x- & y-coordinates, with respect to the GRT.
- Velocity, the magnitude of the velocity, in m/s.
- Heading, is found due to the nonholonomic properties of a vehicle, in rad.

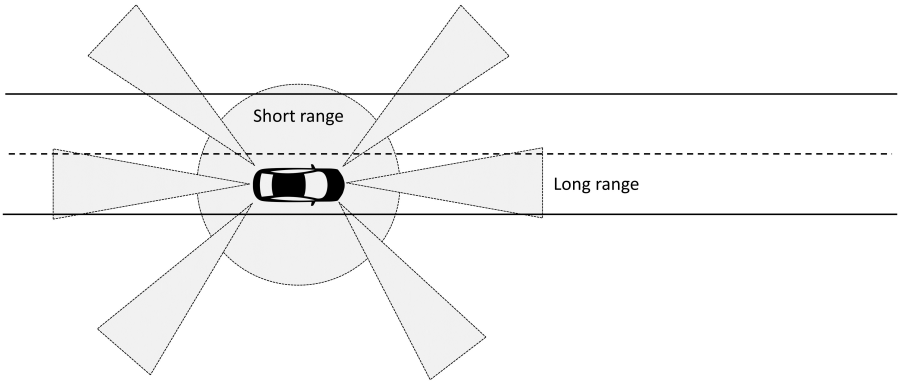

Figure 3-3: The radar layout on the GRT, with the long and short range sensors.

3-1-2 Detailed description

This thesis is focused on creating a motion prediction model for the surrounding vehicles of the GRT. Therefore the state variables of the surrounding vehicle are considered to be given. It is only essential to establish which state variables will be available. The available state information is given in the section above. So the position, velocity, and heading of the surrounding vehicles are given in world coordinates, and the GRT has acceleration as an additional state input. It would be preferred to have the acceleration of the surrounding vehicles because this will result in a better performing motion model. Nevertheless, this is currently not considered in the model.

3-2 Block 2: Route possibilities

This section is dedicated to block number two; route possibilities. Due to the choice for an interaction-aware motion model, the model needs to know the route possibilities of the ego vehicle and the surrounding vehicles to determine their interaction in block three. The choice for the method for determining the route possibilities is explained in the next section, followed by a detailed description of the method.

3-2-1 Method choice

Several methods for determining route possibilities were introduced in Section 2-2; breadth-first method, Bayesian/neural networks and a predefined set of given route possibilities. The available map data at 2getthere will influence the choice for the selected model. 2getthere will supply the map data in a lanelet format [2, 34]. The lanelet structure defines each lanes' shape as a polygon. Besides holding information about the shape of a lanelet, it will also contain information about the driving direction and the connecting lanelets. The connection lanelets contain two categories; predecessors and successors. The names suggest a predecessor, already passed, while a successor is a possible future road section. The breadth-first method is less complex to implement than a network, and the start transition parameters of a network do not have to be determined beforehand. Therefore the most significant advantage of the breadth-first method is that it does not need data sets for training and construction. So, therefore, the breadth-first method is chosen as the desired method.

3-2-2 Detailed description

Figure 3-4 shows an example of a vehicle approaching a T-junction. It shows that it finds the next possible successor lane(s) for each step in the horizon based on a vehicle's current position, which was determined in the previous building block. The position of a vehicle will determine on which lane a vehicle is driving. Due to the lanelet format, each lane has information about its successors for that specific lane. Figure 3-4 found the current lanelet in the left graph. Secondly, the breadth-first method will determine the possible connection lanes on the intersection based on the current lane of the vehicle, resulting in two options; turn right or left. Finally, the final road sections, also known as goal lanes, correspond to the two route options. Eventually, combining these results will result in a total of two possible manoeuvres up to the prediction horizon (l_H) for this specific vehicle; a right or left turn.

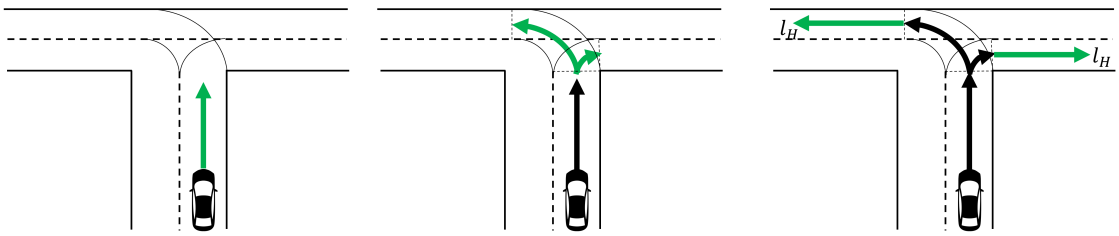

Figure 3-4: The breadth-first method used at a T-junction, with the possible routes shown in green up to the specified horizon l_H. [40]

3-3 Block 3: Interaction determination

The block of interaction determination is also considered in the block choices by selecting an interaction-aware motion model. Interaction determination is essential for creating a realistic and reliable motion prediction. It will create a motion model closest to human driving behaviour. This accurate and reliable motion prediction originates from incorporating the motion of the surrounding vehicles. Therefore the model can adapt quickly to the changing environment, resulting in a better understanding of the traffic scenario. The following section describes the chosen model, followed by a detailed description of the chosen method.

3-3-1 Method choice

As mentioned in Section 2-3 does the interaction determination block no longer describe a vehicle as an individual moving object, but as an object which has dependencies with other traffic participants [28]. Although interaction-aware motion models are not broadly used in the literature, there were three methods found in the literature for taking these dependencies into account; conflict areas using a formation tree, conflict areas using constant velocity and the occupancy of a specific lane. The formation tree focuses on the dependencies between the surrounding vehicles, while the constant velocity and occupancy filter are focused on predicting the ego-vehicle motion based on a possible conflict. The occupancy filter only indicates if a certain lane is accessible based on the occupancy of that lane. In comparison, the constant velocity model assumes a constant velocity for the surrounding vehicle until the intersection point. This literature study aims to find the best suitable motion prediction model for the surrounding vehicles of the GRT. Therefore the formation tree is the best option compared to the other two methods, although the formation tree is the most extensive method of the three.

3-3-2 Detailed description

This section will elaborate on the chosen method for determining the interaction between the vehicles in the traffic scenario. The chosen model will work on the same concept as the formation tree described in section 2-3. The first step of the model is to gather all the previous information; the state information, and the route possibilities of each vehicle in the current traffic scenario. The ego vehicles' route options are also considered because they will influence the behaviour of the other vehicles in the traffic scenario as well. Figure 3-5 shows the possible route options of car A & B. Car A can go left or right, and car B can either go straight or left. The second step is to determine the possible conflict zones between the individual route possibilities based on that information. The conflict areas between the vehicles in the traffic scene can be found by utilizing the lanelet format. The lanelet format presents a lane as a polygon. The polygon shape of each possible route is used to determine intersections between the individual route polygons of the vehicles in traffic scenario. Figure 3-5 two conflict zones were found by finding the intersection of the polygon shapes of both vehicles. The intersection between the shapes was found for two routes. In the yellow zone the conflict zone between car A & B is found. If car A is turning left, and car B is turning left, and when car A turns left, and car B goes straight. If car A takes a right turn, there is

no intersection between the right turn and the polygon shapes for the route possibilities of vehicle B.

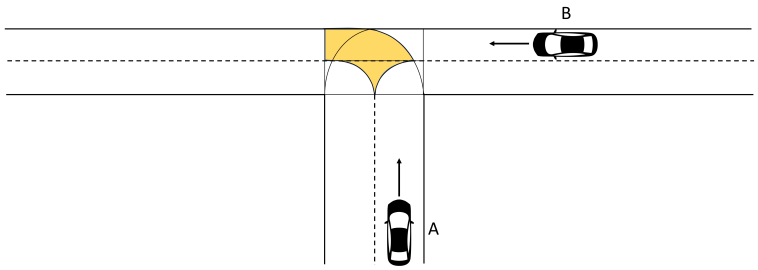

Figure 3-5: The conflict zone between vehicle A & B, based on the polyshapes of the lanes

Once the conflict zones are established, the model will built a formation tree for the specific traffic scenario. The manoeuvre tree for the T-junction scenario is shown in Figure 3-6 with all possible routes and a branch for each possible combination of routes. The manoeuvre tree is built by assuming that a manoeuvre passes the conflict area. Therefore, each vehicle in this scenario has two route options, resulting in a manoeuvre tree with four branches, with two branches (branch 1 & branch 2) representing the conflict areas. If such a conflict area is found, the model determines all possible passing orders of the conflict area for that branch. The possible passing orders for the T-junction scenario are also shown on the right of Figure 3-6.

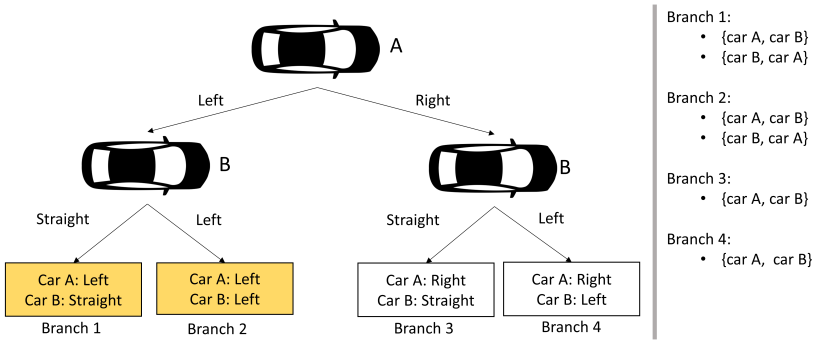

Figure 3-6: The manoeuvre tree shown on the left and the corresponding branches shown on the right for the T-junction scenario

The figure shows that for branch 3 & 4 there is only one manoeuvre due to the lack of conflict between the vehicles if they will execute these routes. Therefore, branch 1 & 2 have two possible options; first car A or B, followed by the other vehicle. The passing order determines which vehicle will have to adapt its motion to the other vehicle if any conflict is occurring. In Figure 3-7 a visual representation of the way the branches are constructed using the principle of the formation tree, mentioned in Section 2-3. So if there is no conflict area between the manoeuvres in the traffic scenario, neither vehicle has to adapt its trajectory. The calculation

of a possible conflict using the trajectories will be discussed in the next section. Due to the concept of reviewing all possible passing orders for a traffic scenario, the model can be applied in any traffic situation. Currently, the traffic rules are not considered because they will limit the possible passing orders. The traffic rules could be considered in the model as a constraint or added as an additional penalty on the cost function.

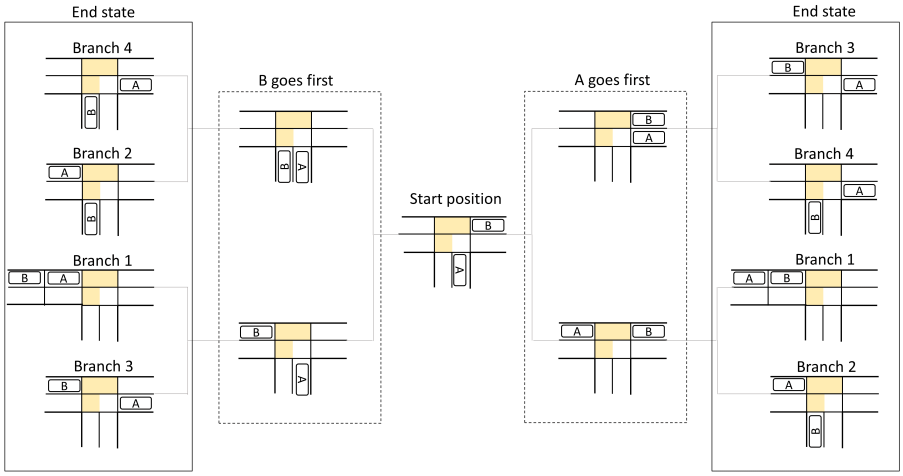

Figure 3-7: The formation tree based on the T-junction scenario. The start situation is shown in the middle. The scenario progresses towards the left and right depending on the vehicle that passes the conflict area first. The final states are shown on the left and right hand sides.

3-4 Block 4: Trajectory planning

This section will discuss the fourth building block, trajectory planning. Trajectory planning calculates the predicted trajectory for each manoeuvre of every vehicle in the traffic scenario found in Section 3-3. The calculation of a trajectory is a vast subject in the literature, as shown in Section 2-4. The following section will describe the best method for the 2getthere application by comparing the literature. Hereafter, the chosen method will be broadly explained.

3-4-1 Method choice

This section will give a brief discussion on the chosen method. The choice is based on the methods introduced in Section 2-4. At first, the CYRA model is straightforward to implement compared to the other methods. However, the model does not consider the infrastructure layout and is only suitable for short prediction horizons (up to 1 second). Therefore the CYRA model is unsuitable for 2getthere because a long prediction horizon is desired while considering the infrastructure. Section 2-4 mentioned several methods using the principle of trajectory planning in the Frènet-Frame. The advantage of separating the longitudinal and lateral motion in the Frènet-Frame is that it assures that the trajectory can be calculated in a short processing time with temporal consistency, with no overshoot if the vehicle will perfectly

follow the trajectory. Due to the separation in the Frènet-Frame, the model is sensitive to curvature in the road segment if there is lateral deviation. The sensitivity to road curvatures is not desired in intersection scenarios because a high curvature is typical in intersection scenarios. Therefore the trajectory planner with lateral and longitudinal optimisation in the Frènet-Frame should be used for highway scenarios. This thesis assumes that a vehicle will follow the centerline of a lane when approaching and driving at the intersection. Therefore the lateral deviation is zero, resulting in a one-dimensional trajectory planner. Section 2-4 mentioned two methods for one-dimensional trajectory planning; path-velocity decomposition and bounded acceleration model. The sensors at 2getthere cannot measure the acceleration of the surrounding vehicles and therefore is the path-velocity decomposition the desired method. The path-velocity decomposition will find the optimal one-dimensional trajectory for each vehicle by using the principle of the quintic polynomial to ensure a jerk optimal velocity profile. The following subsection will explain how the method is used in the model.

3-4-2 Detailed description

This section will elaborate on the chosen method for the trajectory planning. As input, the model uses all previous building blocks: measurements, route possibilities and interaction determination. The model uses the principle of trajectory optimization in the Frènet-Frame by assuming a vehicle will follow the centerline of a lane (lateral deviation is zero) [51]. The trajectory is calculated from the initial state $F_0 = (s_0, \dot{s}_0, \ddot{s}_0)$ to the final state $F_1 = (s_1, \dot{s}_1, \ddot{s}_1)$ in the Frènet-Frame, where s is the longitudinal position, $\dot{s}$ the longitudinal velocity and $\ddot{s}$ the longitudinal acceleration [13, 14]. The jerk optimal trajectory is found using a quintic polynomial, guaranteeing a jerk continuous unique solution [45, 50, 52]. There are four types of trajectories that need to be calculated for each vehicle; no conflict trajectory, merging behind trajectory, merging in front trajectory, and the vehicle following trajectory. Each of the four trajectories is solely based on the polynomial for position s, because lateral deviation is zero. The polynomial is defined as;$x(t) = c_5 * t^5 + c_4 * t^4 + c_3 * t^3 + c_2 * t^2 + c_1 * t + c_0$. The coefficients are obtained by solving Equation 3-1. The type of trajectory will influence the given state information $(s_0, \dot{s}_0, \ddot{s}_0, s_1, \dot{s}_1, \ddot{s}_1)$. The given state information will be mentioned in the sections below in order to solve the coefficients, with different states for each type of trajectory.

$$\begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0^1 & 1 \\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1^1 & 1 \\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0^1 & 1 & 0 \\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1^1 & 1 & 0 \\ 20t_0^3 & 12t_0^2 & 6t_0^1 & 2 & 0 & 0 \\ 20t_1^3 & 12t_1^2 & 6t_1^1 & 2 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \dot{s}_0 \\ \dot{s}_1 \\ \ddot{s}_0 \\ \ddot{s}_1 \end{bmatrix} \tag{3-1}$$

Due to the consideration of centerline following is the state information of s changed to; $s = x$, $\dot{s} = v$, and $\ddot{s} = a$. The calculated velocity profiles are constrained by the comfortable range for the acceleration and jerk domain; $|a_{max}| \leq 3$ and $|\alpha_{max}| \leq 3$. Also is assumed that a vehicle will not exceed the v_{max} of a certain road section. The comfort of driving constrains the v_{max}. The trajectory calculation will be executed one by one for the vehicles in the scenario. So the trajectories for the first and no conflict vehicles will be calculated at first, followed by the

second vehicle, then the third car and so on. Calculating one vehicle after the other ensures that the vehicles will adapt their trajectory based on the trajectory of the previous vehicles in the scenario and ensure no conflict.

No conflict trajectory

At first, the no conflict trajectory will be calculated. The no conflict trajectory is first calculated for each vehicle in the traffic scenario to determine if certain vehicles will intersect at their conflict area. The existence of conflict is found by comparing their arrival times at the conflict area ($t_f = t_{in} + \frac{L_{conflict}}{v_{max}}$), with t_f the arrival time at the conflict area, t_{in} the arrival time to start of the intersection, $L_{conflict}$ the distance from the start of the intersection to the conflict area, and v_{max} the maximum velocity on the intersection. If the difference between the arrival times is larger then the safety time gap of one second, there is no need to change the trajectory of a vehicle, because there will be no conflict between the vehicles. The given state information of the no conflict trajectories are shown in Table 3-1. As mentioned in Section 3-1 can the acceleration of the surrounding vehicles not be measured and therefore is the acceleration zero at ($t = 0$). Moreover, the velocity can be measured and the end velocity should be equal to the maximum velocity of that specific route option. So the known variables are L_{start} distance to intersection, v_{max} maximum velocity for that specific route and v_{start} the start velocity.

Start of prediction	Start of intersection
$t = 0$	$t = t_{in}$
$x(0) = 0$	$x(t_{in}) = L_{start}$
$v(0) = v_{start}$	$v(t_{in}) = v_{max}$
$a(0) = 0$	$a(t_{in}) = 0$

Table 3-1: The end and start constraints for a no conflict trajectory

By solving Equation 3-1 using the state information from Table 3-1 the coefficients can be obtained. These equations result in: $c_0 = 0$, $c_1 = v_{start}$, and $2c_2 = 0$ so $c_2 = 0$. This will adapt the equations for $t = t_{in}$, resulting in the values for the last coefficients.

$$c_3 = \frac{10L}{t_{in}^3} + \frac{6v_{start}}{t_{in}^2} - \frac{4v_{in}}{t_{in}^2} \tag{3-2}$$

$$c_4 = -\frac{15L}{t_{in}^4} + \frac{7v_{start}}{t_{in}^3} + \frac{8v_{in}}{t_{in}^3} \tag{3-3}$$

$$c_5 = \frac{6L}{t_{in}^5} - \frac{3v_{start}}{t_{in}^4} - \frac{3v_{in}}{t_{in}^4} \tag{3-4}$$

These parameters are used in Equation 3-5 for jerk to find the actual value for t_{in}, arrival time at the intersection. The jerk optimal trajectory profile for a vehicle is symmetric to ensure minimal jerk. Due to the properties of the jerk optimal velocity profiles is the jerk of a trajectory $\alpha(\frac{1}{2}t_{in}) = 0$, where $\alpha = \dot{a} = \dddot{s}$.

$$\begin{bmatrix} 60t_0^2 & 24t_0^2 & 6 & 0 & 0 & 0 \\ 60t_1^2 & 24t_1^2 & 6 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} \dddot{s}_0 \\ \dddot{s}_1 \end{bmatrix} \tag{3-5}$$

By combining the coefficient from equations 3-2, 3-3 and 3-4 in $\alpha(\frac{1}{2}t_{in}) = 0$ the final value for $t_{in} = \frac{2L_{start}}{v_{start}+vmax}$.

Merge behind & in front trajectory

Secondly, the trajectories for the merging manoeuvres will be explained. The merge trajectories are only calculated if the vehicles will have conflict based on their arrival times at the conflict areas using the no conflict trajectories. Once a possible conflict is found, the second vehicle will have to adapt its trajectory based on the arrival time of the first car. For the calculation of the behind and in front trajectory is the data from Table 3-2 available, with L_{start} distance to intersection, $L_{conflict}$ distance from the start of the intersection to the conflict point, v_{max} is maximum velocity for specific route option, v_{start} is the current velocity of a vehicle, and t_f is the desired final arrival time.

Start of prediction	Start of intersection	Conflict point
$t = 0$	$t = t_{in}$	$t = t_f$
$x(0) = 0$	$x(t_{in}) = L_{start}$	$x(t_f) = L_{start} + L_{conflict}$
$v(0) = v_{start}$	$v(t_{in}) = v_{in}$	$v(t_f) = v_{max} \geq v_{in}$
$a(0) = 0$	$a(t_{in}) = 0$	$a(t_f) = 0$

Table 3-2: Table with variable information for a merge behind and in front trajectory

The desired arrival time t_f is calculated as follows $t_f \pm 1 = t_{in} + L_{conflict} * v_{max}$, where t_f the arrival time of the previous vehicle at the conflict point with 1 second added or subtracted depending on the type of manoeuvre (for behind +1 and -1 for in front) to maintain a safe distance. To determine the coefficients the Equations 3-6 and 3-7 are solved. The final arrival time t_f for a vehicle to reach a conflict area is built using two segments; driving towards the intersection t_1 and driving from the start of the intersection to the point of conflict t_2. Therefore $t_f = t_1 + t_2$. For the first part $t = t_1$.

$$\begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0^1 & 1 \\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1^1 & 1 \\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0^1 & 1 & 0 \\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1^1 & 1 & 0 \\ 20t_0^3 & 12t_0^2 & 6t_0^1 & 2 & 0 & 0 \\ 20t_1^3 & 12t_1^2 & 6t_1^1 & 2 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \dot{s}_0 \\ \dot{s}_1 \\ \ddot{s}_0 \\ \ddot{s}_1 \end{bmatrix} \tag{3-6}$$

For the second part $t = t_2$

$$\begin{bmatrix} t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1^1 & 1 \\ t_2^5 & t_2^4 & t_2^3 & t_2^2 & t_2^1 & 1 \\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1^1 & 1 & 0 \\ 5t_2^4 & 4t_2^3 & 3t_2^2 & 2t_2^1 & 1 & 0 \\ 20t_1^3 & 12t_1^2 & 6t_1^1 & 2 & 0 & 0 \\ 20t_2^3 & 12t_2^2 & 6t_2^1 & 2 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_{11} \\ c_{10} \\ c_9 \\ c_8 \\ c_7 \\ c_6 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \dot{s}_1 \\ \dot{s}_2 \\ \ddot{s}_1 \\ \ddot{s}_2 \end{bmatrix} \tag{3-7}$$

From these matrices (3-6 & 3-7) can be concluded that; $c_0 = 0$, $c_1 = v_{start}$, $2c_2 = 0$, $c_6 = 0$, $c_7 = v_{in}$, and $c8 = 0$. By rewriting the equations above the unknown coefficients can be found, which will result in the following equations.

For the first part

$$c_3 = \frac{10L_{start}}{t_1^3} + \frac{6v_{start}}{t_1^2} - \frac{4v_{in}}{t_1^2} \tag{3-8}$$

$$c_4 = -\frac{15L_{start}}{t_1^4} + \frac{7v_{start}}{t_1^3} + \frac{8v_{in}}{t_1^3} \tag{3-9}$$

$$c_5 = \frac{6L_{start}}{t_1^5} - \frac{3v_{start}}{t_1^4} - \frac{3v_{in}}{t_1^4} \tag{3-10}$$

For the second part

$$c_9 = \frac{10L_{conflict}}{t_2^3} + \frac{6v_{in}}{t_2^2} - \frac{4v_{max}}{t_2^2} \tag{3-11}$$

$$c_{10} = -\frac{15L_{conflict}}{t_2^4} + \frac{7v_{in}}{t_2^3} + \frac{8v_{max}}{t_2^3} \tag{3-12}$$

$$c_{11} = \frac{6L_{conflict}}{t_2^5} - \frac{3v_{in}}{t_2^4} - \frac{3v_{max}}{t_2^4} \tag{3-13}$$

Combined this will result in one equation for t_1 and one for t_2: $t_1 = \frac{2L_{start}}{v_{start}+v_{in}}$ & $t_2 = \frac{2L_{conflict}}{v_{in}+v_{max}}$. By knowing $t_f = t_1 + t_2$, it will result in $t_f = \frac{2L_{start}}{v_{start}+v_{in}} + \frac{2L_{conflict}}{v_{in}+v_{max}}$. Hereafter, the value for v_{in} is found by rewriting the formula. Eventually will v_{in} be used to determine the trajectory to merge in front or behind. The value for v_{in} determines the arrival time at the intersection based on a minimal continuous jerk trajectory.

Following trajectory

The following manoeuvre occurs when a vehicle is driving behind another vehicle while approaching the intersection. The following scenario is only applied to the ego vehicle. The following vehicle will have to adapt its trajectory to maintain a safe distance to the vehicle in front. To maintain the safe distance the ego vehicle uses the parameters from Table 3-3, where $t_{in} = t_{in,lead} + 1.5$. The $t_{in,lead}$ is the arrival time of the leading vehicle at the start of the intersection. The 1.5 seconds are added to ensure a safe following distance between the vehicles. As mentioned in Chapter 1 is the GRT not capable of braking as hard as the other traffic participants due to the standing people inside the vehicle. Therefore a safe following distance is essential. The trajectory of the following vehicle will be calculated in the same way the merge behind trajectory is calculated by using Equation 3-1, assuming the parameters in Table 3-3, resulting in $c_0 = 0$, $c_1 = v_{start}$, and $c_2 = 0$.

Start of prediction	Start of intersection
$t = 0$	$t = t_{in}$
$x(0) = 0$	$x(t_{in}) = L_{start}$
$v(0) = v_{start}$	$v(t_{in}) = v_{max}$
$a(0) = 0$	$a(t_{in}) = 0$

Table 3-3: The end and start constraints for a no conflict trajectory

The other coefficients are found by solving Equation 3-1, resulting in the following values for the coefficients;

$$c_3 = \frac{10L_{start}}{t_{in}^3} + \frac{6v_{start}}{t_{in}^2} - \frac{4v_{in}}{t_{in}^2} \tag{3-14}$$

$$c_4 = -\frac{15L_{start}}{t_{in}^4} + \frac{7v_{start}}{t_{in}^3} + \frac{8v_{in}}{t_{in}^3} \tag{3-15}$$

$$c_5 = \frac{6L_{start}}{t_{in}^5} - \frac{3v_{start}}{t_{in}^4} - \frac{3v_{in}}{t_{in}^4} \tag{3-16}$$

These coefficients are used in Equation 3-5 for the jerk to find the actual value for v_{in}. Due to the properties of the velocity profiles is $\alpha(\frac{1}{2}t_{in}) = 0$. By using the matrix in Equation 3-5 and the coefficients from Equations 3-14, 3-15 and 3-16 will result in the final value for v_{in}.

3-5 Block 5: Manoeuvre estimation

This section is dedicated to the last building block; manoeuvre estimation. Manoeuvre estimation will determine the likelihood between the different trajectories found in the previous section. In Section 2-5 several methods from the literature were introduced. The following section will elaborate on the chosen method between the methods from Section 2-5. Hereafter the chosen method will be described in detail.

3-5-1 Method choice

This section will give a brief discussion based on the several methods introduced in Section 2-5. First, the cost function methods where no data set is needed to determine the starting transition probabilities. These cost functions lack performance compared to the IMM and Bayesian networks because the other methods use past measurements and data sets to train the method for better performance. In contrast, the cost functions only use the current measurement (optimal cost) or the current and the previous measurement (lowest cost-gradient). Secondly, the IMM outperforms both the cost methods [39]. However, the most significant difference in overall performance is the poor performance of the cost-gradient method for an overtaking manoeuvre, which is not considered in this thesis. The cost methods outperform the other methods on calculation time because they can make an instant assessment based on the cost function determined in building block four. Although the cost functions have a shorter assessment time, the networks will be preferred because the networks will be trained using data sets. Therefore will a network outperform the cost functions. Unfortunately, there were no data sets available at 2getthere to train the network and therefore will the model choose the cost functions to determine the probability distribution between the manoeuvres.

3-5-2 Detailed description

In the previous section is the cost function chosen as the desired method for determining the probability distribution between the different manoeuvres. This section will elaborate on calculating and incorporating the cost function into the model. At first, the cost function will use the trajectories calculated in the fourth building block as an input for the calculation. The cost for a trajectory is $cost = j^2 + \Delta v$, where j is the jerk m/s^3, and Δv is the difference between the current velocity and the calculated end velocity for each possible trajectory. The cost will be calculated for each possible trajectory. The number of possible trajectories for a vehicle is dependent on the manoeuvre. There is only one trajectory for a vehicle with no conflict or the first vehicle to pass the conflict area, which is the no conflict manoeuvre. If a vehicle has to adapt its trajectory to a previous vehicle, that specific vehicle can merge behind and in front. So if a vehicle has multiple possible trajectories, the cost function has to choose the trajectory with the lowest cost. Once the choice is made, the next vehicle in the passing order will adapt its trajectory based on the chosen trajectory for the previous vehicle. The cost function will be calculated each time step and for each vehicle. As mentioned in Section 3-3 the model does determine all possible passing orders for a specific traffic scenario. The individual cost functions will be combined to determine the total cost function for a specific passing order of each branch. Once the total cost for each branch is calculated, the probability for each passing order is determined by the probability of a sequence in $\% = \frac{\frac{1}{costofsequence}}{\sum \frac{1}{costofsequence}}$. This equation results in a sequence with low cost having a high probability and high-cost manoeuvres having a low probability.

3-6 Summary

The model is based on an interaction-aware motion model, resulting in five separate building blocks to summarise this chapter. This chapter highlights the type of model and the methods used for each building block. The chosen model is an interaction-aware motion model, resulting in five separate building blocks. This chapter highlighted the choices made for each building block and gave a detailed description of each chosen method. At first, the variables used for the calculations were chosen. The number of variables was limited by the application at 2getthere, resulting in three variables; position, the magnitude of velocity and direction of velocity. The direction of the velocity can be measured due to the nonholonomic properties. Once the variables are collected, the route possibilities for each vehicle are found by the breadth-first method. The breadth-first method is fast and straightforward. The position of a vehicle will correspond to a specific lane for that vehicle in the traffic scenario. The current lane will hold the successor route information to find each vehicle's possible future route possibilities in the traffic scene. Based on all actors' route possibilities, the conflict areas between the vehicles are determined by finding intersections between the polygon shapes of each route possibility. Based on the conflict areas, an initial formation is constructed that progresses by changing the relative order of the vehicles to create a formation tree. The formation tree will hold all possible passing orders for the specific scenario. The first vehicle's trajectory is not influenced, but the vehicles that follow will adjust their trajectory based on the previous vehicles passing the conflict area. The trajectory for each vehicle is based on a one-dimensional velocity profile, assuming a vehicle will follow the centerline of a lane. The trajectories minimize the jerk of each vehicle's trajectory, which directly affects comfort. Eventually, the last building block determines the cost for each vehicle's trajectory. It combines all vehicles' cost to determine the total cost of a specific passing order of the traffic scenario. Finally, the probability between these passing orders is calculated based on the total sum of the cost to determine the probability of the possible trajectories for each vehicle.

Chapter 4

Simulation & results

This chapter will give an insight into the performance of the chosen model discussed in Chapter 3. The model's performance will be tested in a fixed environment on the Zaventem map. The map is a T-junction, where two different scenarios are run. This chapter will first explain the set-up and how the scenarios are created. In the second section, the two scenarios will be highlighted with an explanation of the scenario, followed by the results for that specific scenario. Hereafter, the results will be discussed and compared with an ordinary Adaptive Cruise Control (ACC) model. The comparison will help give an insight into the answer for the problem statement: is it possible to create a motion prediction model for the surrounding vehicles of the ego vehicle to allow for pro-active velocity planning of the ego vehicle itself.

4-1 Set-up

This section describes the set-up used for creating the scenarios and results. The 2getthere application for the GRT is applied at the Zaventem airport. The map of the infrastructure at Zaventem is provided using an OpenDrive format, as mentioned in previous sections. Due to the OpenDrive format, the data is converted to the lanelet format described in Section 3-2. Figure 4-1 shows the set-up scenario in the lanelet format.

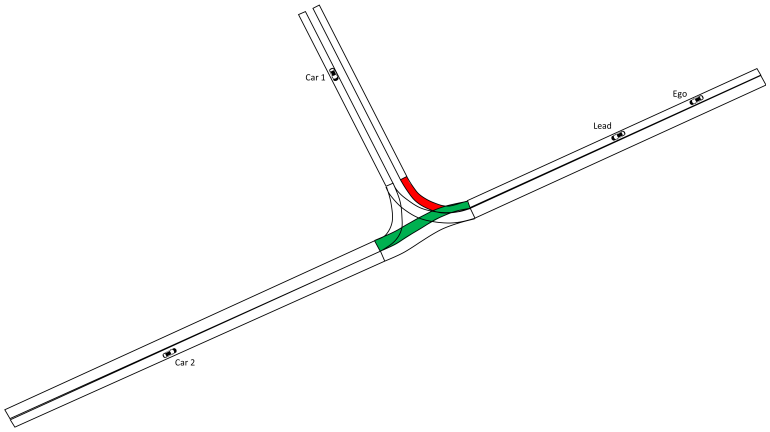

Figure 4-1: The T-junction scenario shown in the lanelet format

The choice for this T-junction is due to the clear connection and separation of the individual road sections. For the different scenarios, there are four vehicles within the scenario. The four vehicles in the scenarios are shown in Figure 4-1; the ego vehicle, lead vehicle and two surrounding traffic. The maximum of four vehicles in the scenarios is due to the sensors' limitations on the GRT itself. The limitations originate from sensor field blocking by the surrounding vehicles. In the two scenarios, the ego vehicle will go straight ($v_{max} = 14m/s$). Although it will have the possibility to go right, the planner knows the route intentions of the ego vehicle, and therefore the right turn is not taken into account. The lead vehicle will have two route possibilities; straight ($v_{max} = 14m/s$) and a turn right ($v_{max} = 7.5m/s$). The coloured lanes in Figure 4-1 represent the two route possibilities for the lead vehicle at the T-junction; the red lane for the turn right and the green lane for going straight. These colours will create more clarity for the predicted trajectories in the results section. The first surrounding vehicle is approaching from the right from the standpoint of the ego vehicle. This vehicle will have two route possibilities: turn left ($v_{max} = 10m/s$) and turn right ($v_{max} = 7.5m/s$). The last vehicle is the second surrounding vehicle, known as car 2, which will only have one option to go straight ($v_{max} = 14m/s$) at the intersection, as can be seen in Figure 4-1. All vehicles in the scenario will start with $v = 14m/s$ in every scenario. The scenarios are created using the driving simulator toolbox from MATLAB. The driving simulator toolbox offers the possibility to load the OpenDrive Zaventem map into the program. Once the map is loaded into the editor window, vehicles can be placed anywhere on the map. The vehicles are represented as rectangles with their heading and world coordinates. To create a moving scenario in the toolbox, waypoints must be drawn for each vehicle. These waypoints can be placed anywhere on the map. They will hold the position, heading and velocity of a vehicle at the waypoints, resulting in a step profile for the acceleration of each vehicle as shown in Figure 4-2. This step profile is not desired to create smooth traffic scenarios, but it can still give an insight into the model's performance.

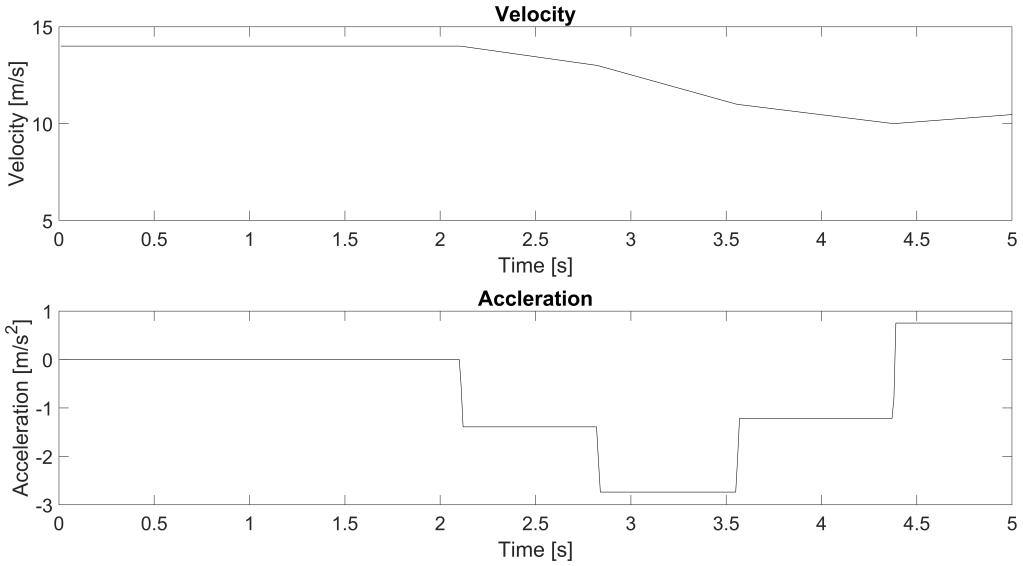

Figure 4-2: The velocity and acceleration profile from the driving simulator toolbox of MATLAB

4-2 Scenarios

This section will introduce the different scenarios on the T-junction mentioned in the section above. There will be a total of two scenarios to test the chosen model. Each of the two scenarios has four vehicles and are all constructed in the set-up from Section 4-1. The first part of each scenario will be explained and visualized by a bird's eye view of the map with the route choices of every vehicle shown on the map. The second part shows the results of the chosen motion model at three different time instances. At each time instance, the motion prediction model makes a prediction based on the distance to the intersection and the current velocity of a vehicle. Six metrics show the results of the motion prediction model.
The first metric will be the predicted velocity trajectories for the lead vehicle to the start of the intersection, with the route options colour-coded by the red and green lines (top left graph in the figures). The second metric are the predicted velocity trajectories for the ego vehicle based on the scenario and the predicted motion of the lead vehicle (bottom left graph in the figures). The ego vehicles' motion is predicted because it will influence the behaviour of the two surrounding vehicles in the traffic scenario. The velocity trajectories are shown till the start of the intersection. If a vehicle has the desired velocity at the end of the predicted trajectory, the vehicle will continue with a constant velocity over the intersection. However, when a vehicle has a lower velocity than that route option's maximum velocity, the motion model predicts a smooth trajectory accelerating to the desired velocity at the intersection. These acceleration profiles are used to determine the arrival time at conflict areas. However, these acceleration profiles will not be shown in the results because the prediction model will predict the motion of the lead vehicle to the start of the intersection. The third metric is a probability distribution of the predicted trajectories for the lead vehicle at 2 seconds ahead from the current time instance (top right graph in the figures). Fourth, a probability distribution of the predicted trajectories for the ego vehicle at 2 seconds ahead from the current time instance (bottom right graph in the figures). The fifth metric is the weighted average velocity of the ego vehicle based on the probability distributions for two seconds in advance from the specific time instant, based on the lowest 20% (bottom right graph in the figures). Finally, the conservative future velocity of the lead vehicle up to 2 seconds in advance is calculated based on the probability distribution in the top right figure. The conservative trajectory is represented by the lowest 20% of the predicted trajectories, ensuring that the time gap between the ego vehicle and the lead vehicle never becomes too small, leading to activation of the emergency braking. It could increase the time gap between the vehicles, but this would not result in unsafe or uncomfortable behaviour for the ego vehicle. Therefore the lowest 20% will always ensure that the ego vehicle will keep a safe distance to the lead vehicle and establish a safe and comfortable ride for the passengers.

4-2-1 Scenario 1: Lead vehicle turning right

This section will explain the first scenario and the calculated results based on the given traffic scenario. In the first scenario, the vehicles will behave as follows; the ego vehicle will go straight, the lead vehicle will execute a right turn, the first surrounding vehicle will turn right, and the second surrounding vehicle will go straight. The lead vehicle will decelerate from $v = 14m/s$ to the desired $v_{max} = 7.5m/s$ for the right turn. Due to the deceleration of the lead vehicle. The ego vehicle will decelerate accordingly to remain a safe time gap of 1.5 seconds between the ego vehicle and the lead vehicle. The visual representation of the scenario is depicted in Figure 4-3.

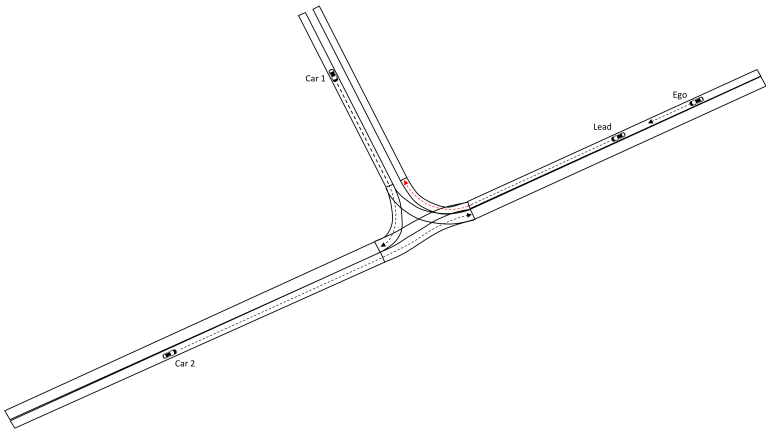

Figure 4-3: The visual representation of the first scenario on the map

Scenario results

In the first scenario, the vehicles will follow the routes as shown in Figure 4-3. The introduction of this chapter introduced how the results of each time instance will be depicted. The results for the first time instance ($t = 0$) are shown in Figure 4-4. The current velocities and headway are; the ego vehicle and lead vehicle $v = 14m/s$ with a current time headway of 1.4 seconds. So the ego vehicle should increase the time headway between the vehicles. As seen in the figure, does the model not predict any conflict between the lead vehicle and the surrounding traffic. Due to the prediction of a single trajectory for each route option of the lead vehicle, as can be seen by the one red profile (right turn) and one green profile (straight) in the top left of Figure 4-4. Therefore there will also be only two velocity profiles for the ego vehicle shown in the bottom left graph of Figure 4-4. The straight route's velocity profile will brake slightly to increase the gap to the leading vehicle to 1.5 seconds. The right turn velocity profile will ensure that the ego vehicle will have a 1.5 second gap when arriving at the intersection. The weighted average for the lead vehicle based on the lowest 20% probability is $v_{futurelead} = 13.5m/s$, meaning there is a 20% chance that the lead vehicle will turn right. Both velocity trajectories for the ego vehicle have a velocity of $v_{future,ego} = 13.5m/s$ for the lowest 20%, based on the probability distribution shown in the bottom right of Figure 4-4. So the model will output that the velocity of the ego vehicle at the end of the next 2 seconds should be $v = 13.5m/s$, to maintain a safe distance between the vehicles while maintaining a comfortable ride for the passengers.

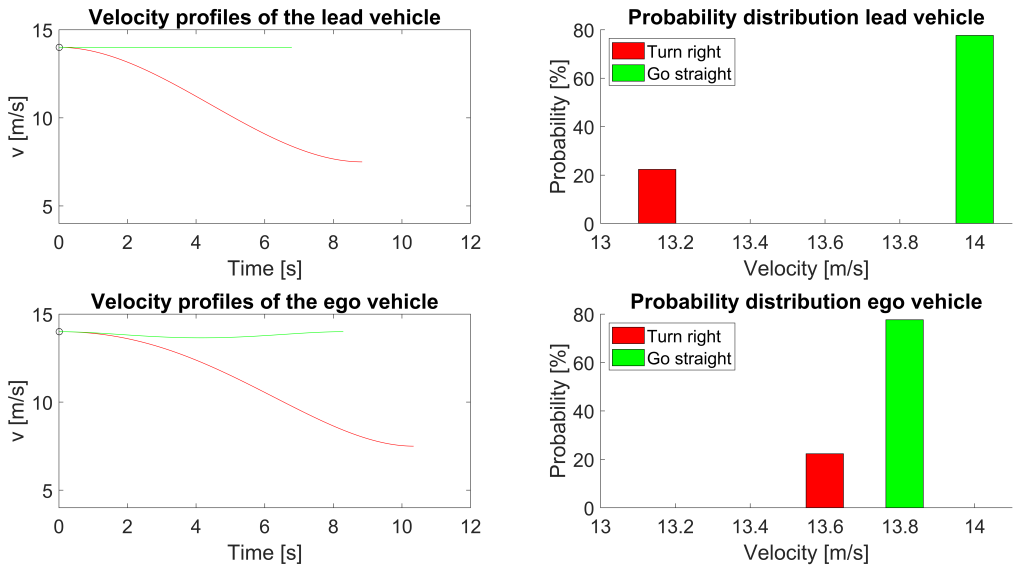

Figure 4-4: The results of the motion prediction for the first scenario at t = 0 s

As the scene continues, the model will adopt the motion prediction based on the newly gathered data. The new time instance is shown in Figure 4-5 at $t = 2s$. At the point of gathering the data does the ego vehicle have a velocity of $v = 12.2m/s$ and the lead vehicle drives $v = 11.8m/s$. The distance between the vehicles is 18.5 meter with a time headway of 1.5 seconds, so the ego vehicle does not have to increase the time headway. In the top left of Figure 4-5, there are four different velocity profiles for the lead vehicle; three for straight (green) and one for turning right (red). The increase in the number of velocity profiles shows that the model predicts conflict between the vehicles in the traffic scenario, based on the vehicles no conflict trajectories. To avoid a collision between the vehicles, the motion model predicts alternative trajectories for each vehicle, resulting in four velocity profiles shown in the top left. These four velocity profiles also result in four velocity profiles for the ego vehicle. The top right figure shows a small increase in the probability for the right turn manoeuvre. In the bottom left plot of Figure 4-5 it can be seen that the velocity profiles for the ego vehicle for 2 seconds ahead are relatively similar to each other. The probability distribution of the trajectories for the lowest 20% for the lead vehicle results in a predicted velocity for 2 seconds ahead of $v_{futurelead} = 11m/s$. The model output for the ego vehicle velocity based on the lowest 20% is $v_{future,ego} = 9.9m/s$. The model will output that the velocity of the ego vehicle should decrease by around $2m/s$ for the next 2 seconds to remain a safe and comfortable ride.

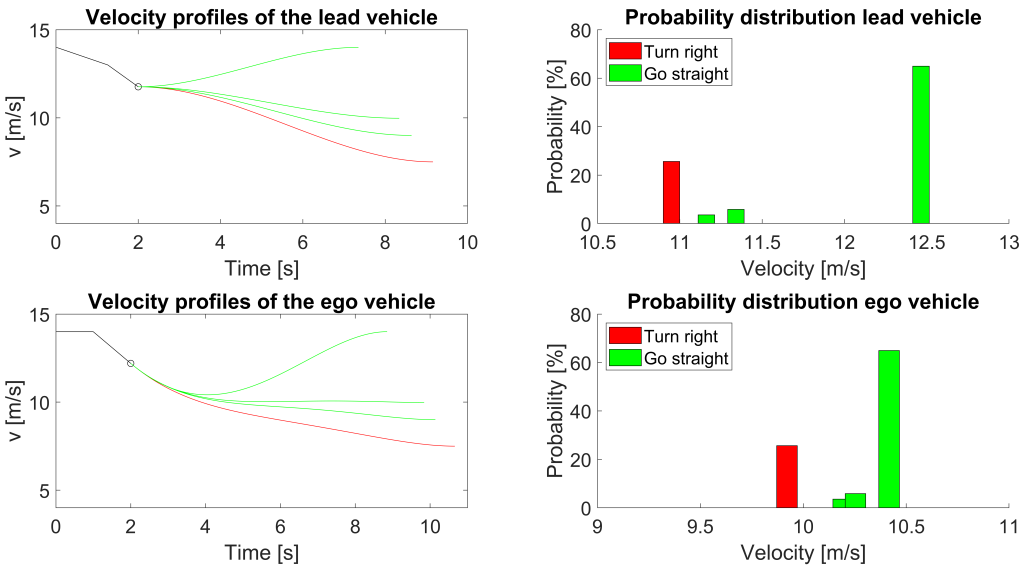

Figure 4-5: The results of the motion prediction for the first scenario at t = 2 s

While the scene further progresses, the motion prediction for the traffic scene will change accordingly. Figure 4-6 shows the results for the motion prediction at $t = 4s$. During the previous time instance and the current time instance, the vehicles kept decelerating in order for the leading vehicle to execute a right turn at the start of the intersection. The continuation of the deceleration of both vehicles is shown by the black lines in Figure 4-6. The new velocities of both the ego vehicle and the lead vehicle are $v = 9.5m/s$ with a current time headway of 2 seconds. So the ego vehicle should decrease the time headway between the vehicles to reach the desired headway of 1.5 seconds. The figure shows that the motion model still predicts conflict between the vehicles in the traffic scene, resulting in four trajectories. A single right turn trajectory, because no conflict will occur when turning right as shown in Figure 4-3, and three trajectories for going straight. The probability distribution of the trajectories for the lead vehicle is shown in the top right of Figure 4-6. As can be seen in the figure is the probability of the leading vehicle taking a right turn almost doubled from 22 % to 41 %, compared to the results at $t = 2s$. The probability distribution of the trajectories for the lowest 20% for the lead vehicle results in a predicted velocity for 2 seconds ahead of $v_{futurelead} = 9m/s$. The model output for the ego vehicle velocity based on the lowest 20% is $v_{future,ego} = 9.5m/s$. The model will output that the velocity of the ego vehicle should be $v = 9.5m/s$ for the next 2 seconds to remain a safe and comfortable ride. Due to the time headway of 2 seconds when gathering the data at $t = 4s$, does the ego vehicle need to decrease the time headway. Therefore is the future velocity for the ego vehicle estimated at $v = 9.5m/s$.

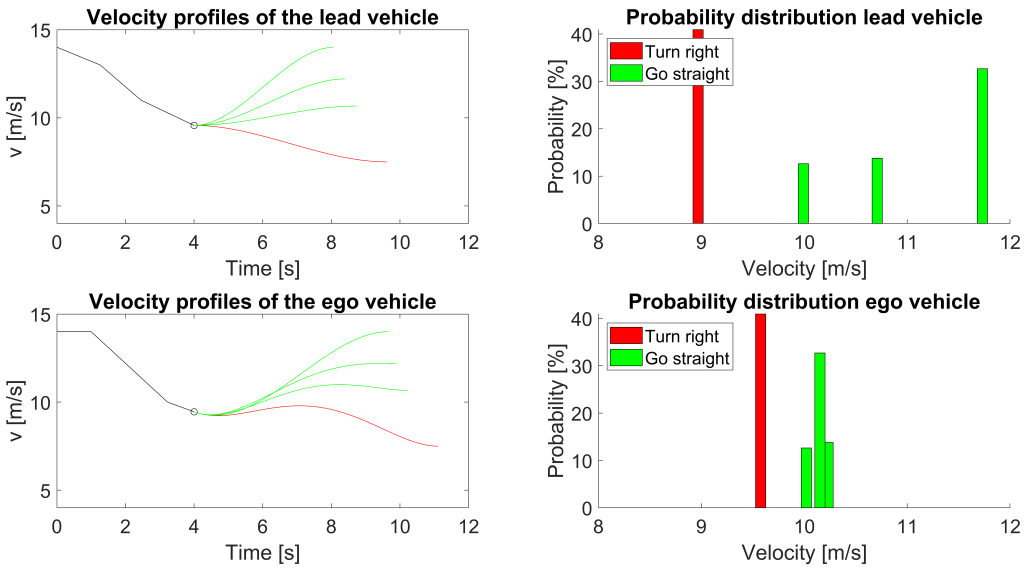

Figure 4-6: The results of the motion prediction for the first scenario at t = 4 s

As shown in the three figures in this section, the prediction model always predicts a smooth trajectory for the different trajectories of the lead vehicle while considering the desired headway between the vehicles. The ego vehicle will induce extra braking if the time gap is too small and suggest maintaining its velocity if the time gap is too large if the model suspects the lead vehicle will decelerate in the future. By maintaining the velocity of the ego vehicle, the time gap between the vehicles will slowly decrease to the desired headway and result in comfort and safety for the passengers.

4-2-2 Scenario 2: Lead vehicle going straight

This section highlights the second scenario and the results based on the given traffic scenario. In the second scenario, the vehicles will behave as follows; the ego vehicle and lead vehicle will go straight, the first surrounding vehicle will turn left, and the second surrounding vehicle will go straight. The lead vehicle will decelerate from $v = 14m/s$ to $v = 10m/s$ and accelerate back to $v = 14m/s$ for going straight. The lead vehicle will slightly brake to let car one pass in front and then accelerate back to the desired velocity of $v = 14m/s$. The bird's eye view of the second scenario is shown in Figure 4-3.

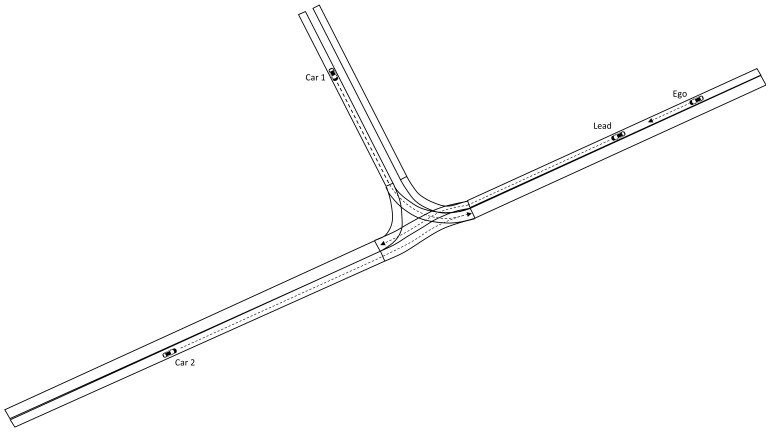

Figure 4-7: The visual representation of the second scenario on the map

Scenario results

In the second scenario, the vehicles will follow the routes as shown in Figure 4-7. The results for the first time instance ($t = 0$) are shown in Figure 4-8. The velocities of both the ego vehicle and lead vehicle are $v = 14m/s$ with a current time headway of 1.5 seconds. Therefore the ego vehicle will not have to influence the time headway because it is already equal to the desired headway of 1.5 seconds. In Figure 4-8 it can be seen that the model predicts conflict between the lead vehicle and the surrounding traffic. Due to the prediction of three trajectories for going straight (green) and one profile for turning right (red) shown in the top left of Figure 4-4. The straight route's velocity profile of the ego vehicle will not induce any braking due to the time headway of already 1.5 seconds. The probability distribution of the lead vehicle shows that there is a 40% chance for the lead vehicle to make a right turn. Therefore the weighted average for the lead vehicle based on the lowest 20% probability is $v_{futurelead} = 13.3m/s$. For the ego vehicle the predicted velocity is $v_{future,ego} = 13.7m/s$ for the lowest 20%, based on the probability distribution shown in the bottom right of Figure 4-8. So the model will output that the velocity of the ego vehicle at the end of the next 2 seconds should be $v = 13.7m/s$, to maintain a safe distance between the vehicles to maintain a comfortable ride for the passengers.

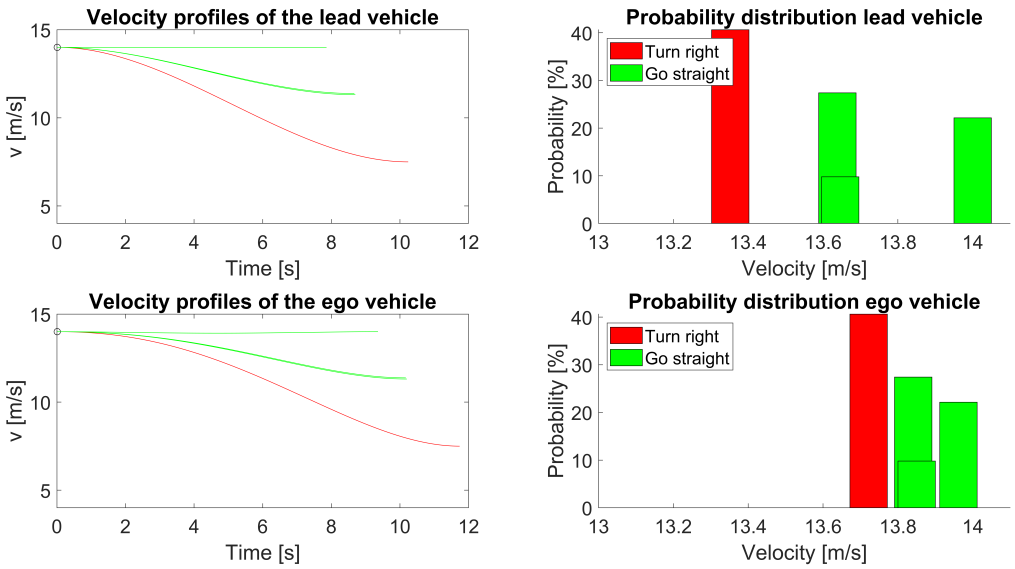

Figure 4-8: The results of the motion prediction for the second scenario at t = 0 s

As the scene continues, the model will adopt the motion prediction based on the newly gathered data. The new time instance is shown in Figure 4-9 at $t = 3s$. At the point of gathering the data does the ego vehicle have a velocity of $v = 12.5m/s$ and the lead vehicle drives $v = 10.5m/s$. The distance between the vehicles is 17 meter with a time headway of 1.3 seconds, so the ego vehicle needs to increase the time headway. In the top left of Figure 4-9, there are two unique velocity profiles for the lead vehicle; one for straight (green) and one for turning right (red). These two velocity profiles also result in two velocity profiles for the ego vehicle, shown in the bottom left figure. By deceleration of the lead vehicle and the ego vehicle, the motion model no longer predicts any conflict between the vehicles in the scenario. In the bottom left plot of Figure 4-5 it is shown that the velocity profiles for the ego vehicle for 2 seconds ahead are relatively similar to each other. The probability for a right turn is increased to 50%, therefore is the probability distribution for the lowest 20% for the lead vehicle for 2 seconds ahead $v_{futurelead} = 10m/s$. The model output for the ego vehicle velocity based on the lowest 20% is $v_{future,ego} = 8.8m/s$. The model will output that the velocity of the ego vehicle should decrease by around $4m/s$ for the next 2 seconds to remain a safe and comfortable ride while considering the desired time headway of 1.5 seconds.

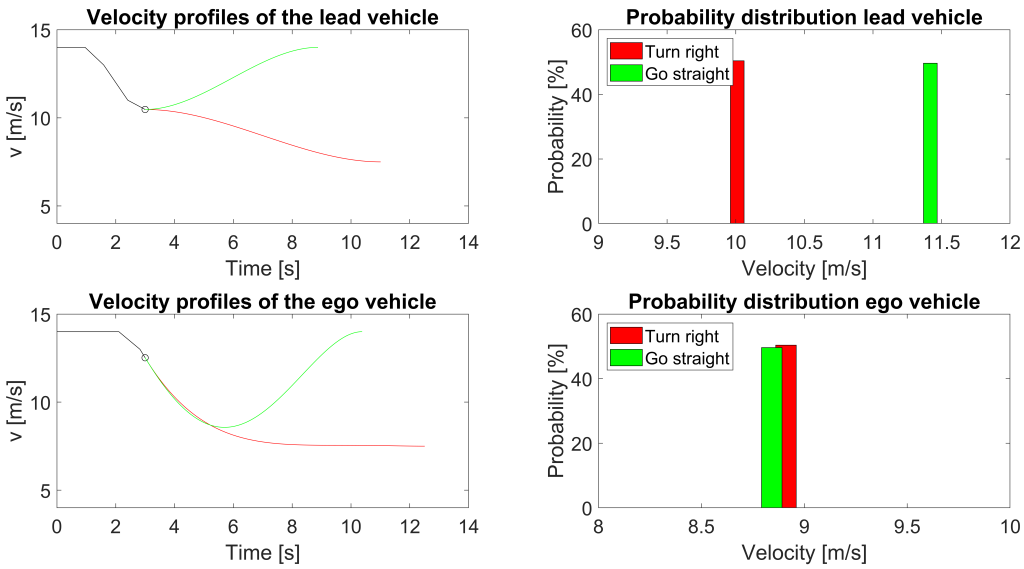

Figure 4-9: The results of the motion prediction for the second scenario at t = 3 s

Finally, Figure 4-10 shows the results for the motion prediction for scenario two at $t = 5s$. During the previous time instance and the current time instance, the lead vehicle accelerated towards the desired velocity for going straight at the start of the intersection. The change from decelerating towards accelerating of both vehicles is shown by the black lines in Figure 4-10. The new velocities of both vehicles are; the ego vehicle $v = 10.5m/s$ and the lead vehicle $v = 11.25m/s$ with a current time headway of 1.6 seconds. So the ego vehicle should decrease the time headway between the vehicles to reach the desired headway of 1.5 seconds. The top left of Figure 4-10 shows that the motion model still does not predict any conflict between the vehicles, resulting in only two trajectories. A single right turn trajectory and a single straight trajectory, because no conflict will occur in the scenario of Figure 4-7. The probability distribution of the trajectories for the lead vehicle is shown in the top right of Figure 4-10. As shown in the figure, the leading vehicle taking a right turn decreased from 50% to around 30%, compared to the last time instance $t = 3s$. The probability distribution of the trajectories for the lowest 20% for the lead vehicle results in a predicted velocity for 2 seconds ahead of $v_{futurelead} = 10.1m/s$. The model output for the ego vehicle velocity based on the lowest 20% is $v_{future,ego} = 11.4m/s$. The model will output that the velocity of the ego vehicle should be $v = 11.4m/s$ for the next 2 seconds to remain a safe and comfortable ride. Due to the time headway of 1.6 seconds when gathering the data at $t = 5s$, does the ego vehicle need to decrease the time headway. Therefore is the future velocity for the ego vehicle estimated at $v = 11.4m/s$.

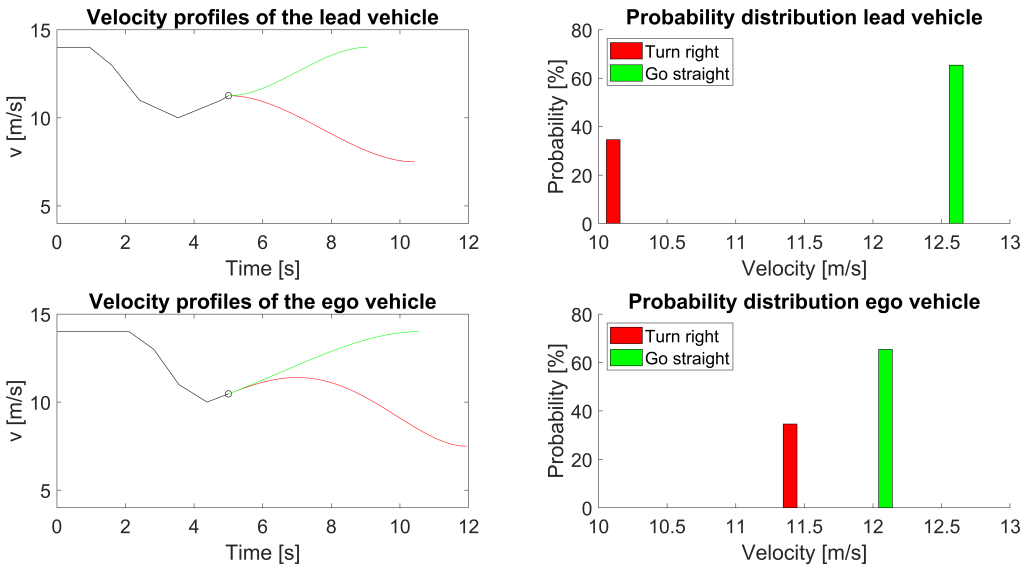

Figure 4-10: The results of the motion prediction for the second scenario at t = 5 s

The three figures in this section show that the motion model always predicts a smooth trajectory based on the changing circumstances of the scenario. The ego vehicle will always ensure that the time headway is equal to the desired value at the start of the intersection. The ego vehicle will not instantly change the time headway because the distance to the intersection in both scenarios is no longer than 100 meters. If the ego vehicle would change the time headway within a couple of seconds, it will result in high acceleration or deceleration profiles to increase or decrease the time headway. Therefore the ego vehicle establishes a time gap of 1.5 seconds at the start of the intersection. The 1.5 seconds time gap also ensures that other vehicles will not merge between the lead and ego vehicle, resulting in abrupt braking for the ego vehicle and decreased safety and comfort for the passengers.

4-3 Discussion

This section will discuss the results shown in the previous section. At first, the results of the two scenarios described in the previous section will be compared to a common adaptive cruise control (ACC) model. To do so, the ACC model needs to be explained first. Adaptive-Cruise Control assists a driver for the longitudinal movement control by maintaining a selected time gap, velocity or prevent a collision by automatically accelerating and decelerating. The ACC uses the distance and the velocity difference between the vehicles as an input and will react only when the actual time gap between the vehicles differs from the desired time gap [22, 25]. Only reacting to the time gap between the vehicles results in the ACC being a reactive model and will only react when the situation has already changed. The results of the scenarios will highlight the comparison between the two models.
In Section 4-2-1 the time headway at $t = 2s$ between the ego vehicle and the lead vehicle is equal to the desired headway of 1.5 seconds. Therefore, an ordinary ACC model will not change the time gap between the vehicles because it is already equal to the desired value.

The ACC will change the velocity of the ego vehicle only if the time gap changes in the next time instance. Whereas the prediction model predicts a deceleration for the ego vehicle to maintain the desired headway at the start of the intersection as shown in the bottom left graph of Figure 4-5. Another example of such a prediction is shown in the second scenario at $t = 5s$ (Figure 4-10). Here the time gap is equal to 1.6 seconds, with the leading vehicle only having two possible trajectories. To guarantee a safe and comfortable ride for the passengers, the motion model always takes the lowest 20% into account. Therefore the ego vehicle should only increase its velocity slightly to decrease the time gap before decelerating again to adjust its velocity to the lead vehicle making a possible turn right. In contrast, an ACC model will increase the velocity of the ego vehicle till a new time instance before adjusting the velocity of the ego vehicle again. The ACC is a reactive model only changing the velocity of the lead vehicle if the time gap is not equal to the desired value. The prediction model will proactively predict the traffic scenario, resulting in a smooth trajectory profile for the ego vehicle. While keeping in mind the 20% most conservative velocity profile for the lead vehicle.
Although the results show good performance in these scenarios, several measures could improve the results even more. At first, is the acceleration of the surrounding vehicles currently not considered in this motion model due to the limitations of the GRT. The lack of acceleration negatively affects the results because the motion model cannot determine if the surrounding vehicles are accelerating or decelerating. Therefore is the initial acceleration of the surrounding vehicles zero. For instance, knowing the current acceleration of a vehicle would increase the probability of that vehicle continuing on that acceleration profile. The decelerating and accelerating chances are currently equal because the initial acceleration is zero for a surrounding vehicle. Secondly, the motion model does not store previously predicted trajectories. In Figure 4-4 the motion model predicts a specific trajectory for a right turn (shown in red). When the scenario progresses towards $t = 2s$ the vehicle seems to follow that predicted trajectory perfectly. Therefore, the model should recognise that the vehicle follows that predicted trajectory and increase the probability for a right turn manoeuvre. Therefore the results of the motion model could be increased by storing the previously predicted trajectories. Finally, the current motion model determines the probability of a specific trajectory for the lead vehicle based on the cost for all vehicles in the scenario. By doing so, all the vehicles in the traffic scenario are considered for deciding the probability. Although this seems the right measure, driving does not work this way in real life because the vehicle's movement will not be influenced by the vehicles driving behind him. Currently, a right turn can have a low probability because it is considered that the following vehicles also will have to slow down for the lead vehicles, increasing the total cost of that manoeuvre.

Chapter 5

Conclusion & recommendations

5-1 Conclusion

This thesis aims to show that motion prediction for the surrounding vehicles of the ego vehicle allows for pro-active velocity planning of the ego vehicle. The pro-active velocity planning will increase the smoothness of the ego vehicle's velocity, resulting in an increase of comfort for the passengers. The following four chapters answered the problem statement.

Chapter 1 gave an introduction to the problem statement and a general introduction to the currently used motion prediction models in the literature. The literature's motion models can be divided into three levels; physics-based motion models, manoeuvre-based motion models, and interaction-aware motion models. The physics-based motion model only takes each vehicle's vehicle kinematics and dynamics into account. This way, the actor will be addressed as a moving object without any dependencies or interactions to the surrounding environment (vehicles and infrastructure). The manoeuvre-based motion model represents a vehicle as a single moving object with no dependencies but considers that the motion of a vehicle consists of different predefined manoeuvres, known as route options. The interaction-aware motion model no longer represents a vehicle as a single moving object with no dependencies but represents a vehicle as a moving object with dependencies with the surrounding traffic. So the motion of a vehicle is restricted by the motion of surrounding vehicles. Each of the three motion levels is built using specific building blocks. The level determines the number of building blocks used. The interaction-aware motion model uses all five building blocks. The first building block is measurements, which gathers the state information of the actors in the traffic scenario. The second building block determines the route possibilities of each vehicle in the traffic scenario up to the prediction horizon. The third building block will determine the possible conflict between the vehicles in the traffic scenario. Hereafter, the trajectory planning block will calculate a trajectory for each possible manoeuvre of every vehicle in the traffic scenario. Finally, the manoeuvre estimation block will calculate the probability between these trajectories to determine the probability between the individual manoeuvres.

Chapter 2 briefly explained several methods found in the literature for each building block. As mentioned in Section 1-2, the interaction-aware motion model contains five building blocks, and therefore each of these blocks is considered.
At first, the measurements block. This block determines and collects the state information of the surrounding vehicles and the ego vehicle for the motion prediction model. The mainly

used state information are position and velocity. Additionally, the acceleration and yaw rate are added in some papers.
Block two gathers the route possibilities of the surrounding vehicles based on their current lane position. The breadth-first method determines the future lanes by the successor lane information of each lane in the scenario. The networks will also find the route possibilities based on the current lane. Other methods in the literature use a predefined set of possibilities for each vehicle.
Thirdly, the model determines the possible conflict points and interaction in the block interaction determination based on the route possibilities. This block contains two main methods; occupancy grid map and conflict areas. The occupancy map divides a map into cells, where each cell holds a probability of being occupied or free. The conflict areas method is divided into two approaches; the formation tree and the constant velocity model. The formation tree builds a tree based on all possible order changes for the vehicles in a traffic scenario. The constant velocity model assumes that a vehicle will continue on the centerline with a constant velocity till the conflict point. Once the data is gathered in block one, the route possibilities in block two, and the interaction determination in block three, the data from these blocks are combined in the block trajectory planning.
Trajectory planning is a vast subject in the literature, so only the methods that can rapidly create a smooth trajectory are mentioned. First, the constant yaw rate and acceleration model is introduced, which only works for physics-based motion models. It assumes constant acceleration and yaw rate during a time step and does not consider the infrastructure. Secondly, the trajectory planning in the Frenét-Frame contains four different methods using this principle. At first, the lateral and longitudinal separated trajectory optimization. The separation ensures a smooth trajectory within a short processing time. Hereafter, the double integrator method optimizes the cost of acceleration of the trajectory because a human driver directly affects the acceleration of a vehicle. The path-velocity decomposition assumes the vehicle will drive on the centerline of a lane, resulting in a one-dimensional problem. The trajectory is determined by finding the lowest cost for avoiding static and dynamic obstacles. At last, the bounded acceleration model determines the trajectory of the ego vehicle based on five aspects that bound the vehicle's acceleration.
Eventually, block 6, manoeuvre estimation, determines the probability between the several trajectories of each vehicle. The literature introduces several methods for determining these probabilities, integrating multiple model filter, cost function probability, and networks. The IMM filter uses separate filters for each vehicle state and updates the filter based on their covariance. The cost function probability determines the cost of a trajectory compared to the cost of other trajectories to find the probability of a specific trajectory. The networks are split into Bayesian networks and neural networks. The networks are trained on big data sets to determine their weights and biases.

Chapter 3 made a decision on the type of model and the methods used for each building block. The chosen model is an interaction-aware motion model, resulting in five separate building blocks to summarise this chapter. At first, the variables used for the calculations were chosen. The number of variables was limited by the application at 2getthere, resulting in three variables; position, the magnitude of velocity and direction of velocity. The direction of the velocity can be measured due to the nonholonomic properties. Once the variables are collected, the route possibilities for each vehicle are found by the breadth-first method. The breadth-first method is fast and straightforward. The position of a vehicle will correspond to

a specific lane for that vehicle in the traffic scenario. The current lane will hold the successor route information to find each vehicle's possible future route possibilities in the traffic scene. Based on all actors' route possibilities, the conflict areas between the vehicles are determined by finding intersections between the polygon shapes of each route possibility. Based on the conflict areas, an initial formation is constructed that progresses by changing the relative order of the vehicles to create a formation tree. The formation tree will hold all possible passing orders for the specific scenario. The first vehicle's trajectory is not influenced, but the vehicles that follow will adjust their trajectory based on the previous vehicles passing the conflict area. The trajectory for each vehicle is based on a one-dimensional velocity profile, assuming a vehicle will follow the centerline of a lane. The trajectories minimize the jerk of each vehicle's trajectory, which directly affects comfort. Eventually, the last building block determines the cost for each vehicle's trajectory. It combines all vehicles' cost to determine the total cost of a specific passing order of the traffic scenario. Finally, the probability between these passing orders is calculated based on the total sum of the cost to determine the probability of the possible trajectories for each vehicle.

In Chapter 4 the results of the chosen motion model are shown in a T-junction scenario. The motion model is tested in two scenarios. In the first scenario, the lead vehicle will execute a right turn, and in the second scenario, the lead vehicle will go straight at the intersection. In both scenarios, there are four vehicles, and the ego vehicle is always behind the lead vehicle. A total of six metrics shows the results of the scenarios. The first metric shows the predicted velocity trajectories for the lead vehicle to the start of the intersection. The second metric are the predicted velocity trajectories for the ego vehicle based on the scenario and the predicted motion of the lead vehicle. Hereafter, the probability distribution of the predicted trajectories for the lead vehicle at 2 seconds ahead is calculated. Fourth, the results show the probability distribution of the predicted trajectories for the ego vehicle at 2 seconds ahead. The fifth metric is the weighted average velocity of the ego vehicle based on the probability distributions for two seconds in advance, based on the lowest 20%. Finally, the conservative future velocity of the lead vehicle up to 2 seconds in advance is calculated based on the probability distribution. The conservative velocity value will ensure that the time gap between the ego vehicle and the lead vehicle never becomes too small, leading to heavy braking. Therefore the lowest 20% will always ensure that the ego vehicle will keep a safe distance to the lead vehicle and establish a safe and comfortable ride for the passengers.

The motion prediction model was created while assuming the following limitations. There were no data sets available, resulting in using the driving simulator toolbox to create the scenarios. Secondly, the sensors on the GRT cannot detect the acceleration of the surrounding vehicles, resulting in only position and velocity as input parameters. Despite the limitations of this thesis, it can be concluded that the results of Chapter 4 still show the importance of motion prediction to create a proactive behaviour for the ego vehicle. The proactive behaviour will increase the smoothness of the ego vehicle's path and comfort for the passengers. Especially if the ego vehicles consider the conservative path for the lead vehicle, ensuring a safe time gap to the vehicle in front, excluding the activation of emergency braking.

5-2 Recommendations

Although this work created a motion prediction model for the surrounding vehicles of the ego vehicle, several improvements could be added to the motion model described in this thesis. As a recommendation for further research, the following points should be considered.

- This thesis has proposed a solution based on the application of 2getthere. Therefore certain state information of the surrounding vehicles could not be used in the motion model. Therefore it would be preferred that in future research, the acceleration of the surrounding vehicles is taken into account when predicting the velocity profiles of the surrounding vehicles.
- The simulations of the current motion model are created using the Driving simulator toolbox in MATLAB. This toolbox cannot create smooth velocity profiles for the vehicles in the scenario. Therefore, it would be recommended that the motion model uses actual data for future research.
- The current motion model predicts the future movement and probabilities of a vehicle at every time instance. The probabilities are solely based on the currently predicted trajectories. For future research, the previous predictions should be stored by the prediction model. So they can be used as a reference for better probability calculations.
- The motion model should be combined with the ego vehicle's decision-making and path planning as an addition to the current motion model. By creating feedback to the path planner, the prediction model can actively influence the actual path of the ego vehicle based on the predicted motion for the leading vehicle.

Bibliography

[1] ASAM OpenDRIVE.

[2] P. Bender, J. Ziegler, and C. Stiller. Lanelets: Efficient map representation for autonomous driving. *IEEE Intelligent Vehicles Symposium, Proceedings*, (Iv):420–425, 2014.

[3] J. Dequaire, P. Ondrúška, D. Rao, D. Wang, and I. Posner. Deep tracking in the wild: End-to-end tracking using recurrent neural networks. *International Journal of Robotics Research*, 37(4-5):492–512, 2018.

[4] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.C. Chou, T.H. Lin, N. Singh, and J. Schneider. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, pages 2084–2093, 2020.

[5] Noel E. Du Toit and Joel W. Burdick. Probabilistic collision checking with chance constraints. *IEEE Transactions on Robotics*, 27(4):809–815, 2011.

[6] A. Elfes. Using Occupancy Grids for Mobile Robot Perception and Navigation. *Cornegie Mellon University*, 22(6):46–57, 1989.

[7] T. Fraichard and C. Laugier. Path-velocity decomposition revisited and applied to dynamic trajectory planning. *Proceedings - IEEE International Conference on Robotics and Automation*, 2(December 2014):40–45, 1993.

[8] M. Garcia Ortiz, J. Fritsch, F. Kummert, and A. Gepperth. Behavior prediction at multiple time-scales in inner-city scenarios. *IEEE Intelligent Vehicles Symposium, Proceedings*, (July):1068–1073, 2011.

[9] Z. Ghahramani. An Introduction to Hidden Markov Models and Bayesian Networks. *International journal of pattern recognition and artificial intelligence*, 15, 2001.

[10] T. Gindele, S. Brechtel, and R. Dillmann. A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (000 1):1625–1631, 2010.

[11] S. Hoermann, M. Bach, and K. Dietmayer. Dynamic Occupancy Grid Prediction for Urban Autonomous Driving: A Deep Learning Approach with Fully Automatic Labeling. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2056–2063, 2018.

[12] H. Hou, L. Jin, Q. Niu, Y. Sun, and M. Lu. Driver intention recognition method using continuous hidden markov model. *International Journal of Computational Intelligence Systems*, 4(3):386–393, 2011.

[13] A. Houenou, P. Bonnifait, and V. Cherfaoui. Risk assessment for Collision Avoidance Systems. *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, (October):386–391, 2014.

[14] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. *IEEE International Conference on Intelligent Robots and Systems*, (November):4363–4369, 2013.

[15] Y. Hu, W. Zhan, and M. Tomizuka. Probabilistic Prediction of Vehicle Semantic Intention and Motion. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018-June(October):307–313, 2018.

[16] C. Hubmann, M. Aeberhard, and C. Stiller. A generic driving strategy for urban environments. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (July):1010–1016, 2016.

[17] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller. Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles. *IEEE Intelligent Vehicles Symposium, Proceedings*, (Iv):1671–1678, 2017.

[18] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller. Automated Driving in Uncertain Environments: Planning with Interaction and Uncertain Maneuver Prediction. *IEEE Transactions on Intelligent Vehicles*, 3(1):5–17, 2018.

[19] K. Jo, K. Chu, and M. Sunwoo. Interacting Multiple Model Filter-Based Sensor Fusion of GPS With In-Vehicle Sensors for Real-Time Vehicle Positioning. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):973, 2012.

[20] K. Kant and S.W. Zucker. Toward Efficient Trajectory Planning: The Path-Velocity Decomposition. *The International Journal of Robotics Research*, 5(3):72–89, 1986.

[21] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, A. Wedel, and W. Rosenstiel. Object-Oriented Bayesian Networks for Detection of Lane Change Maneuvers. *IEEE Intelligent Transportation Systems Magazine*, 4(3):19–31, 2012.

[22] A. Kesting, M. Treiber, M. Schönhof, F. Kranke, and D. Helbing. Jam-Avoiding Adaptive Cruise Control (ACC) and its Impact on Traffic Dynamics. *Traffic and Granular Flow'05*, pages 633–643, 2007.

[23] S. Klingelschmitt, M. Platho, H.M. Groß, V. Willert, and J. Eggert. Combining behavior and situation information for reliably estimating multiple intentions. *IEEE Intelligent Vehicles Symposium, Proceedings*, (June):388–393, 2014.

[24] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier. Learning-based approach for online lane change intention prediction. *IEEE Intelligent Vehicles Symposium, Proceedings*, (June):797–802, 2013.

[25] R. Kumar and R. Pathak. Adaptive Cruise Control - Towards a Safer Driving Experience. 3(8):3–7, 2012.

[26] S. Lefèvre, J. Ibañez-Guzmán, and C. Laugier. Context-based estimation of driver intent at road intersections. *IEEE SSCI 2011: Symposium Series on Computational Intelligence - CIVTS 2011: 2011 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems*, (June 2014):67–72, 2011.

[27] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán. Intention-Aware Risk Estimation for General Traffic Situations, and Application to Intersection Safety. *INRIA Research Report No. 8379*, (October), 2013.

[28] S. Lefèvre, D. Vasquez, and C. Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal*, 1(1):1–14, 2014.

[29] A. Hennecke M. Treiber and D. Helbing. *Traffic and Granular Flow '99*. Berlin, 2000.

[30] H.M. Mandalia and D.D. Salvucci. Using support vector machines for lane-change detection. *Proceedings of the Human Factors and Ergonomics Society*, pages 1965–1969, 2005.

[31] N. Mohajerin and M. Rohani. Multi-step prediction of occupancy grid maps with recurrent neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:10592–10600, 2019.

[32] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer. A random finite set approach for dynamic occupancy grid maps with real-time application. *International Journal of Robotics Research*, 37(8):841–866, 2018.

[33] P. Manfredo. Differential Geometry of Curves and Surfaces, 1976.

[34] F. Poggenhans, J.H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr. Lanelet2: A high-definition map framework for the future of automated driving. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018-Novem:1672–1679, 2018.

[35] M. Schreier, V. Willert, and J. Adamy. From grid maps to Parametric Free Space maps - A highly compact, generic environment representation for ADAS. *IEEE Intelligent Vehicles Symposium, Proceedings*, (September 2014):938–944, 2013.

[36] M. Schreier, V. Willert, and J. Adamy. Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems. *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, (October):334–341, 2014.

[37] M. Schreier, V. Willert, and J. Adamy. An Integrated Approach to Maneuver-Based Trajectory Prediction and Criticality Assessment in Arbitrary Road Environments. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2751–2766, 2016.

[38] R. Schubert, E. Richter, and G. Wanielik. Comparison and evaluation of advanced motion models for vehicle tracking. *Proceedings of the 11th International Conference on Information Fusion, FUSION 2008*, (1):730–735, 2008.

[39] J. Schulz, K. Hirsenkorn, J. Lochner, M. Werling, and D. Burschka. Estimation of collective maneuvers through cooperative multi-Agent planning. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 624–630, 2017.

[40] J. Schulz, C. Hubmann, J. Lochner, and D. Burschka. Interaction-Aware Probabilistic Behavior Prediction in Urban Environments. *IEEE International Conference on Intelligent Robots and Systems*, pages 3999–4006, 2018.

[41] D. Shin, S. Yi, K.M. Park, and M. Park. An interacting multiple model approach for target intent estimation at urban intersection for application to automated driving vehicle. *Applied Sciences (Switzerland)*, 10(6), 2020.

[42] S. Steyer, G. Tanzmeister, and D. Wollherr. Object tracking based on evidential dynamic occupancy grids in urban environments. *IEEE Intelligent Vehicles Symposium, Proceedings*, (IV):1064–1070, 2017.

[43] Thomas Streubel and Karl Heinz Hoffmann. Prediction of driver intended path at intersections. *IEEE Intelligent Vehicles Symposium, Proceedings*, (June 2014):134–139, 2014.

[44] C. Sung, D. Feldman, and D. Rus. Trajectory clustering for motion prediction. *IEEE International Conference on Intelligent Robots and Systems*, pages 1547–1552, 2012.

[45] A. Takahashi, T. Hongo, Y. Ninomiya, and G. Sugimoto. Local path planning and motion control for agv in positioning. *IEEE International Conference on Intelligent Robots and Systems*, 1989-Septe:392–397, 1989.

[46] M Toyungyernsub, M Itkina, R Senanayake, and M.J. Kochenderfer. Double-prong ConvLSTM for spatiotemporal occupancy prediction in dynamic environments. *arXiv*, 2020.

[47] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 62(2):1805–1824, 2000.

[48] D. Vasquez, T. Fraichard, and C. Laugier. Incremental learning of statistical motion patterns with growing hidden markov models. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):403–416, 2009.

[49] A. Von Eichhorn, M. Werling, P. Zahn, and D. Schramm. Maneuver prediction at intersections using cost-to-go gradients. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (October):112–117, 2013.

[50] M. Werling, S. Kammel, J. Ziegler, and L. Gröll. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. *International Journal of Robotics Research*, 31(3):346–359, 2012.

[51] M. Werling, J. Ziegler, S. Kammel, and S. Thrun. Optimal trajectory generation for dynamic street scenarios in a frenét frame. *Proceedings - IEEE International Conference on Robotics and Automation*, (March):987–993, 2010.

[52] W. Yao, H. Zhao, F. Davoine, and H. Zha. Learning lane change trajectories from on-road driving data. *IEEE Intelligent Vehicles Symposium, Proceedings*, (June):885–890, 2012.